REAL-TIME DDOS DETECTION BASED ON PREDICTIVE MULTI- AND POLYSCALE METRICS FOR CYBER-PHYSICAL SYSTEMS INTERNET TRAFFIC

by

JESUS DAVID TERRAZAS GONZALEZ

A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfilment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada

(xlv + 178 + Appx. 205) = 428 pp.

To

God the High and Exalted One, Isaiah 52:13
Father, Son (Christ the Saviour), and Holy Spirit
Omnipresent, Omnipotent, and Omniscient
Psalm 139:1-6 | Psalm 139:7-12 | Psalm 139:13-18

my beloved mother, who is a star in the sky
Amparo
spiritual, brave, patient, sweet, vigorous, and an amazing believer

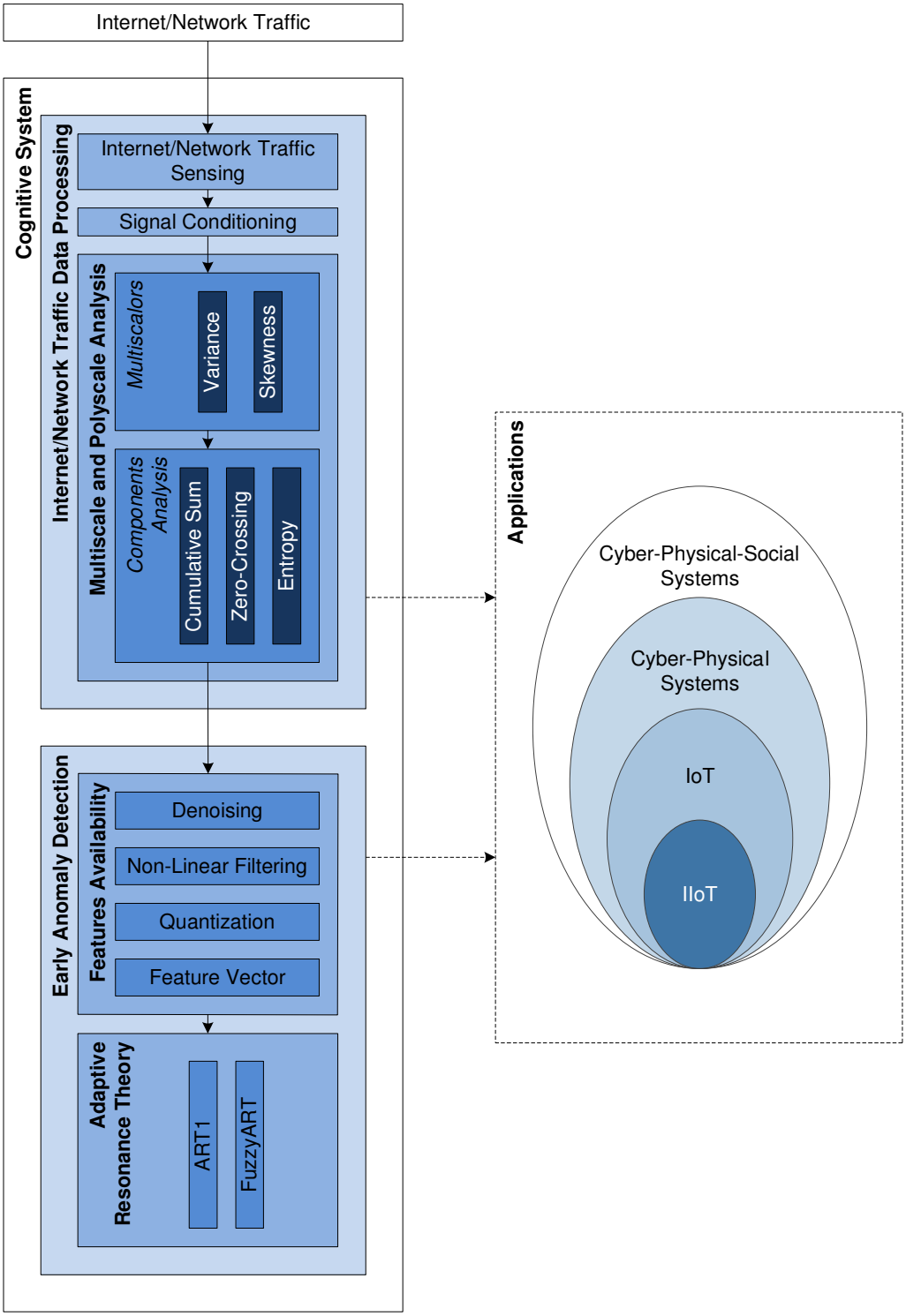
my purposeful father
Jeronimo
courageous and unwavering

my dear
Ting
caring, gentle, gracious, truthful, and virtuous

my precious
Nicholette Amparo
a star twinkling multicolour light into my life

all my family and the generations after
together in unity and for overcoming adversity
Psalm 133:1 | Proverbs 17:17

VISUAL ABSTRACT



ABSTRACT

This research investigates the appropriateness of Information-Theoretic-Based (ITB) metrics compliant with finite sense stationarity (FSS) and derived from the Variance Fractal Dimension Trajectory (VFDT), to augment network security against traffic anomalies. From the distinct and vast cyberattacks (infection, exploitation, probing, deception, cracking, concurrency, and unknown) types, this research focuses in those stemming from concurrency and specifically in Distributed Denial-of-Service (DDoS) cyberattacks.

In this research, the design and application of robust methodologies and metrics to achieve powerful descriptors is pursued. The strength of ITB metrics, applied in alternate research areas like steganography, is a robust justification for this study. The usage of ITB metrics, rooted in multi- and polyscale analysis, for detecting network disruptions is novel in the network security area. This thesis introduces a novel multiscale analysis methodology, *multiscalors*, which permits the usage of arbitrary operators and transforms to be functional in the multiscale domain for inspecting complex signals. Multiscalors provide an analysis depth and insights into the signals that exceeds by far what other types of monoscale based analysis offer. Multiscale-based metrics have been scarcely utilized in the cybersecurity ecosystem. This thesis also showcases specific applications of metrics and methodologies powered by multiscale analysis for DDoS detection.

The methodology presented formulates robust features, based on multi- and polyscale analysis, and successfully classifies DDoS disruptions. Such methodology integrates knowledge from: (i) Data acquisition, by verifying DDoS instances and deriving complementary data from them; (ii) design and implementation of ITB metrics, based on multiscalors operators for analysis; (iii) feature extraction, by applying such metrics to the PREDICT datasets, (iv) preparation of feature vectors that are highly representative of the Internet traffic characteristics carrying DDoS cyberattacks, and (v) classification of anomalies through Adaptive Resonance Theory (ART) as a non-supervised neural network that has provided the real-time component in the detection of DDoS attacks establishing the *time classification* in the one second mark. Concerning ART, through this research a new methodology, *parametogram*, for properly defining the vigilance parameter for both classification approaches used, ART1 and FuzzyART, has been designed, tested, and validated.

Applications of the multiscalors based metrics in this research target Cyber-Physical-Social Systems (CPSS), *e.g.*, Industrial Internet-of-Things (IIoT) sustained by the fact of the usage of non-simulated Internet traffic, which contains legitimate DDoS attacks. This research corroborated the detection of anomalies in Internet traffic with a high classification precision for which the multiscalor methodology is essential for extracting relevant features characterizing the DDoS cyberattacks examined.

ACKNOWLEDGEMENT

Through the course of my professional and academic preparation, there have been countless persons always *encouraging* me to grow daily, *motivating* me to achieve higher goals, and *contributing* priceless resources in order to materialize dreams into reality.

I express my appreciation to my advisor, Dr. Witold Kinsner, who has worn several hats, educator, teacher, friend, counsellor, or mentor, through his goal-oriented interactions focused in research. From the moment I met him, new dimensions have been added continuously in my professional and academic preparation.

I recognize with much appreciation the PhD Committee, Dr. Stephen Pistorius, Dr. Josh van Rees, and Dr. Ekram Hossain, who has always been available always to follow up with my research, provided great suggestions during our meetings, criticized my ideas constructively for the betterment of my research, and opened paths for success at all times.

My gratefulness goes to the many staff members of different departments and offices at the University of Manitoba, who have enhanced my education and certainly maximized my student experience in this house of learning. My student life has been immensely impacted by caring and genuine individuals committed to their positions. I am also appreciative of my extensive network of friends which now spans around the globe, my academic life has been positively impacted by them incalculably.

My deepest admiration for my parents, Amparo and Geronimo, who did absolutely everything to raise me, having an attitude of betterment for this planet and everybody. Even though, you departed not long ago, mama, your presence will live forever in my heart and mind. Your life in this planet went by as a sweet and gentle breeze, which I wish would have lasted much much longer. You will be forever missed. Your example of tenacity, patience, and generosity, has been, is, and will always continue to be a strong pillar for anchoring many of my dreams. I am thankful to my dad, who has always guided me for the best paths of life and has encouraged me to overcome all kinds of challenges. Your example of strength, determination, and ingenious mind has taught me to always look forward for what the future has to bring. I always remember you, I am deeply proud of you, and I find happiness honouring you both. You are inherently one of the motivations for any successful event that I achieve.

Furthermore, I am thankful to all the educators, teachers, and professors that collectively have contributed in my academic preparation in all stages of my education. I have made it to this level partly because of your commitment and effort towards teaching.

Additionally, I would like to warmly and specially thank the National Science and Technology Council in Mexico, *Consejo Nacional de Ciencia y Tecnologia* (CONACYT), for awarding me a scholarship to pursue a PhD in Electrical and Computer Engineering abroad. It has certainly been a life changing experience that I am glad CONACYT allowed me to undertake.

Great is our Lord, and of great power: His understanding is infinite.

Psalms 147:5

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AAAS	American Academy of Arts and Sciences
AC	Alternating current
ACM	Association for Computing Machinery
AcR	Accuracy ratio
ADI	Active digital identity
ADP	Adaptive dynamic programming
AI	Artificial Intelligence
AIR	Automated Intrusion Response
ANN	Artificial neural network
APA	Adaptive-persistent adversaries
API	Application programming interface
AR	Augmented reality
ARP	Address resolution protocol
ART	Adaptive resonance theory
ASIC	Application-specific integrated circuits
BE3	BlackEnergy3
BGP	Border gateway protocol
BNN	Backpropagation neural networks
BP	Backpropagation
C&C	Command and control
CAN	Computer and network attacks
CalTech	California Institute of Technology
CAS	Complex adaptive system
CDJV	Interval-adapted pyramidal filtering algorithm of Cohen, Daubechies, Jawerth, and Vial
CDMA	Code division multiple access
CDN	Content delivery network
CERT	Computer Emergency Response Team
CIRA	Canadian Internet Registration Authority
CLP	Conditional legitimate probability
CNSA	Common network security adversaries
COI	Class-of-Interest
COTS	Commercial off-the-shelf
CPS	Cyber-physical-system
CPSS	Cyber-physical-social system
CSMA/CA	Carrier sense multiple access with collision avoidance
CSS	Computer security system
CTO	Chief technology officer
CSU	Colorado State University
CySS	Cyber-social systems
DARPA	Defense Advanced Research Projects Agency

DCM	Data collection mechanisms
DCS	Distributed control systems
DDoS	Distributed denial-of-service
DF	Domain flux
DFT	Discrete Fourier transform
DGA	Domain generation algorithm
DHS	Department of Homeland Security
DL	Deep learning
DNS	Domain name system
DoS	Denial-of-service
DPWS	Device Profile for Web Services
DRDoS	Distributed reflection denial-of-service
DT	Decision tree
EATCS	European association for theoretical computer science
ECG	Electrocardiogram
EDoS	Economic denial-of-sustainability
EMG	Electromyogram
ERF	Extensible record format
EU	European Union
FA	Flux agents
FAR	False Alarm Rate
FaaS	Fog-as-a-service
FBI	Federal Bureau of Investigation
FDMA	Frequency division multiple access
FEN	Fog edge nodes
FF	Fast-flux
FFM	Fast flux monitor
FFN	Fast fluxing network
FFNA	Fast fluxing network attack
FFSN	Fast flux service network
FN	False negative
FoF	Factory of the future
FOF	Features of features
FP	False positive
FPGA	Field programmable gate arrays
FRC	Fraudulent resource consumption
FRD	Fog reference design
FS	Fog server
FSS	Finite sense stationarity
GA	Genetic algorithm
GPS	Global positioning system
GRN	Gaussian random noise
GWN	Gaussian white noise
H2H	Human-to-human

H&R	Hit and run
HSN	Human social networks
HTML5	Hypertext mark-up language version 5
HTTP	Hyper-text transfer protocol
I	Infectious
IaaS	Infrastructure-as-a-service
ICMP	Internet control message protocol
ICN	Information centric network
ICS	Industrial control system
ICT	Information and communication technology
idd	Independent and identically distributed
IDS	Intrusion detection system
IEEE	Institute of Electrical and Electronic Engineers
IEEE 802	Networking protocols family of standards
IEEE 802.3	Ethernet networking protocol standards
IEEE 802.11	Wireless local area networks and Wi-Fi networking protocol standards
IEEE 802.15.1	Bluetooth networking protocol standards
IEEE 802.15.4	ZigBee networking protocol standards
IEEE 802.15.6	Body area networks networking protocol standards
IEEE 802.15.7	Visible light communications networking protocol standards
IEEE 802.24	Smart Grid networking protocol standards
IIoT	Industrial Internet-of-Things
INL	Idaho National Laboratory
IoP	Internet-of-People
IoS	Internet-of-Services
IoT	Internet-of-Things
IoTSN	IoT social networks
IoV	Internet-of-Vehicles
IP	Internet protocol
IPS	Intrusion prevention system
IP2HC	IP-to-hop-count
IRC	Internet relay chat
ISI/USC	The Information Sciences Institute at the University of Southern California
ISP	Internet service provider
ISR	Intelligence, surveillance, and reconnaissance
IT	Information technology
ITB	Information Theoretic Based
KL	Kullback-Leibler divergence
LAN	Local area network
LHC	Large Hadron Collider
LMS	Least Mean Square
LTI	Linear time invariant
LSQ	Least squares quantization
LTM	Long-term memory

M2M	Machine-to-machine
M&C	Monitoring and control
MAC	Medium access control
MCPSS	Medical-cyber-physical-social systems
MES	Mean energy of the source
MESmax	Peak source energy
MISE	Mean integrated squared error
MIT	Massachusetts Institute of Technology
MITM	Man in the middle
MLP	Multilayer perceptron
MNN	Multilayer neural networks
MoE	Mixture of experts
MOS	Mean Opinion Score
MSE	Mean squared error
MSN	Mobile social networks
NAT	Network address translation
NFC	Near field communications
NSA	National Security Agency
NSE	Network Security Engine
NHS	National Health Service
PAC	Programmable automation controller
PAM	Pulse-amplitude modulation
PaaS	Platform-as-a-service
PCM	Pulse-code modulation
pdf	Probability distribution function
PLC	Programmable logic controller
pmf	probability mass function
pps	Packets per second
PREDICT	Protected Repository for the Defense of Infrastructure Against Cyber Threats
PSD	Power spectrum density
OCDF	Observed cumulative distribution function
OG	Overgeneralization
opdf	observed probability distribution function
OS	Overspecialization
OSI	Open systems interconnect
OSPF	Open shortest path first
PPM	Probabilistic packet marking
QoS	Quality-of-service
R	Recovered
RAS	Reliability, availability, serviceability
RESTful	Representational State Transfer
RF	Random forest
RFID	Radio frequency identification
RIP	Routing information protocol

RPS	Requests per second
S	Susceptible
SaaS	Software-as-a-service
SCADA	Supervisory control and data acquisition
SDN	Software defined network
SI	Susceptible-infectious model
SIGACT	Special Interest Group on Algorithms and Computational Theory
SIoT	Social IoT
SIoV	Social IoV
SIR	Susceptible-infectious-recovery model
SIS	Susceptible-infectious-susceptible model
SNR	Signal to noise ratio
SOA	Service-oriented architecture
SOM	Self-organized maps
SoS	System of systems
SPD	Stability-plasticity dilemma
SSS	Strong sense stationarity
STM	Short-term memory
SVM	support vector machine
SVHN	Street View House Numbers
TAR	True Alarm Rate
TCP	Transfer control protocol
TDMA	Time division multiple access
TN	True negative
TP	True positive
TTL	Time-to-live
UDP	User datagram protocol
updf	Underlying probability distribution function
UWN	Uniform white noise
VANET	Vehicular ad hoc network
VFD	Variance fractal dimension
VFDT	Variance fractal dimension trajectory
VLSI	Very large integrated circuits
VM	Virtual machine
VNI	Visual networking index
VPN	Virtual private network
VSN	Vehicular social network
WBAN	Wireless body area network
WoT	Web-of-Things
WSAN	Wireless sensor and actuator network
WSN	Wireless sensor network
WSS	Wide sense stationarity
WTA	Winner takes all
ZCR	Zero-crossing rate

LIST OF SYMBOLS

$\sim$	Proportional to
$\triangleq$	Relation by definition
$\|\|\|$	Shorthand notation indicating a Ξ operator is applied in multiscale analysis
$\|\|\|^n$	A specific multiscalor component
$\equiv$	Equivalence
$\{\bullet\}$	Set notation
$\{x_i\}$	Neuron inputs
$\{w_i\}$	Synaptic weights
$[\bullet]$	Vectorial notation
$\zeta(\bullet)$	Accumulator function
Γ	Code alphabet
Γ_C	Coding alphabet
Σ	Source alphabet composed of elements σ_j
$\bigcap$	Set intersection operation utilized in ART1
$\wedge$	Fuzzy MIN operator or continuous AND utilized in FuzzyART
$[h_m, h_n]$	Time-to-live range
α	Infection rate
α_{FA}	Choice parameter for FuzzyART
α_i	Legitimate traffic when hackers spoof a single source
α_j	Attack flows magnitude with the delay j at the edge router
α_k	Legitimate traffic when hackers are aware of detection mechanisms
α_P	Pareto index
$\alpha(t)$	Time zone information
$\mathbf{a}$	Incoming vector illustrating normalization option in FuzzyART
a_j	Label where the sample $s(t_j)$ falls in for quantization
A	Automatic response to an attack
A_i	Attack flows
A_{tec}	Attack method and/or technique
A_{res}	Attack harmful results
A_{sou}	Attack source
$A.l[n]$	Aggregated DDoS attack flows length integration
$A_i.l[n]$	A single DDoS attack flow length integration
$A.p[n]$	Aggregated DDoS attack flows packet integration

$A_i.p[n]$	A single DDoS attack flow packet integration
$\{a_1, a_2, ...\}$	Value spaces in packets attribute A
$A, B, ...$	Packets attributes
β	Pairwise rate of infection
β_{FA}	Learning rate parameter in FuzzyART
b	Index for a b-adic process
$B(t)$	Signal continuous or discrete in time
$\hat{C}$	A given statistics estimate in Freedman-Diaconis' Binning Rule
C_i	Legitimate client
C_n	A given coefficient for an exponential function
$C_n(\Delta)$	Cost function to be minimized in Shimazaki-Shinomoto's choice
$CLP(\bullet)$	Conditional legitimate probability
δ	Discrimination threshold in indication function I_{X_i,X_j}
δt	Sampling period
Δ	Unit of time
Δ_H	Decrease in Shannon's entropy
ΔB	Amplitude difference in signal B
Δt	Sample displacement
d_i	Noisy data element
$D(p,q)$	Kullback-Leibler (KL) distance for $p \neq q$
$D_H(p,q)$	Hellinger distance
$D_J(p,q)$	Jeffrey distance
$D_S(p,q)$	Sibson distance
D_σ	Variance fractal dimension
$DFT[R_{XX}]$	Autocorrelation DFT
e	Natural logarithm
e_I	Empirical wavelet coefficients in thresholding denoising
E	Euclidian dimension
$E[\bullet]$	Expectation
ER	Error rate
$\hat{f}$	Predicted value for f
f	Unknown function in Donoho's sense
f_0	Nominal frequency
f_N	Nyquist frequency
f_p	Probability mass function
f_s	Sampling frequency

fn	False negative
fp	False positive
$F[n]$	Frame for a multiscalor operator
$F1$	Comparison layer in ART
$F1_j$	A given neuron in the comparison layer $F1$ in ART
$F2$	Recognition layer in ART
$F2_j$	A given neuron in the recognition layer $F2$ in ART
FNR	False negative rate
FPR	False positive rate
FV_n	A given feature vector
g	Activation function
G_i	Genuine traffic flows
$\hat{h}_D$	Doane's binning rule
$\hat{h}_{FD}$	Freedman-Diaconis' binning rule
h_k	Hop count
$\hat{h}_{Sc}$	Scott's binning rule
$\hat{h}_S$	Sturges' binning rule
H	Hurst exponent
$H(\bullet)$	Shannon's entropy
$H[m_{2\|\|\|^n}]$	Shannon's entropy applied to each variance multiscalor component
$H[m_{3\|\|\|^n}]$	Shannon's entropy applied to each skewness multiscalor component
$H_f(X)$	Entropy of flows
idd	Independent and identically distributed
int	Integer value
I	Infectious individuals
$\mathbf{I}$	Vector of elements $(I_1, I_2, ..., I_N)$. Inputs for ART.
I_A	Indicator for DDoS attacks
I_i	Input binary pattern for ART1
I_j	Shannon's self-information
I_t	Infected hosts at time t
I_0	Hosts infected initially
I_{X_i,X_j}	Function indicating whether a DDoS attack is present. Flow correlation coefficient
j	Index used to traverse samples in given window for a VFD calculation
J	Neuron receiving the largest T_j input from the $F1$ layer

J_{cm} Total number of classes used in a confusion matrix
k Number of cycles for computing a VFD
$k_\sigma(\bullet)$ Gaussian kernel
K_{buf} Number of points in the log-log plot that are discarded to obtain a VFD
K_{hi} Maximum number of points in the log-log plot that are considered to obtain a VFD
K_{max} Maximum number of points included in the log-log plot to obtain a VFD
K_{low} Minimum number of points in the log-log plot to obtain a VFD
$K(t)$ Sinc pulse
λ Birth rate
ℓ^1 The L^1 norm
ℓ^2 The L^2 norm (Euclidean distance)
L Parameter for the calculations of the synaptic connections z_{ij}^{bu}
lim Function limit
$\log_b$ Logarithm base b
μ Mean. Expected value
$\mu(\bullet)$ Generalization of measure
$m_{1_{ss}}$ First statistical moment in choice cost function $C_n(\Delta)$
m_2 Variance. Second statistical moment
$m_{2_{ss}}$ Second statistical moment in choice cost function $C_n(\Delta)$
$m_{2|||^n}$ A given component of the variance multiscalor
m_3 Skewness. Third statistical moment
$m_{3|||^n}$ A given component of the skewness multiscalor
M Bag of all the symbols and strings that forms a message
M Number of neurons in the $F2$ layer in ART1
MAD Median absolute deviation
Mo Mode
$\eta_t(\bullet)$ Soft thresholding nonlinearity in thresholding denoising
n Sample index in a time series
n_c Number of committed neurons in ART1
n_i Legitimate packets
n_j Event in the Shannon's entropy sense
n_k Number of samples used to calculate a VFD at a given cycle k
$\mathbb{N}$ Natural numbers
N Total population
N_a Attack packets

N_{id}	Unique node identifiers
N_S	Size of the source alphabet in Shannon's entropy sense
N_k	Number of windows to calculate a VFD in a frame
N_m	Measured packets
N_n	Normal packets
N_T	Sample space
$\rho_{X_i,X_j}[k]$	Correlation coefficient
p_j	Probability of an event in the Shannon's entropy sense (j^{th} symbol)
p	Packet
$p.X$	Attribute X of packet p
p_k	Hop-count h_k probability
P	Precision
Pr	Probability
$\Pr_a$	Probability for attack packets
$\Pr_a(a_i, \ ...)$	Joint probability among attack packets with attributes
$\Pr_m$	Probability for measured packets
$\Pr_m(a_i, \ ...)$	Joint probability among measured packets with attributes
$\Pr_n$	Probability for normal packets
$\Pr_n(a_i, \ ...)$	Joint probability among normal packets with attributes
q_α	A set of quanta with a v finite value used in quantization
Q_α	Disjoint v sets used in quantization
$Q[S[m_{2\|\|^n}]]$	Quantization of cumulative sum applied to each variance multiscalor component
$Q[S[m_{3\|\|^n}]]$	Quantization of cumulative sum applied to each skewness multiscalor component
$Q[H[m_{2\|\|^n}]]$	Quantization of Shannon's entropy applied to each variance multiscalor component
$Q[H[m_{3\|\|^n}]]$	Quantization of Shannon's entropy applied to each skewness multiscalor component
$Q[Z_n[m_{2\|\|^n}]]$	Quantization of ZCR applied to each variance multiscalor component
$Q[Z_n[m_{3\|\|^n}]]$	Quantization of ZCR applied to each skewness multiscalor component
$r_{X_i,X_j}[k]$	Correlation between two flows X_i and X_i for $i \neq j$ with the same length N
ρ	Vigilance parameter in ART
$\rho_{X_i,X_j}[k]$	Correlation coefficient between two flows X_i and X_j
$\rho_{X_p,X_q}[k]$	Correlation coefficient between two flash crowds X_p and X_q

R Recovered individual
R Recall
R_j Rectangle with corners defined by vectors $\mathbf{u}_j$ and $\mathbf{v}_j$
$R_{XX}(\bullet)$ Autocorrelation function
$\hat{\sigma}$ Standard deviation estimate
σ Standard deviation
σ^2 Variance
σ_D Noise level in Donoho's sense
σ_j Element (source symbol) of a source alphabet Σ
σ_{m_3} Tuning parameter in Doane's binning rule
s Slope
$s(t)$ Voltage signal
$s(t_j)$ Value of s at the j^{th} sampling instant
S Susceptible individual
S Cumulative sum
$S[m_{2\|\|^n}]$ Cumulative sum applied to each variance multiscalor component
$S[m_{3\|\|^n}]$ Cumulative sum applied to each skewness multiscalor component
S_F Total streams present in the Internet/network traffic
sgn(•) Sign function
$S.l[n]$ Total stream flows length integration
$S.p[n]$ Total stream flows packet integration
$t_{(\bullet)}$ Time sampling instant
t_i An element of standard Gaussian white noise
$\hat{t}_n$ Estimated threshold for Donoho's denoising
t_n Chosen threshold for Donoho's denoising
tn True negative
tp True positive
T Total time over which a sample space N_T is obtained
T_j Choice functions. Input to the j^{th} $F2$ neuron in ART1
T_J Choice function for the active neuron in the layer $F2$ in ART1
γ Skewness
$\gamma(x)$ Label function in quantization
γ_1 Constant for Donoho's denoising
γ_j Encoding symbols of a coding alphabet Γ_c
γ_{no} Normalization option for FuzzyART

u	Activation potential
$\mathbf{u}_j$	Vector containing $\mathbf{a} = (a_1,\ a_2)$
$\mathbf{v}_j^c$	Vector containing $\mathbf{a}^c = (1-a_1,\ 1-a_2)$
var(•)	Variance operator
$\hat{V}_{m,\sigma}(A,\ B)$	Correntropy for two finite data sequences A and B
V_i	Input from the i^{th} $F2$ neuron to the $F1$ in ART1
$\mathbf{V}$	Vector of elements $(v_1,\ v_2,\ ...,\ v_N)$
W	A frequency value used in Lloyds quantization
w[•]	Window with a stationary segment of a signal for ZCR digital computation
θ	Activation threshold or bias
Ξ	Arbitrary operator applied in a multiscale approach
$\Xi_{\|\|\|^n}$	A given component of a given multiscalor
x_i	Individual realization of the random variable X
x_N	A given neuron in the recognition layer $F1$
$x[n]$	Ordered sequence of numbers (time series) in relation to sample index n
χ	Sample space of X
X	Discrete random variable
$\mathbf{X}$	Vector of elements $(x_1,\ x_2,\ ...,\ x_N)$
$X_i[n]$	Network flow where $i \geq 1$ represents the network flows, and n denotes the n^{th} element in the data sequence
X_k	Abscissa value in the log-log plot at a given cycle k
X_p and X_q	Flash crowds
$X'_0[j]$	Delayed fingerprints by j time units
X'_i	Fingerprint of flow X_i
y_j	Output from the j^{th} neuron in the $F2$ layer in ART1
Y_i and Y_j	Noise flows
Y_k	Ordinal value in the log-log plot at a given cycle k
$\mathbf{z}_j$	Weight vector in ART
z_i	Standard Gaussian white noise
z_{ij}^{bu}	Strengths of the bottom-up synaptic connections
z_{ji}^{td}	Strengths of the top-down synaptic connections
Z	Zero-crossing rate
Z_{anti}	Detection rate when hackers are aware of detection mechanisms
Z_m	Detection rate for multiple sources

Z_n Zero-crossing rate in digital implementation

$Z_n[m_{2|||^n}]$ ZCR applied to each variance multiscalor component

$Z_n[m_{3|||^n}]$ ZCR applied to each skewness multiscalor component

Z_s Detection rate for single source

CHAPTER I

INTRODUCTION

1.1 Scientific, Engineering, and Humanitarian Preamble

Current research in engineering follows a twofold approach [Kins015] by embracing both the **scientific method** (*e.g.,* [Blak012]) and the **engineering method** (*e.g.*, [Koen003], [Kins009]). Theodore von Karman (an aerospace engineer from CalTech) attempted to explain the difference between science and engineering: "Science is about understanding nature, understanding what is. Engineering is about creating what has never been" [Wulf008]. Pure science is *analytic* with the intent to understand nature, while pure engineering is *synthetic* with the intent to build things. Many contemporary scientists and engineers operate between these two extremes [Kins015].

The difference between the two methods used to be very wide. However, with the advent of high-performance computer simulation tools, the difference is diminishing as simulated prototypes resemble physical prototypes, and multiple simulations can provide a much broader insight into its operation than it is possible through testing of a single prototype. Furthermore, many advanced technologies (such as nanotechnology devices) require deep science in order to do deep engineering [Kins015].

This unification is further justified by the shift from engineering design for enhanced consumer consumption (going back to the industrial revolution) to engineering for humanity [Kins015] as demonstrated by many new initiatives in major technical organizations such as the Institute of Electrical and Electronic Engineers (IEEE) and the American Academy of Arts and Sciences (AAAS) (*e.g.*, [Kins014], [Wulf008]). This social/humanitarian method should also be added to the scientific/engineering method in order to make the discovery/development process more stable (sustainable). The older single-method approach resembles a monopod —useful to

take quick snapshots on the go, but not useful to take a video. Even the two-method approach appears incomplete when one considers a responsible stewardship in a technological democracy [Kins015]. The research conducted and described herein is sustained by a tripod (*i.e.*, science, engineering, and humanitarian methods). The different research stages (proposal, experiments planned to be conducted, and conclusions presentation) presented here and its potential derived works, first and foremost, have a tripartite motivation looking for the betterment of our rapidly growing digital society. The specific problem undertaken in this research is described as follows.

The cybersecurity research presented in this thesis is very relevant because it summarizes years spent by the author in a cybersecurity framework that changed both dynamically and rapidly. These transformations occur as a consequence of the fast evolution of the technologies that enable and support the Internet communications, the appearance of new and more intense/harmful cyberattacks, the cybersecurity ecosystem struggling aggressively to be current and aware of new cyberthreats, and the cyberattackers finding new vulnerabilities and exploiting them. Were all the years spent in this research worthy? Absolutely. Having gone through all the raw data, signal acquisition, data analysis, techniques and methods, modeling, applied machine learning, and all the coding subtleties has been one of the most gratifying learning experience. Nonetheless, as this experience seems to conclude, this thesis encapsulates some of the dynamical operations in cyberspace, which are never ceasing and will always be coated with novelty in both the cyberoffense and cyberdefense fronts.

1.2 Motivation and Problem Definition

Recently, humanity started swimming in a new sea of technology with the Internet-of-Things (IoT) as its recent tidal waves. Everyday diverse devices (*e.g.*, air-conditioning systems, heating systems, thermostats, stoves, laundry machines, driers, refrigerators, TVs, to name a few) join the IoT ecosystem. Specialized networked industrial ecosystems, Industrial Internet-of-Things (IIoT), seek to take advantage of data flows for creating smart processes. However, humanity should not forget that the Internet has active predators exploiting everything (*e.g.*, software leading to hacks or connections leading to exposed hardware). This realization should (i) shake everyone because many aspects of human life are now networkable and sharable through software and hardware platforms (often in the form of codesign) and (ii) pose a

challenging question: *Who with a right mind jumps in water full of predators?* It is clear that adversaries threaten physical, economic, and national security. As an example in the physical world, adversaries have demonstrated hacking of an automobile connected to the Internet. This example of automobile hacking can be extrapolated to an industrial realm where a process controller can be abused and misused. When software hacking leads to deep implications in the physical world, very logically if humanity is dependent on Things that are indefensible either such dependence should be diminished, or such Things should be more defensible. Security researchers are working on the second option to catch up with predatory adversaries [Corm014].

In order to be a step ahead of cyberoffense, cybersecurity is in constant need of new analysis techniques, methodologies, and machine learning approaches that can effectively and accurately classify anomalies. This research embarks precisely in an attempt to diminish the gap between cyberattacks occurrence and their detection from the cybersecurity front. At the onset of this research, *distributed denial of service* (DDoS) detection ranks past a time interval that comprises minutes, if a well-trained machine learning model is used. This sets a challenge for the achievements in this research, if considered as a detection benchmark. In regard to the machine learning model selected, this has been pragmatically chosen to be adaptive resonance theory (ART), which specifically considers two of its variants ART1 and FuzzyART.

Events that occurred in the midst of the 2020 global pandemic, have demonstrated that the world around us functions in a sea of networked systems. Whether for virtual meetings, IoT devices, remote sensing, online shopping, reliable news, government and health advice, or when to go out to get groceries, these networks and the services that operate on them are vital to a functioning society. These networks and services have become not only critical, but also vital, infrastructure, in a way that mirrors discussions about roads, hospitals and electrical grids. The work presented in this thesis could facilitate efforts within the security ecosystem for improving resiliency.

The collective term for this underlying support is *infrastructure*, and when the associated services are considered essential to society, we refer to the associated underlying support as *critical infrastructure*. An additional popular definition of critical infrastructure is any underlying service support that, if removed (even temporarily), would create serious problems for society. If any of the critical infrastructure components became degraded or unavailable, the

consequences to society would be severe. A problem with critical infrastructure security is that practitioners tend to apply protections that were designed for smaller systems. This is an issue, because the needs of a large and small computer system can be as different as one might find for, say, a jumbo jet or bicycle. Maintenance, monitoring, trust, and compliance are example factors that are directly influenced by size, scale, and scope. The demands of infrastructure, however, especially in support of critical services, introduce considerably more risk, primarily due to the increased consequences of attack [Amor020].

There are 16 critical infrastructure sectors (*i.e.*, chemical, commercial facilities, communications, critical manufacturing, dams, defense industrial base, emergency services, energy, financial services, food and agriculture, government facilities, healthcare and public health, information technology, nuclear reactors, materials and waste, transportations systems, and water and wastewater systems) whose assets, systems, and networks, whether physical or virtual, are considered so vital to the United States that their incapacitation or destruction would have a debilitating effect on security, national economic security, national public health or safety, or any combination thereof [Cybe020]. The main grounding force in cybersecurity is having deep knowledge about the importance of the assets that are in need to be defended. This often boils down to intellectual property. However, different stakeholders might have different priorities and value for the assets that they own [Amor020].

It is vital that these network systems continue to operate in the face of adversity such as DDoS attacks. In order to respond to a DDoS attack, it is paramount that first the attack can be detected so that there is proper awareness about it. Classic methods for DDoS detection include: Packets profile based detection of time-to-live (TTL), packet score, spectral analysis, distances to distinguish between flash crowds and traffic carrying an attack, monoscale entropy detection, metrics similarity based on distances, and correlation of flows, among the most common. An in depth and extensive report on these methodologies is found in Appendix A.

Nonetheless, these methods are insufficient and limited because: Attackers could use random distributions to make a DDoS attack look like normal traffic, DDoS packets scores can be made to mimic normal traffic packets scores, measurements based on distances are prone to ignore significant parts of the signal of interest when this one is embedded in a dynamical environment, spectral analysis loses the connection to time unless a short-time form is used and

still then a precise connection to time cannot be achieved, attackers could make a DDoS attack to mimic a flash crowd, and monoscale entropy detection fails to detect DDoS efficiently because both normal traffic and a DDoS attack can have similar monoscale entropy values.

After considering how hard is to detect or isolate a DDoS attack from normal traffic and the shortcomings from the DDoS detection methods just commented on, it is clear that a DDoS detection method that could effectively and robustly perform within the challenging and dynamical nature of Internet traffic is needed. The core of this research is to assess if the performance of multi- and polyscale features could provide a valuable answer for this need.

Cybersecurity is an area already with a *past* that has thought significant lessons to computer security related companies and individuals; a *present* in which corporations and agencies cooperate for developing better computer security systems (CSSs) to battle against known or unknown forces wanting to access relevant data and computer assets; and certainly a *future* claiming for the embodiment of the very best CSS in which the most advanced mathematical theories and engineering practices developed by humans today would then be used to construct them. Cybersecurity has been a concern in the modern information era since the days of the Milwaukee 414s teenage hackers in the early 1980s (capable of breaking into the Los Alamos National Research Laboratory and the Sloan-Kettering Cancer Center amongst other prominent computer systems) [Voll015] until now. In the late 80s the "Morris worm" was released at the Massachusetts Institute of Technology (MIT) by Robert T. Morris (creator of one of the first web-based applications to build and host online stores sold to Yahoo in 2005) graduate student at Cornell University [Davi015]. The Morris worm, first computer worm Internet-distributed, infected computer systems at U.S. universities, research centers, and military bases and caused an estimated $20 million USD worth of damage [Davi015]. This attack prompted the Defense Advanced Research Projects Agency (DARPA) to fund the establishment of the Computer Emergency Response Team (CERT) at Carnegie Mellon University, in Pittsburgh, for anticipating and solving cybersecurity challenges [Davi015 and Pate015]. CERT until now continues partnering with government, academia, law enforcement, and industry to develop methods and tools to deal with cyberthreats [Pate015].

Many online resources exist that summarize the frequency and associated trends of different cybersecurity events. Some of these resources are the following: PrivacyRights is a

repository for data breach incident reports, Hackmageddon is a website which collects public reports of cybersecurity incidents, Databreaches.net is a website that collected databreach incidents, Cyberwire is a cybersecurity-focused news service which provides daily briefing of cybersecurity news. While these resources create reports that show the relative frequency of events, they do not provide comprehensive details about them that are supported by specific data measurements. Some resources categorize cybersecurity events by event type, attack pattern, and type of malware [SaFF019]. This in this research, this becomes part of the motivation for developing strong metrics capable of extracting meaningful features from network traffic.

As specific malware examples, one can think about the Slammer and Nachi worms that occurred in 2003, which used UDP ports for SQL activities and ping cascading respectively. The Nachi worm, supposedly a vigilante worm, was responsible to create 40% of the active sessions on the Internet in the late part of 2003 [Amor020]. A highly specialized malware was the Stuxnet worm from 2010, which targeted some facilities of Iran's nuclear industry and since then has metamorphosed and spread to other industries related to the energy sector. One shall recall that a worm works by finding a system with a vulnerability, replicating the malware program onto that system, and then executing remotely such malware. The usage of worms has declined significantly in later years because attackers have found that botnets, like the ones used for DDoS, are more powerful, which makes them an attack weapon of choice in cybersecurity [Amor020]. It is challenging for companies wanting to hire cybersecurity professionals to assess the knowledge, experience, and value the credentials of new individuals joining their workforce [Plat15]. This uncertainty is a problem for the cybersecurity industry causing either (i) that financial firms, government agencies (*i.e.*, the Federal Bureau of Investigation (FBI)), and telecommunications companies hire "ethical" hackers or (ii) rolling out educational programs to equip professional engineers with the latest hacking techniques, methodologies, tools, and tricks [Roze015a].

Cybersecurity firms and mass media report successful cyberattacks daily, which are growing in terms of complexity and volume ([ArGu021], [Gree021] and [MaCa021]). Diverse and recent cyberattacks have targeted and affected Canadians including: (i) The Equifax breach that exposed information about 19,000 Canadians, hundreds of thousands of Britons, and 145 million Americans [Desc020]; (ii) Canadian researchers becoming a target for spear-phishing

cyberattacks on COVID-19 research [Nowa020]; (iii) in August 2020, the Canada Revenue Agency (CRA) temporarily shut down its online services after confirming being hit by two cyberattacks that compromised 5,500 accounts [PaLi020]; (iv) the Chartered Professional Accountants Canada, which sets standards and guidance for 210,000 accountants, was subject to phishing attacks exposing personal and contact information of 329,000 individuals, including members and other stakeholders [Solo020]; (v) in July 2019, TransUnion confirmed that the personal data of 37,000 Canadians was compromised when someone illegally used a legitimate business customer's login [Bick019]; and (vi) in October 2019, the Canadian Internet Registration Authority (CIRA) reported that 71% of organizations experienced at least one cyberattack that impacted the organization in some way, including time and resources, out of pocket expenses, and paying ransom [Call019]. This list recalls a few impactful cybersecurity incidents impacting Canadians.

Many mobile app developers rely on third-party programs (*i.e.*, Google Maps or Facebook) to be integrated into their programs without understanding how these are using the data collected and whether they might cause potential privacy or security threats to the users. Moreover, many developers do a poor job of encrypting the data that comes from mobile apps [Roze015b]. Smartphones and tablets companies have not developed a default setting on devices to encrypt data from mobile apps or make it simpler for mobile app developers to do this on their own [Roze015b]. Even if mobile apps developers would care about security, their skill set to build in security is lacking [Roze015b]. In our data-reliant world cyberthreats take many forms, including troublemakers hijacking electronic equipment, hackers conducting cyberespionage, or globe-spanning cybercrime rings perpetrating bank fraud, to mention a few [Pate015]. Cybersecurity today involves much more than defensive measures. Organizations should also build secure foundations and anticipate security challenges like designing secure code, finding software vulnerabilities, putting management structures in place to deal with risks, and identifying possible threats from inside a company [Pate015]. Computer technology companies are making their best efforts to keep both the products placed on the consumers' hands and their online services safe from hackers. Even with all this, there are neither standardized metrics for gauging software security [Pate015] nor cybersecurity systems that incorporate cognition in their

operations. Appendix B contains a report of the diversity of computing systems in the modern cybersecurity ecosystem.

It is the growing demand for better CSS that motivates the efforts put in the research presented here. This thesis work attempts to follow closely both the **scientific** (*e.g.*, [Blak012]) and **engineering** methods (*e.g.*, [Koen003] and [Kins009]) by appealing to their *analytic* and *synthetic* nature respectively [Kins020]. Many contemporary scientists and engineers operate between these two extremes [Kins020]. This amalgam is worth highlighting because this research work explores deep aspects of mathematics and puts them into action, by engineering the implementation of systems, for augmenting network security with the ultimate goal of making this world a safer place for everyone. Henceforth, this research approach expands from engineering design to engineering for humanity ([Kins014] and [Kins020]).

1.3 Network Security

Social networking and content sharing have become ubiquitous and essential in our modern society. Certainly, our society cannot be conceived without the complex networking it has achieved so far. Social media is fast changing the public discourse in society and setting trends and agendas in very distinct topics (*e.g.*, environment, politics, technology, and entertainment). Social media feeds are effective indicators of real-world concerns and possible future reactions to a given event [AsHu010]. Society is dependant of the Internet since it requires many applications that are Internet-based [Yu014].

Data security in networking scenarios is certainly a major concern as there could be multiple vectors of attack. New *common network security adversaries* (CNSA) are motivated not only by economics or greed (financial gain), but also by ideology (political views) [Yu014]. From the insights of deep research, two adversaries are relevant: (i) Even though, in general, adversaries lack talent in their attacks sophistication they have demonstrated that security companies are also, in general, doing a poor job, and a new form of “hacking power” is potentially available to everyone. (ii) On the other hand, nation state sponsored security adversaries and espionage actors (motivated by politics and ideologies) are much more sophisticated for preparing and launching attacks (*e.g.*, adaptive, adaptive-persistent, deliberate,

goal-oriented, persistent, or undeterred) to undertake a given target. Many companies have been affected, losing a form of trade secret or intellectual property, by adaptive-persistent adversaries (APA) [Corm014].

The combination of CNSA and APA has eroded even more the degree of security of our digital society, which globally is falling short even though the best and brightest that know about security and adversaries are in the front line of defense [Corm014]. Due to the nature of the Internet and the lack of cyber laws, cyberspace has been a heaven for intelligent attackers. It is easy to launch attacks, but hard to identify the persons who commit the attacks and even harder to bring them to justice. One critical form of attack in cyberspace is the distributed denial-of-service (DDoS) attack.

1.3.1 Disruptions in Networked Computer Systems

Disruptions are unwanted phenomena reaching networked computer systems by either attacks or intrusions. Intrusion techniques have been the object of extensive research [LiJo997]. Anomalies, surreptitious scans and server nudges are attempts to compromise a system's integrity. About 90% of disruptions attempt to induce a 'buffer overflow' through digital entry points into computer hardware and operating systems thorough which superfluous amounts of data are written into a system's memory in an attempt to make it fail, opening it up to exploitation [Perk010].

The motivations are multiple ranging from: (i) Installing malicious software (intended to co-opt system resources, keystroke loggers, and to scan user information and passwords), on everyday computer systems; (ii) stealing computing power in high-performance computer systems, intellectual property and instrument designs in organizations, health data, or private communications data; and (iii) deploying highly complex cyberweapons, potentially designed for economic, political, or military intentions, in critical infrastructure [Perk010].

"*There is no sector that has been able to withstand this onslaught of intrusions*," advises Steven Chabinsky, deputy assistant director in the cyber division of the FBI in Washington DC. Protection of sensitive data is one of the most challenging tasks in computer systems [Perk010].

Most information technology (IT) professionals suggest ensuring that large or sensitive data stores are managed by a centralized IT team that can monitor and administer systems,

keeping a close watch over traffic and limiting access. Within this scheme, some Universities firewalls block millions of daily scans from Internet drive-bys looking for open communications ports according to cyberinfrastructure services. A "significant percentage" of these scans are likely to stem from "professionals in the employ of organized crime". Many common disruptions are simply ignored [Perk010]. Worse yet, many of them are undetected by current network security technologies and thus go unreported.

Some institutions use a battery of common but effective defenses (*e.g.*, pushing operating system and antivirus patches out to users, remotely monitoring network traffic, establishment of secure virtual private networks (VPNs) for encrypted communication and virtual machines (VMs) acting as hardware surrogates).

VMs allow for easy rolling back to put a hacked computer back online, secure professionally, monitor continuously, back-up and restore easily, and in the event of a breach contain intrusions effectively by operating on an isolated architecture [Perk010].

1.3.2 Cyber-Physical Systems

Technological progress has made possible significant advances in the computation and communication fields, and enabled the emergence of large, networked infrastructures (*e.g.*, agriculture, farming, food, transportation, health care, manufacturing, supply chain, and energy domains) [SuRH016].

Such large systems already include what are known as embedded systems, *i.e.*, computational systems designed to control and/or monitor a physical system. Although the concept of embedded systems has been in use for many years, the growing trend of interconnecting many physical and computational components to form large networks presents new challenges and requires novel approaches to design, control, and cybersecurity [SuRH016].

The concept of cyber-physical systems (CPS) was introduced in 2006 [KuBL994]. A CPS may be described as a typically large networked system, made of tightly interconnected physical and computational components, operating in a networked fashion. The history of CPS may be traced to the seminal article titled "As we may think" by Vannevar Bush in 1945 ([CFBG006] and [SuRH016]).

Significant advances in the fields of communications and network engineering;

computation, control and systems theory and engineering; information systems; Internet engineering; and sensor systems have led to the progression of the human-machine experience, thus paving the way for the evolution of the theory and hardware of CPS ([PFND006] and [SuRH016]).

CPS may be seen as similar to the Internet, but applied to the physical world. For example, IoT [BiVS013] may be considered as an enabler for CPS because CPS applications require efficient sensing and communication infrastructures. CPS are at the frontiers of the engineering and computer science fields because of the aim to combine the most recent advances in both disciplines. The systems integration approach is therefore a central element to the CPS concept. Current challenges at the frontier of both fields include architectures, interoperability, networked control, standards and test procedures, verification and validation, and security. CPS-enabled critical infrastructures (*e.g.*, energy, transportation, or smart cities) may possess the promise of solutions to the grand challenges facing the engineering community in the twenty-first century ([CRDB011] and [SuRH016].

Only considering the technical aspects is insufficient, especially when humans are expected to use and be impacted by the designed CPS. A system can operate perfectly from a technical point-of-view, but if its users are unable to understand how to interact with it or are not convinced of its usefulness, the system may never reach the point of fulfilling the intended purpose. Thus, the social aspects of such systems must be considered in the design process [SuRH016].

1.3.3 Cyber-Physical-Social Systems

An emerging, yet challenging, frontier for CPS applications is the inclusion of the social aspect in engineering. The end-user of a critical engineering infrastructure determines the utility of that domain, and any advance should also increase the quality of life of the end-user. A “smart” infrastructure includes an active end-user (sometimes described as a “prosumer”) with a notably different role in participation than in the past [GuYS012]. The active end-user, enabled with information in *real-time* or *near real-time* and the ability to make decisions, is no longer a passive participant in the control and operation of the critical infrastructure. An empowered end-user, with a hitherto unprecedented level of information and control, is a paradigm shifting

concept such as residents of "smart" homes in the electricity domain controlling their energy usage to save money and provide grid ancillary services. To understand, model, simulate, develop, build, test, analyze, and enhance these futuristic manifestations imposes a fundamental requirement of the consideration of the social (and societal) aspect of CPS. The human-centric CPS, which marks the next generation of CPS, is called the cyber-physical-social system (CPSS) ([SBRS007] and [SuRH016]). A CPSS integrates computation, physical components, and human cognition to achieve socially aware advancement in the operation of critical infrastructures and their interdependencies [SuRH016].

1.4 Research Questions Posed A Priori

The significance of networked computer systems in our information era is categorical. Our society is absolutely dependent on services provided by networked computer systems, becoming more dependent on them as time progresses. This has become evident in critical times as society has to rely on virtual environments to function when there are unexpected limitations. New digital services are created on a daily basis with the capacity of pragmatically changing and impacting everyone. Nevertheless, safeguarding networking environments is certainly the greatest challenge in cybersecurity. Within cybersecurity, the aim of maintaining a given network operational and accessible to legitimate users, the appropriate and accurate detection of DDoS attacks is of fundamental importance ([AaAr013] and [Kasp014]). The literature provides a large number of examples in which distinct metrics are used to extract features of network traffic that could lead to detection of DDoS attacks. Considering the complex cybersecurity scenario, the *main research question* is: Can *new multi- and poly-scale-based metrics* be helpful in deriving a *set of features* capable of detecting DDoS attacks accurately and effectively? Can a deep learning (DL) architecture, from the feature extraction perspective, utilizing adaptive resonance theory (ART) as the pragmatic machine learning approach offer a high classification performance when processing a polyscale feature vector?

Secondary research questions stemming from the previous one are: From the introduced multiscale-based metrics, which are the per case relative merits for analyzing data and signals obtained in networked computer ecosystems for detecting disruptions by DDoS? Which are the relative performances of the multiscale-based operators considered in the metrics design? If these

multiscale-based metrics are collectively aggregated or processed further as poly-scale-based metrics, can these be considered robust metrics to characterize DDoS disturbances found in data streams in networked ecosystems?

The *thesis question* in the research presented is:

> *Can new multi- and polyscale-based metrics as a set of features, capable of enabling arbitrary operators, detect DDoS attacks through adaptive resonance theory based ANNs with a high classification performance (accurately and effectively) considering a time-multifractality approach?*

1.5 Thesis Statement

This thesis addresses the development of an *early anomalies detection system*, in Internet/network traffic, supported by both *polyscale analysis based robust metrics* and pragmatically focused on an unsupervised machine learning model based on ART.

The focal application herein is the detection of departures, from what is perceived as the expected behaviour from clear Internet/network traffic, through *polyscale analysis* based methodologies that allow the implementation of arbitrary operators.

The multiscale analysis methodologies of reference, the *variance fractal dimension* and the *variance fractal dimension trajectory*, are extensively exploited to harness their analysis power and through them a new methodology, known as *multiscalors*, has been posed through this research effort. Robust operators are used through *multiscalors*, which provide a set of relevant features, a vector, that are channelled pragmatically to ART machine learning models.

The relevant metrics describe a DDoS cyberattack with diverse degrees of multiscale resolution in order to build up incremental learning within the ART machine learning models in the procurement of the real-time classification of DDoS cyberattacks causing anomalies in Internet traffic.

This research considers *multi-* and *polyscale* analysis inspired modelling for enhancing cybersecurity: (i) The development of an *early anomalies detection system* for a cognitive computing engine system, which proposes the supporting architecture; (ii) a signal analysis methodology, *multiscalors*, capable of extracting features from a signal analogous to perception stages in neurological systems; and (iii) a learning process based on modeling of the short and

long-term memories of the human brain represented through ART implementations, which provides powerful unsupervised algorithms applied in DL with a novel viewpoint [SiPK017], from the feature extraction perspective.

1.6 Statement of Objectives of the Research

This research effort presented in the area of network security involves the following stages: (i) *Access* to a relevant dataset that contains a documented DDoS attack; (ii) *Insight development* through the deep inspection of the DDoS dataset and deriving packet flows for understanding the dynamics of a concurrent attack; (iii) *Implementation* of robust information theoretic based (ITB) metrics (variance fractal dimension (VFD) and variance fractal dimension trajectory (VFDT)); (iv) *Processing mechanism isolation* of the VFD and the VFDT in an effort to harness the multiscale analysis power so that alternative and arbitrary operators can be used in the multiscale domain; (v) *Operators definition and implementation* through the previous processing mechanism for achieving robust metrics; (vi) *Compare* the qualitative performance between the metrics implementation in this research; (vii) *DDoS attacks feature characterization* through the multi- and polyscale metrics; (viii) *Preparation of feature vectors* and their proper representation for further processing in machine learning models; (ix) *Implementation* of ART based artificial neural networks (ANNs) models pragmatically as a *detection*, *analysis*, and *classification* methods. The metrics prepared are then amalgamated into a feature vector, characterizing disruptions caused by DDoS attacks in networked computer systems, and then fed into the ART models. These *classification* models associate DDoS disruptions to discernible collections, *classes*; and (x) *Examination* of the machine learning models classification precision as well as comparing their relative merits.

1.7 Organization of the Thesis

This thesis consists of the subsequent construction: (i) The background theory is covered in chapters II to IV; (ii) chapter V is dedicated to experiment design; (iii) chapter VI describes the experimental results and discussion; (iv) conclusions about the research conducted in this thesis are presented in Chapter VII; and (v) detailed appendices are also available.

The first background chapter, Ch. II, introduces the fundamental network security concerns that are relevant to this research. Of special importance is the description of distributed denial-of-service as a focal point in this work. Overview of DDoS attacks is discussed along with the malicious networks required to launch them. Concealment aspects are also mentioned, and different detection methods rooted in information theory are included.

Chapter III presents feature extraction as a whole. This chapter describes topics like the traffic sensing of Internet/network traffic, signal conditioning, metrics design capable of detecting network anomalies, concepts of monoscale and multiscale analysis, the implementation of the VFDT as a precursor to the new multiscale methodology, *multiscalors*, introduced in this research, the selected multiscalor operators, secondary signal analysis methodologies applied to the multiscalor components, and the preparation of the extracted features so that the selected machine learning models are capable to process them for successful classification of anomalous events.

Chapter IV encompasses the cornerstone machine learning models pragmatically chosen and utilized for classification of Internet traffic in this research. These models are based on the adaptive resonance theory implementations, ART1 and FuzzyART. A complete description of the various generations of ART is also surveyed and explored in detail.

The design of experiments is addressed in Ch. V. It begins describing the computing resources utilized to carry out the dataset analysis. The integration of packet count and packet length is described. Also, the isolation of attack traffic and attack flows is fully covered. Validation of the VFD algorithm is carried through white noise. The signal processing pipeline involving signal conditioning, analysis, feature extraction, and classification through ART is provided. The selected operators used as multiscale metrics are presented. Secondary signal analysis methodologies applied to multiscalors components are also introduced in an effort to create composite and more robust features. The feature preparation in a vector form so that it can be processed by the ART approaches is covered.

The experimental results and discussion provided in Ch. VI depicts the practical implementation of the research conducted. Contributions from the packets count and data rate in single attack flows to the overall attack traffic are analysed. The results of analysing known white noise signals through the VFD are also covered. A test case for preparing features through

denoising, nonlinear filtering, and quantization with Internet/network traffic is presented. The results of the multiscalor operators and secondary processing methodologies are shown and commented. The preparation of the feature vector and the classification results are also discussed.

The conclusions of the research conducted are finalized in Ch. VII. Appendices containing complementary information are also included. These appendices provide information about: IIoT, the dataset containing DDoS attacks used in this study, signal processing definitions, taxonomies of computer systems disruptions, ensembles of classifiers, diversity of computing systems in cybersecurity, the geometric interpretation of FuzzyART learning, histogram binning, malicious networks, DDoS attacks detection and software defined networking, and computational intelligence approaches. Furthermore, extensive and detailed outcomes are included, as part of these appendices: Results of selected primary operators applied through multiscalors, results of selected secondary operators applied to variance multiscalor, results of selected secondary operators applied to skewness multiscalor, results of synthetic classes detection through ART1, results of selected secondary operators applied to skewness multiscalor, results of synthetic classes detection through FuzzyART, confusion matrices for ART1 performance, and confusion matrices for FuzzyART performance.

CHAPTER II

NETWORK SECURITY AND

DISTRIBUTED DENIAL-OF-SERVICE

2.1 Distributed Denial-of-Service

A cyberattack usually consists of several stages such as reconnaissance, DDoS, man in the middle (MITM), elevation of privilege, data tampering, among others. Attackers first gather information about the target system during the reconnaissance phase to identify network topology, software versions, and critical targets to attack [HXCL014]. After attackers gain knowledge about the system, they plan further attacks by researching known vulnerabilities against the detected software versions; possible attack vectors and pivot points for bypassing firewalls and intrusion prevention systems (IPSs); and options for removing evidence after the attack, such as deletion and manipulation of system logs [ZKHC019].

Distributed denial of service has been the most prominent attack in CPSs over the last decade. Myriads of new strategies and approaches have been proposed to defend against different types of DDoS attacks. DDoS attacks have become a weapon of choice for hackers as well as for cyber terrorists [DaVS020]. DDoS attacks require an especially sharp real-time capability for analysis. Additionally, despite the obvious wave-like effects that are experienced after a DDOS attack, only subtle indicators are generally present for analysts in advance of such attacks [Amor020].

Based on various techniques such as cloud computing, software defined networks (SDNs), backbone web traffic, big data strategies, and data science, DDoS attack detection can be categorized into filtering mechanism, routers function, network flow, statistical analysis, and machine learning [DaVS020].

A demonstration of the first documented cyberattack that could destroy a 27 tons diesel engine coupled to a generator was conducted, as a proof of concept, by the Department of Homeland Security, at the Idaho National Laboratory (INL) in 2007. This cyberattack is now known as Aurora to describe when circuit breakers are opened and closed, resulting in an out-of-phase condition that can damage alternating current (AC) equipment connected to the grid. Communication protocols used by control systems and supervisory control and data acquisition (SCADA) systems vary based on the design of utilities. The most common protocols are DNP, Modbus, IEC 60870-5-103, IEC 61850, Telnet, QUIC4/QUIN, and Cooper 2179. Compromising any of these protocols would allow a malicious agent to control systems outside utility operations. Communications protocols compromise allows access to devices and the ability to compromise their associated passwords to infiltrate a system. An electricity system comprises generation resources, transmission facilities, distribution facilities, and participation within an energy marketplace. A compromised power grid, through a cyberattack, hinders the reliability of its operation is in question and the interconnection of resources and execution of market transactions becomes highly disrupted and ultimately stopping [SBWH13].

Major cyberattacks, in its majority DDoS, have been detected in ICSs, CPSs, and CPSSs for at least a decade. Some of the most known cyberattacks are described as follows. In 2010, Stuxnet attacked nuclear enrichment centrifuges in Iran, causing severe equipment damage. In 2012, Shamoon worked against national oil companies in South Arabia and Qatar. In 2014, German Steel targeted a metallurgic mill in Germany, and Havex carried out an espionage campaign focusing on energy, aviation, pharmaceutical, defense, and petrochemical sector targeting victims primarily in the United States and Europe. In 2015, BlackEnergy3 (BE3) attacked the power grid in Ukraine. In 2016, the Mirai botnet was used in some of the largest and most disruptive DDoS attacks, and Sandworm struck the Ukrainian power grid. In 2017, HatMan (also known as TRITON and TRISIS) affected Triconex controllers by modifying firmware to add additional programming, Crash Override (known as Industroyer) also targeted Ukraine's power grid, Palmetto Fusion attempted intrusions of USA energy utilities, BrikerBot attempted to permanently destroy insecure IoT devices, and Dragonfly 2.0 targeted the energy sector in Europe and North America [ZKHC019].

Based on the attackers research and the ultimate goals of the attack, several types of

attacks may be launched. DDoS attacks against either ICSs or CPSSs may aim to disrupt communication between the SCADA master and slaves, which could cause the SCADA master to lose control of local control systems and actuators. Privilege escalation may be needed to access the low-level hardware on a system or to read and write to protected system files; necessary escalation of privileges can be achieved using zero-day attacks and known vulnerabilities in the operating system and software used. Interception of commands and sensor data can be performed using an MITM attack, while data tampering and false data injection attacks go a step further to modify sensor data in transit in order to mislead the monitoring systems and operators while the attack is in progress. Data tampering could alter SCADA masters commands to cause the actuator to actuate inappropriately; it could alter the feedback process data to manipulate the control; and it also could alter data in a data historian to modify the operation log and system control-related data to obfuscate the details of the attack, which misleads the defender in postattack analysis [ZKHC019].

Following the math on botnet-originated DDoS attacks produces frightening conclusions from the perspective of national critical infrastructure protection. If a bot running on a home PC, for example, can originate one million bits per second or 1 Mbps, then doing the math on how big an aggregate DDoS attack might be for botnets of different sizes can be inferred. Botnet data generation volumes can grow quickly. Since botnets in the last years have ranked in millions of bots, then a million size botnet can generate 1 Tbps, which is more than enough to knock off a large size enterprise gateway (typically in the 10 Gbps range) or cause an Internet backbone (100 Gbps) to get congested. For any essential network that provides a service to society that cannot be replaced, or whose removal could lead to loss of safety or lives, becoming the target of a DDoS attack could be substantially damaging. Given the relatively modest work required to build a ten thousand-member botnet, it becomes much too easy to interrupt infrastructure. Consider that with the IoT, billions of poorly secured devices have been scattered across the global Internet. If botnets begin to efficiently harness the attack capacity of these devices, then DDoS attacks of immense strength might be produced. This provides a glimpse of the potential for DDoS attacks that the world has not seen is real. Furthermore, if a significant series of concurrent DDoS attacks were to be initiated at the same time to the same set of targets, it is unclear if the associated volumes could be stopped [Amor020].

A challenging problem in networks arises from their characteristics of packet switching, variable bit-rate and on-demand bandwidth. Approaches to address this problem require knowledge not only of the statistics of the source, but also of the rules for assembling the packets in order to monitor traffic.

Intrusion detection systems (IDS) detect unauthorized access to the system. There are three types of IDS: Signature-based, anomaly-based, and hybrid. *Signature-based IDSs* are developed to detect known attacks using their documented behaviour. This class of IDS is very effective for known attacks with low false alarm rates but are not able to detect zero-day attacks since the IDS is not yet aware of this behaviour. *Anomaly-based IDSs*, on the other hand, model the normal behaviour using data mining techniques or machine learning algorithms and report deviations from normal behaviour as an anomaly or potential attack. They are customized to the normal behaviour of each system to detect attacks, including unknown attacks, making it difficult for attackers to learn the capabilities of IDSs, further complicating attackers ability to launch undetectable attacks. The very nature of this makes anomaly-based IDSs result in a high number of false positives [BuGu016]. The *hybrid IDS* is a combination of signature-based and anomaly-based detection; this approach combines the accuracy of signature-based approaches for known attacks with the generalizability of anomaly-based systems [ZKHC019]. Anomaly-based IDS over signature-based has a better detection accuracy, which favours the detection of unseen attacks, but at the expense of a lot of false identification of unusual activities as anomalous [HOHR015].

Data-driven, hybrid IDSs, are promising approaches to enhance industrial control systems (ICSs) cybersecurity and the situational awareness of defenders. Cybersecurity is defined as: "*Strategy*, *policy*, and *standards* regarding the *security* of and *operations* in *cyberspace*, and encompass[ing] the full range of *threat* reduction, *vulnerability* reduction, deterrence, international engagement, *incident response*, resiliency, and *recovery* policies and activities, including computer *network operations*, information *assurance*, law enforcement, diplomacy, military, and intelligence missions as they relate to the *security and stability* of the global information and communications infrastructure." This definition is obtained from the National Initiative for Cybersecurity Careers and Studies (NICCS) [Nati018].

A recent survey of cybersecurity research using data mining and machine learning

algorithms identified the following methods as effective in cyber-attack detection: Clustering, decision tree (DT), genetic algorithms (GAs), naïve Bayes, support vector machine (SVMs), ANNs, and random forest (RF) ([BuGu016] and [ZKHC019]).

Automated generation of attack trees is used in cybersecurity analysis to give an analyst a view of all the ways in which an attack can be carried out [BLNS020]. Attack trees can consider DDoS. However, this requires a library of attack templates, and an abstract model of the network architecture under attack [DaVS020].

2.1.1 Overview of DDoS Attacks

The Internet has become an important part of our society in numerous ways, such as in economics, government, business, and daily personal life. An increasing number of critical infrastructures (*e.g.*, smart grid and air traffic control) are managed and controlled via the Internet ([KPHD015], [KPBH015] and [ÖzBr015]), in addition to traditional infrastructure for communication. Today's cyberspace is rife with cyberattacks, such as DDoS, information phishing, financial fraud, email spamming, among the most known ([KSSS014] and [Yu014]).

Cyberattacks on communications networks can be categorised into either passive (*e.g.*, eavesdropping or traffic analysis) or active attacks (*e.g.*, spoofing or DDoS) [DaVS020]. Among various cyberattacks, denial-of-service (DoS) attack is a critical and continuous threat in cybersecurity. DoS attacks are implemented by either forcing a victim computer to reset or consume its resources (*e.g.*, access to application programming interfaces (APIs) [BaAZ014], CPU cycles, memory or network bandwidth ([BeDe014] and [BKBK014])). Hence, the targeted computer no longer provides its intended services to legitimate users. When the DoS attacks are organized by multiple distributed computers, it is called DDoS attack, which is a popular attack method in the cyberspace [Kasp014]. Network security branches into three categories: Confidentiality, availability and integrity. DDoS attacks belong to the availability category [Yu014].

DDoS continues to plague the availability of online services. As cybersecurity problems, DDoS are evolving and non-stationary. The constant deployment of new services and protocols adds to Internet traffic additional non-stationarity. Attack patterns in DDoS shift as new protocols and applications are introduced, further compounded by burstiness, seasonal variation,

and diversity of network traffic across varying timescales. When it comes to cyberdefense against DDoS, it is difficult to apply machine learning-based techniques and defenses in practice. Cyberattacks and anomalies have measurable consequences and symptoms which allow a skilled analyst to infer new signatures for detection by misuse-based classifiers, conversely unseen attacks may only be defended against after-the-fact. It has been long-hoped that anomaly-based detectors would surpass the element of surprise by making effective use of statistical measures (monoscale analysis) [SiRP020]. This research in particular, goes beyond the use of statistical measures and centres on the search of long range dependencies by characterizing Internet traffic with multi- and polyscale measures. Long-term expectations in cybersecurity are to augment what existing misuse-based solutions can provide, by automatically alerting, recording and controlling what are believed to be illegal system states [SiRP020].

The idea of denial of service appeared in the digital world in 1984 from the research on operating systems [Glig984]. With the booming of the Internet in the middle of the 1990s, DDoS attacks are getting more and more familiar to general public ([Fu011] and [PeLR007]). It is reported that there were only six DDoS related attacks in 1988. The first well-documented DDoS attack appears to have occurred on August 1999, when a DDoS tool called 'Trinoo' was deployed in at least 227 systems, to flood a single University of Minnesota computer, which was knocked down for more than two days [DaVS020].

In 2000, well-known web sites, such as CNN, Amazon and Yahoo, became the targets of DDoS attacks, and the attack rate was around 1 Gigabit per second, Gbps. In 2007, a DDoS attack rate reached 70 Gbps. In 2013, the peak of the biggest DDoS attack reached 300 Gbps [Yu014]. Also, the first quarter of 2013 registered the average attack bandwidth exceeded 48.25 Gbps [AsLa014]. In 2016, the largest DDoS attack was carried out by the Mirai botnet based on IoT, which compromised devices targeting the domain name system (DNS) provider Dyn, affecting many popular sites including Twitter, Reddit, Spotify, GitHub and the New York Times [Losh016]. In 2018, GitHub was taken offline briefly by a 1.35 Terabit per second (Tbps) DDoS attack, confirmed and mitigated by Prolexic Technologies, the DDoS mitigation subsidiary of Cambridge, MA, company Akamai. Also in 2018, Arbor Networks confirmed a 1.7 Tbps DDoS attack requiring just one line of Python code against vulnerable memcached (distributed memory object caching systems intended to speed up dynamic web applications)

servers [Hell018]. In 2019, 8.4 million DDoS attacks were detected, every minute 16 DDoS attempts took place. The most powerful DDoS attack recorded in 2019 held a bandwidth of 622 Gbps. However, as noted by Netscout, such attacks can generally be considered "overkill" and would most likely draw the attention of law enforcement. Thus, attacks are now generally becoming "stealthier" between 100 to 200 Gbps. The number of DDoS campaigns beyond the 300 Gbps mark has dropped [Osbo020]. Imperva reported volumetric DDoS attacks on the following occasions for 2019: January 2019 sustained a 500 million pps (packets per second) attack in layers 3/4, and April 2019 peaked at 580 million pps in layers 3/4. This last incident is considered the largest DDoS attack by packet count to date [Cran019]. Another DDoS event in 2019 involved a botnet that coordinated 402,000 different Internet protocol (IP) addresses, directed a peak flow of 292,000 requests per second (RPS) in layer 7, and a nonetheless interesting fact, it also lasted 13 days [Simo019].

For 2020, the DDoS trends are described next. DDoS attacks are anticipated to reach 14.5 million by 2022, according to 2017 data from the Cisco Visual Networking Index (VNI). DDoS attacks, both in size and number, have been on a downward trend since the FBI shut down 15 of the largest DDoS-for-hire websites in December 2018. DDoS attacks can represent up to 25 percent of a country's total Internet traffic while they are occurring. China and the USA ranked as the top two targets for DDoS attacks in Q2 2019, with 63.8 percent and 17.5 percent of the attacks, respectively. Neustar discovered in its DDoS attack research that the increasing trend of strategic, "low-intensity incursions" that degrade the performance of servers over time. Using these lowball attacks enables hackers to carry out longer attacks that fall below the level of intensity that would trigger DDoS defenses. A significant number of attacks feature over four vectors. The number of IoT devices that are estimated to exist by the end of 2020 is 20.4 billion, according Gartner. IoT devices, notorious for lacking any real IT security or cybersecurity measures, are vulnerable to DDoS attacks. According to Bulletproof, a DDoS attack could cost up to $120,000 USD for a small company or more than $2 million USD for an enterprise organization. A10 Networks tracked more than 23 million DDoS weapons (infected computers, IoT devices, and servers). According to Kaspersky, in 2019 although DDoS attacks are down, a clear increase in politically-motivated DDoS attacks was registered. Akamai indicates that financial organizations are seeing a rise (800 attacks between December 2018 and May 2019,

which is 40% of the attacks during this time) in DDoS attacks. IBM X-Force indicates that more than 80% of all observed activity from Mirai botnet variants in 2019 targeted the media/information services and insurance industries. From $2.4 billion USD in 2019, it is estimated that the DDoS protection and mitigation market would reach $4.7 billion USD by 2024 [Cran020].

Scientists in Synopsys state that performing amplified DDoS attacks using memcached servers is trivial because this service was never intended to be connected to the public Internet but originally designed and implemented for an internal, benign environment, so it actually responds to requests without requiring authentication and its implementation of user datagram protocol (UDP) is flawed returning a large number of bytes when queried with a small number of bytes causing an amplification as much as 50,000 times the request [Hell018].

All occurrences of DDoS attacks can be neither detected nor documented, but the available information about the DDoS attacks that have been detected and documented indicates that DDoS remains one of the major threats for network security [AsLa014]. DDoS attacks are getting highly sophisticated with the potential to be launched from any layer (application, protocol, session, transport, network, data-link, and physical) of the open systems interconnect (OSI) model [Kuma016], and diverse types targeting distinct weaknesses [Java018]. DDoS detection through IDS, either network or host based, is very useful for collecting forensic evidence that may be used in legal proceedings if the attacker is prosecuted [BAUM014].

Despite all the efforts from industry participants and academia, DDoS attack is still an open problem. Some of the essential reasons for this passive situation are: (i) The design of the ARPARNET network lacked a security focus. The Internet originated from this private network, ARPARNET. As a private network, there were very limited security concerns in the original design [PeLR007]. This private network became a public network in the 1990s, and now many applications have become an essential part of the Internet. Security patches have been developed and installed to circumvent the inherent vulnerabilities; however, the effectiveness of these efforts is sometimes limited. For example, the Internet was designed stateless, therefore, a receiver has no information about which routers a received packet went through. Hence, it is easy to perform source IP spoofing; (ii) Internet is the largest man-made system in human history. Cyberspace is huge, complex, and stays in an anarchy status; (iii) Cyber attackers are

enjoying one incredible advantage of the cyberspace: It is hard for defenders to technically identify attackers. Moreover, there lacks international laws or agreements among nations to bring cyber criminals to justice who commit crimes in one country but are living in other countries; (iv) Hacking tools and software are easy to obtain. An attacker may not need profound knowledge of networking or operating systems to initiate a cyberattack [Yu014].

Distinct instances of DDoS attacks have been scrutinized and it is found that the attack can be mitigated by one of these three approaches or defense mechanisms, namely, *attacker-end* approach, *victim-end* approach, and *in-network* approach, depending on their locality of deployment. The existing detection approaches can be categorized into statistical, soft computing, clustering, knowledge-based, and hybrid. These approaches can also be classified as supervised or unsupervised based on the type of dataset. In the evolution of IDSs, anomaly-based detection is more preferred than signature-based detection [DaVS020].

2.1.1.1 How to Launch DDoS Attacks

DDoS attacks can be launched in two forms: (i) A system is targeted by sending one or more carefully crafted packets (*e.g.*, "ping-of-death" attack causes some operating systems to crash, freeze, or reboot), which are designed based on the vulnerability of the victim; and (ii) Using a large amount of traffic to exhaust the resources of a victim, such as network bandwidth, computing power, or operating system data structures, among others. Henceforth, the quality of service of the victim is significantly degraded or disabled to its legitimate clients [Yu014]. The most well-known DDoS attacks are transfer control protocol (TCP) TCP-SYN flood, Internet control message protocol (ICMP) ICMP/UDP flood attack, ping-of-death, Smurf, process table, UDPstorm, syslogd, mailbomb, and Apache2, which consume the uplink bandwidth or server bandwidth [BoAy013].

Launching an effective DDoS attack requires cyber attackers to firstly establish a network of computers, which is known as a *botnet* or *army*. The individual controlling a botnet is called *botmaster* or *botnet owner.* Attackers take advantage of various techniques (referred to as *scanning techniques*) to find vulnerable hosts on the Internet to gain access to them ([PeLR007], [SCGK011], and [CCGP010]). The next step for the attacker is to install programs (known as *attack tools*) on the compromised hosts. The headquarters of a botnet is called *command and*

control (C&C) server. The command and control server communicates with its bots for updating the attack tools, and issuing attack orders [Yu014].

Sustaining C&C servers from detection may require botnet programmers: (i) Setting up intermediate nodes as *stepping-stones* between the C&C server and bots, and (ii) encrypting the messages of their communication with cryptographic techniques [Stin006]. Avoiding evictions may require botnet programming techniques like IP flux or domain flux, to conceal their C&C servers [Yu014].

Two different DDoS attack classes: Typical DDoS attack and distributed reflection denial-of-service (DRDoS) attack. Unlike typical DDoS attacks, a DRDoS attack network consists of C&C servers and reflectors. In a DRDoS attack bots, led by C&C servers, send a stream of packets with the victim's IP address as the source IP address to uninfected machines (*reflectors*). A variation of a DDoS attack in cloud computing is the *economic denial-of-sustainability* (EDoS) attack [SqAS011] or the *fraudulent resource consumption* (FRC) attack [IdTJ013]. DDoS defense can be classified into three categories: Detection, mitigation and traceback [Yu014].

2.1.1.2 Challenges in DDoS Related Research

Understand the cyberspace theoretically and deeply. The American National Research Council proposed a new research field as network science in 2006 for advancing knowledge of networks and networking. The majority of current dominant Internet modelling is based on the random graph model proposed in 1959, which is far before the birth date of the Internet and the Web. Recent observations indicate that there is a great discrepancy between the random graph based models and reality. Power law (usually represented by the Zipf or the Pareto distributions) has been found to be pervasive in nature, for example economics and man-made systems, such as individual income among a group of people, or word frequency in a language. Researchers have also found many phenomena in cyberspace that follow power-law relationships (*e.g.*, popularity of web pages follows the Zipf distribution [BCFP999] and the size of web documents follows the Pareto distribution [CrBe997]) [Yu014].

Understand our cyber opponents in a correct way. It is hard to collect, or share cyberattacks data from industry and government agencies. Cyber opponents are only partially

defined, or the information is misleading. Understanding opponents in time and in an appropriate manner is mandatory [Yu014].

Solid consideration of the previous two aspects, effective and efficient strategies to defeat cybercrimes, including DDoS attacks, can be designed. Nevertheless, this aspect is very challenging.

2.2 Baseline for Anomalies

Anomaly detection requires defining a *baseline* for the network behaviour [AsLa014]. This baseline, supported by specific traffic features, is a depiction of the acceptable network behaviour. The traffic features are further fed into a classifier, part of a network security engine (NSE), which can assist in making automated decisions and triggering specific threat mitigation and defense events. The baseline can be set by the information provided by the features that characterize the normal traffic.

2.3 Summary

The concepts related to network security and DDoS have been provided. DDoS attack variants have also been addressed. DDoS attack launching techniques are covered extensively as well as research challenges for DDoS. The next chapter delivers discussion about Internet traffic, its preparation through signal conditioning and subsequent stages for its analysis. A novel and advanced signal processing methodology, *multiscalors*, is presented.

CHAPTER III

MULTISCALORS BASED FEATURE EXTRACTION

Much of the past modeling, analysis and synthesis of autonomous intelligent systems, autonomic systems, cognitive systems, and natural cognitive processes have been conducted using *monoscale* metrics that had yielded features later used in machine learning approaches [Kins011]. The proposed cognitive system here uses the fundamentals of both *multiscale* and *polyscale* analysis for extracting useful features capable of providing relevant information content for the detection of cyberthreats as DDoS.

Cognition is the ability for a system or systems to monitor, record, sample, test, and be *aware of the surrounding environments* and then to adapt, modify, or change the system to improve the quality-of-service (QoS), including learning from past experiences [Bull014].

In this research, Internet/network traffic is considered for cognitive analysis through a pragmatic set of subsystems. In this chapter, a clear walkthrough of all the signal analysis methodologies required for feature extraction is provided. This digital journey includes: The sensing of the Internet/network traffic, the context about monoscale analysis and its limitations, the critical importance of multiscale analysis for sifting information available in long range dependencies, the novel methodology "*multiscalors*" that allows arbitrary operators and signal analysis methodologies to be functional in the multiscale analysis context, the statistical moments used as multiscalors operators and the secondary signal analysis methodologies further applied to the multiscalors components, and the preparation of the features to be used subsequently by machine learning stages.

3.1 Internet Traffic

Will Leland and Daniel Wilson [LeWi991] present a preliminary analysis of unique high-quality data and comment in detail on the presence of "*burstiness*" across an extremely wide

range of time scales: Traffic "*spikes*" ride on longer-term "*ripples*," that in turn ride on still longer term "*swells*," and so forth. These self-similar patterns at different scales in Internet traffic are due to the presence of long-range dependencies. This self-similar or fractal-like behaviour of aggregate Internet traffic is very different from both conventional telephone traffic and from currently considered formal models for packet traffic (*e.g.*, pure Poisson or Poisson-related models such as Poisson-batch or Markov-Modulated Poisson processes [HeLu986], packet-train models [JaRo986], and fluid flow models [AnMS982]), which places a strong requirement for a new traffic modeling perspective. The term "self-similar" was coined by Mandelbrot ([Kins020] and [LTWW994]).

Internet data shows that the generally accepted argument for the "Poisson-like" nature of aggregate traffic, namely, that aggregate traffic becomes smoother (*less bursty*) as the number of traffic sources increases is unrealistic. In fact, using the degree of self-similarity as a measure of "complexity," it is observed that the burstiness of Internet traffic typically intensifies as the number of active traffic sources increases, which contradicts commonly held views [LTWW994].

The studies about self-similar processes by Benoit Mandelbrot [Mand969] are later extended by Murad S. Taqqu and Joshua B. Lévy [TaLe986], based on aggregating many simple renewal reward processes exhibiting inter-renewal times with infinite variances. These studies focus originally in an economic framework involving commodity prices and it is also applicable in the context of high-speed packet traffic like the case of Internet traffic [LTWW994].

Slowly decaying variances, long-range dependence, and a spectral density obeying a power-law are different manifestations of the underlying stationary process X. In their research about Internet traffic, tested through Ethernet local area network (LAN) traffic, Will Leland and Daniel Wilson conclude that: (i) Internet traffic is statistically self-similar, irrespective of when during the four-year data collection period 1989-1992 the data was collected and where it was collected in the network, (ii) the degree of self-similarity measured in terms of the Hurst parameter H is typically a function of the overall utilization of the Ethernet and can be used for measuring the "burstiness" of the traffic (namely, the burstier the traffic the higher H), (iii) major components of Ethernet LAN traffic such as external LAN traffic or external TCP traffic share the same self-similar characteristics as the overall LAN traffic, and (iv) the packet traffic

models still considered in the literature are incapable of capturing the self-similarity properties of Internet traffic [LTWW994]. There are novel research works using self-similarity measures to assess network flows through the Hurst exponent [LXKX020]. It has been found that normal OpenFlow traffic usually has a low degree of self-similarity due to the unique SDN/OpenFlow architecture, but when subject to saturation attacks has a higher degree of self-similarity [LXKX020]. Hence, self-similarity fluctuations are used for anomaly detection.

The ample indications of the impact of the self-similar nature of Internet packet traffic for engineering, operations, and performance evaluation of high-speed networks elaborated by Will Leland and Daniel Wilson in their study: (i) Source models for individual Internet users showing extreme variability in terms of interarrival times of packets (*i.e.*, the infinite variance syndrome), (ii) commonly used measures for "burstiness" such as the index of dispersion (for counts), the peak-to-mean-ratio, or the coefficient of variation (for interarrival times) becoming no longer meaningful for self-similar traffic and becoming replaced by the Hurst parameter (or other methodologies capable of self-similar behaviour analysis), (iii) the nature of congestion produced by self-similar network traffic models differing drastically from that predicted by "standard formal statistical models" displaying a far more complicated picture than has been typically assumed, and (iv) first analytic results showing a clear distinction between predicted performance of certain queueing models with traditional input streams and the same queueing models with self-similar inputs [LTWW994], seem to be overlooked by some.

Henceforth, the research conducted in this thesis considers methodologies capable of analyzing Internet traffic that are capable of capturing its self-similarity properties, as is the case of *multiscalors* that are fully described at the end of this chapter.

3.2 Internet/Network Traffic Sensing

Internet traffic sensing is a computer networking term for intercepting data packets crossing or moving over a specific computer network. Malicious agents can also use traffic sensing techniques to steal data that is being transmitted over a network [Tech017b].

One type of Internet sensing is packet filtering, in which filters are applied over network nodes or devices where data is captured. Conditional statements determine which data is captured (*e.g.*, a filter might capture data coming from ABC route and having W.X.Y.Z as an IP

address) [Tech017b].

Instead of filtering a specific portion of a packet, complete packets can also be captured. The full packet includes two things: A payload and a header. The payload is the actual contents of the packet, while the header contains extra information, including the packet's source and destination address [Tech017b].

3.3 Signal Conditioning

Once a packet is captured through a *packet analyzer*, it is stored temporarily so that it can be dissected. The packet is inspected to help diagnose and solve network problems and determine if network security policies are being followed [Tech017b].

A packet analyzer (*aka* a sniffer, network analyzer, or protocol analyzer) is a computer application used to track, intercept and log network traffic that passes over a digital network. A packet analyzer also may be used by malicious agents to intrude on networks and steal information from network transmissions [Tech017a].

A packet analyzer shows the complete status of all network activities by providing a complete picture of bandwidth and resource utilization. Every action of a packet analyzer is performed in real-time [Tech017a].

Signal conditioning of the sensed traffic is achieved by the data that a sniffer provides about individual packets intercepted. These data usually include the following: Packet number, arrival time, source IP, target IP, protocol, size, and additional information. Both the Internet/network traffic sensing and the signal conditioning mirror the light path that takes place in the human eye, which goes through the cornea, pupil, lens, and fovea.

3.4 Signal Analysis for Detection of Network Anomalies

Metrics design has the ultimate purpose of preserving the key features (*e.g.*, natural sounding speech, edges and textures in images, motion in video) of signals (*e.g.*, audio, images, video) [Kins002]. It is then of extreme importance developing robust metrics, which would be capable of identifying key features in the signals and data of interest pertaining to network security in this research.

It has been extensively discussed in the literature that energy itself carries no information

and that energy-based metrics are of no use to tackle highly complex research problems (*e.g.*, those posed by cognitive systems). However, for completeness, energy-based metrics commonly used in engineering problems, as is the case of *data compression*, are included here and not limited to *mean squared error* (MSE), *mean energy of the source* (MES), *signal to noise ratio* (SNR), and *peak SNR* obtained by taking the *peak source energy* (MESmax) [Kins002].

The increasing number of nodes on the networks, different protocols and port numbers, new applications (*e.g.*, multimedia delivery or cloud services) challenges network administrators and researchers when measuring and monitoring Internet traffic on high speed networks. Network monitoring branches into *active*, where network routers are queried directly and periodically to collect statistics, and *passive*, where the network is analysed only at the edge routers and the network measurement parameters are deduced by applying mathematical formulae [DDHT008].

Fractal signal processing seems suitable for both direct and indirect measurements of networks features and anomalies due to the self-similarity nature of network traffic [Kins020]. Network traffic flows are neither completely understood in their dynamics nor easily controllable. The fractal properties of time series are revealed by the presence of *self-similarity*, a rigorous statistical property. Self-similarity denotes fractal behaviour where similar patterns in the new time series are obtained regardless of the sampling time scale used for examining the data ([DDHT008] and [Kins020]).

Detection of anomalies is a major goal in network security monitoring. Anomalies represent deviations from normal network behaviour. The *network anomalies* sources are *network failures and performance problems* (*e.g.*, file server failures, broadcast storms, and transient congestion) or *network security* (*e.g.*, denial of service attacks variants and network intrusions). Network anomalies are characterized by correlated transient changes in measured network data that occur prior to or during an anomalous event [ThJi003]. The term transient fluctuations refers to the measured data abrupt fluctuations occurring in the same order of magnitude of the sampling interval [DDHT008]. The idea of events taking place on all scales on an object (self-similarity) is important because it indicates long-range relations (power-law distributions) [Kins020]. Similarly, the Devaney definition for fractal states that it is a self-similar object whose fractal dimension exceeds its topological dimension [Kins020].

It is not possible to identify network transients using current probing tools or dedicated monitoring software. Hence, the analysis of network transients requires the development of specialized and robust metrics, as presented in this research, which are rooted in fractal techniques and methodologies. Network anomaly detection methods are [DDHT008]: Rule-based approaches [ScKW996], finite state machine models [LaWD992], pattern matching [ClSc004] (implemented in field programmable gate arrays (FPGAs)), and statistical analysis [Duff004]. Statistical analysis based methods are capable of continuously tracking the behaviour of the network and require no significant recalibration or retraining [DDHT008]. Methods incorporating fractal analysis for network security are also present in the literature ([KhFK015a] and [KhFK015b]).

3.5 Monoscale Analysis

Monoscale analysis of a signal requires (in the context of cybersecurity): (i) An *epoch* of Internet traffic, the overall period of interest in the system from which such signal is acquired; (ii) a stationary *frame* that produces continuous segments, derived from the epoch, that may be either non-overlapping or partially overlapping; and (iii) either an *operator* or a *transform* affecting the samples contained in the operating frame. The epoch realization of a signal is sampled with a regular scale defined by $\delta t = 1/f_s$, where f_s represents the sampling frequency satisfying $f_s > f_N$, which is the requirement for the sampling frequency to exceed the Nyquist frequency ([Kins020], [PrMa996]).

The sampling period, δt, creates inherently a constant distance between samples, analogous to a ruler with a single scale unit. Monoscale analysis, in different schemes and applications, follows the conventional treatment of sequences connected with most of the signal processing being done in the traditional monoscale ecosystem. Hence, monoscale analysis utilizes all the information available within an epoch, which when acquired satisfies the Nyquist sampling frequency. Our scientific and technological society is familiar with the monoscale analysis approaches predominantly. Monoscale analysis is based on a single frame of an epoch at a time. The set of samples contained in a frame are operated on or transformed.

3.5.1 Monoscale Information-Theoretic Based Measures

The objective measures described previously based on energy (*e.g.*, MSE and peak SNR), carry no information and they do not agree with subjective quality measures. An instance of a subjective measure is the evaluation of the quality of an image according to the mean opinion score (MOS) protocol. The MOS protocol requires a set of human observers that are proficient for deeply appreciating fine, precise details, and specific features about an event or experience. Each expert provides a measure that describes the perceived experience. The sum of all the measures provided by each expert or judge is then averaged by the observers set size. The MOS protocol is regarded as one of the best methods for judging the subjective quality of images [WaBo006]. Hence, energy-based metrics are not suitable to look for features in either signals or data because of the ambiguity potential. Instead, information-theoretic-based metrics should be considered in feature extraction. Consequently, all of the studies in this thesis consider objective metrics based on information theory. Examples of the relevance of information-theoretic based metrics in engineering problems are: (i) An edge of an object in an image may not carry much energy, but it may be critical in its shape recognition, (ii) a stop consonant in speech may be insignificant energetically and broadband spectrally, but it may be critical in speech recognition, (iii) whispering a message requires negligible energy, but the message itself unquestionably carries information, (iv) formants of the utterance and their transformations in time carry much more information than their energy, (v) fricatives also convey more information that would be implied by their energy [Kins004]. Non-energy based metrics relate to the concepts of information, a measure of complexity, and entropy. For measuring sets related to processes governed by power-law relationships, the Lebesgue and Hausdorff measures can be used. Measure here means assigning a number $\mu(\bullet)$ to a set in an n-dimensional space with the purpose of characterizing such set. This projected number is enclosed in the interval $[0, \infty]$. The Lebesgue measure assigns a number $\mu(\bullet)$ to an n-dimensional set that exists in the Euclidean space. The Hausdorff measure is a generalization of Lebesgue that is operational in lower dimension subsets from an n-dimensional set (*e.g.*, a curve, a surface, or a fractal set, where each of them could be contained in a 3-dimensional set) [Edga008].

3.6 Multiscale Analysis

Multiscale analysis inspects an epoch with a varying scale, *frame*, denoted by Δt. The samples contained in a frame can be discontinuous when compared with an epoch in the monoscale sense and are further partitioned by *volume elements* or *vels*. A frame is usually bigger than the sampling period, $\Delta t > \delta t$. However, if the frame size either equals the vel size or matches the sampling period $(\Delta t = \delta t)$, an epoch is then digitally processed according to the monoscale analysis. A frame should include at least 30 vels to satisfy minimal statistical significance in order to have validity for multiscale analysis [Kins994].

It has been in the last decades that innovative signal processing approaches based on fractal measurements have been developed, as described above, for creating frames containing discontinuous samples. The discontinuity in the samples contained in a frame allows searching for information that may be scattered at different scales in an epoch. These unconventional approaches depart from monoscale analysis and allow signal processing to develop new techniques and methodologies applicable as potential solutions to real engineering problems. Cybersecurity signals particularly require the searching of information which may be dispersed in a dataset, hence, the need for multiscale analysis in cybersecurity. The information-theoretic based multiscale analysis methodology, VFDT, and the implementation are described next.

3.6.1 Variance Fractal Dimension Trajectory

This subsection describes a polyscale methodology that measures the complexity of a signal [Kins011]. The methodology of polyscale analysis requires simultaneously computing: (i) A partitioning process in which the signal scale between samples is multiplied or divided by a constant factor, which creates subsignals (scaled signals), (ii) an information-theoretic-based measure for each scaled signals, which describes them numerically, and (iii) a complexity-based measure for the overall number of subsignals created by the partitioning process. The partitioning process allows accessing properties of the subsignals, based on the scale size used at a given stage, for their analysis, while the information-theoretic-based measure compresses the subsignal to a single number or subsignal subdescriptor, and the complexity-based measure compresses the subdescriptors further to a complexity measure or signal descriptor.

If the subsignals, organized according to the partitioning process, present variation of their properties resembling a **power law** through the distinct involved scales, then it is concluded that the signal under analysis is a fractal, and subsequently a fractal dimension describing those properties can be computed. The calculation of a fractal dimension in terms of variance known as VFD ([Kins020], [KiGr008], [KiGr010], [Kins007], [KCCP003] and [KCCP006]) is used as a tool to determine the complexity of signals produced by natural phenomena or synthetized by computers. This subsection describes and verifies the implementation of the VFD algorithm by relying on a known process, as is the case of Gaussian white noise (GWN) and uniform distributed noise.

A time series can be analysed directly in time by computing the spread of the increments in the signal amplitude (*i.e.*, through its polyscale variance denoted as σ^2. The variance fractal dimension can be computed in real-time [Kins020]. An important characteristic of the VFD is that it does not require a window in the Fourier sense, and therefore it does not introduce corresponding artifacts [Kins020].

The variance fractal dimension, D_σ, is determined by the Hurst exponent (in honour of Harold Edwin Hurst) denoted by H and which measures long-term dependences in a time series. A Hurst exponent falling in the interval $H = [0,\ 0.5]$ denotes a time series having long-term alternations between high and low values (or uncorrelated) in contiguous samples (*e.g.*, a given sample with a low value would probably be followed by a sample with a high value. This alternating tendency would persist a long time into the future), whereas a Hurst exponent in the interval $H = [0.5,\ 1]$ (*e.g.*, $H \approx 0.91$ for the Nile river, which reflects its long droughts) is an indicator of a time series with a long-term positive autocorrelation (*e.g.*, a given sample in the series would be followed by a sample of similar value and further samples long time into the future would tend to fall into similar values). A special case is presented when $H = 0.5,$ which is an indicator of a time series that is completely uncorrelated [Kins020].

The variance, $\sigma^2,$ of the amplitude increments of a signal $B(t)$ (continuous or discrete in time) over a frame Δt is related to the time increment according to the following power law [Kins020]

$$\mathrm{var}\left[B(t_2)-B(t_1)\right] \sim \left|t_2-t_1\right|^{2H} \tag{3.1}$$

where var(•) denotes the variance operator, the symbol ~ reads "is proportional to", and $t_{(\cdot)}$ stands for a time instant.

For $\Delta t = \left|t_2 - t_1\right|$ and $(\Delta B)_{\Delta t} = B(t_2) - B(t_1)$ the exponent H can be calculated from a log-log plot by Shannon [Shan948]

$$H = \lim_{\Delta t \to 0} \frac{1}{2}\left(\frac{\log_b \left[\mathrm{var}\left(\Delta B\right)_{\Delta t} \right]}{\log_b \Delta t} \right) \tag{3.2}$$

in the analysis performed here, the base b is 2. The embedding Euclidean dimension E (*i.e.*, the number of independent variables in the signal under analysis), the VFD can be computed from

$$D_\sigma = E + 1 - H,\ 1 \le D_\sigma \le 2 \text{ and } 0 \le H \le 1 \tag{3.3}$$

3.6.2 Implementation of the Variance Fractal Dimension Trajectory

A signal sampled over a time interval T with a constant sampling rate given by

$$f_s = 1/\delta t \tag{3.4}$$

where δt is the interval between two consecutive samples, produces a sample space with N_T points. This sample space is defined by

$$N_T = \mathrm{int}\left(\frac{T}{\delta t}\right) \tag{3.5}$$

The implementation of the technique to calculate the VFD in a digital signal consists of the following stages [Kins020]: (i) The signal sample space of N_T points is identified and stored in an array for further manipulations. (ii) The number of sizes of Δt (for creating the distinct scales or subsignals) at which the spread of ΔB should be computed for the log-log plot is obtained by $\Delta t_{K_{\max}} = n_{K_{\max}}$, $\Delta t \le T$. The frame Δt should not exceed the total sampling time T of the sample space. The parameters for the loop computation of the VFD are prepared as follows:

$$K_{\max} = \mathrm{int}\left(\frac{\log_b N_T}{\log_b} \right) \tag{3.6}$$

where $b=2$, in this case, is the base to form a *b*-adic sequence for time intervals n_k; $K_{\text{buf}}=\left[\log_b(8{,}192)/\log b\right]$, where, as an example, $N_T=8{,}192$ (desirable to be greater than 30 for statistical significance) represents the number of samples contained in the frame for the first computation in the loop; $K_{\text{hi}}=K_{\max}-K_{\text{buf}}$; and $K_{\text{low}}\geq 1$. The main loop to obtain the VFD performs k cycles from $k=K_{\text{hi}}$ to $k=1$ in which the number of samples at each k is $n_k=b^k$. The number of windows in the signal is represented as $N_k=\text{int}\left(N_T/n_k\right)$, and the variance for each stage is then

$$\begin{aligned}\text{var}(\Delta B)_k &= \text{var}\left[(B(jn_k)-\mu)-(B((j-1)n_k)-\mu)\right] \\ &= \left[\frac{1}{N_k-m_k-1}\right]\left[\sum_{j=1}^{N_k}(\Delta B)_j^2-\frac{1}{N_k-m_k}\left[\sum_{j=1}^{N_k}(\Delta B)_j\right]^2\right] \\ &= \left[\frac{1}{N_k-m_k-1}\right]\left[\sum_{j=1}^{N_k}(\Delta B)_j^2\right]\end{aligned} \tag{3.7}$$

The amplitude increment is given by

$$(\Delta B)_j=B(jn_k)-B((j-1)n_k) \quad \text{for } j=1,\ \ldots,\ N_k \tag{3.8}$$

Figures 3.1 and 3.2 describe graphically the calculation of the VFD for developing intuition for its internal process.

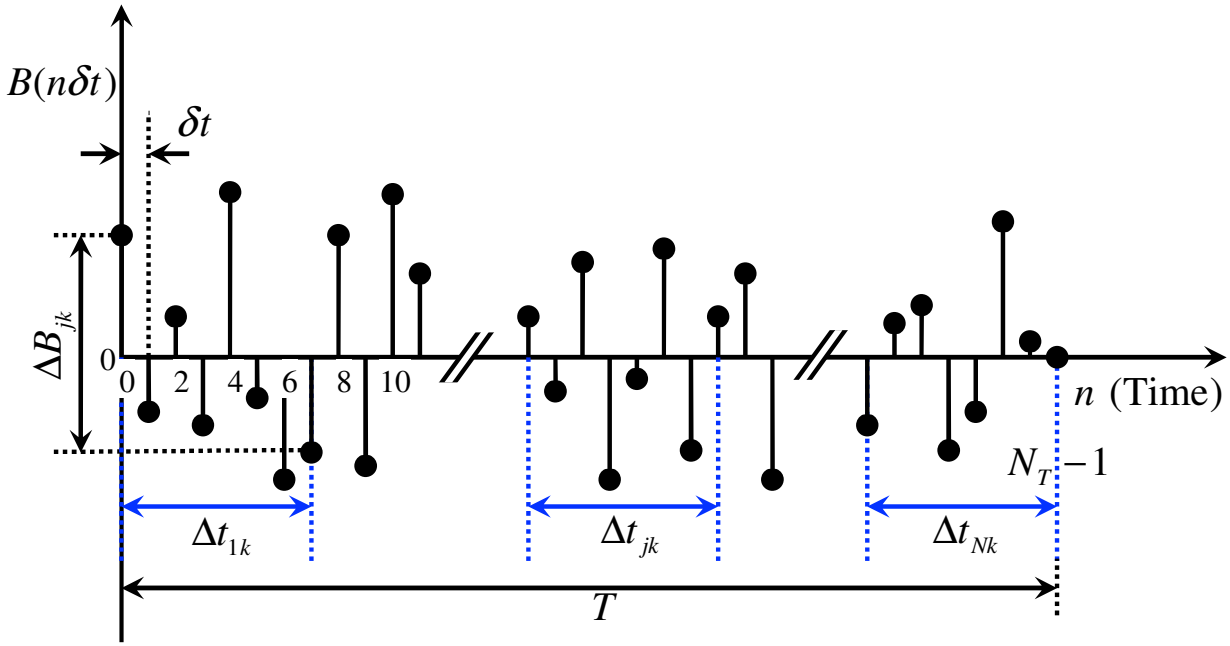


Fig. 3.1. Application of the variance fractal dimension (VFD) to an arbitrary signal. The segments in blue denote the time displacement applied at a given scale. After [Kins020].

The log values $X_k = \log[n_k]$ and $Y_k = \log[\text{var}(\Delta B)_k]$ are stored for the $\log - \log$ plot and the least-squares fit to obtain the slope s of the line is obtained using

$$s = \frac{K\sum_{i=1}^{K} X_i Y_i - \sum_{i=1}^{K} X_i \sum_{i=1}^{K} Y_i}{K\sum_{i=1}^{K} X_i^{\,2} - \left[\sum_{i=1}^{K} X_i^{\,2}\right]^2} \tag{3.9}$$

The Hurst exponent is computed by $H = (1/2)s$, and the VFD is obtained by applying (3.1).

For a non-stationary sequence, this process is repeated on successive windows (either non-overlapping or overlapping) to obtain a VFDT [Kins020]. If the VFDT is constant, then the sequence is a monofractal in time. Also, if the VFDT has segments with different slopes, the sequence is then a multifractal in time. Both VFD and VFDT, as multiscale analysis methodologies, have been extensively studied and used by the author ([TeKi012a], [TeKi012b], [TeKi012c], and [Terr012]). Similarly, monoscale statistical analysis has been performed [TeKi013a].

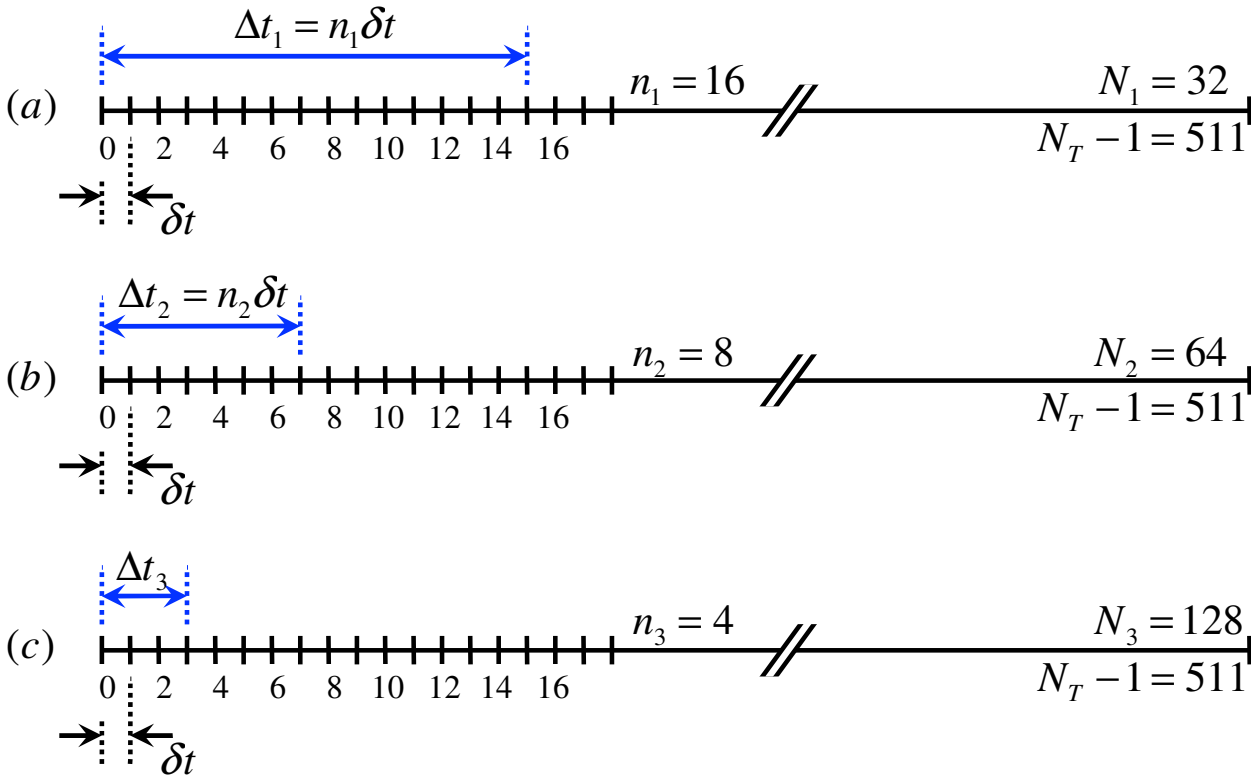


Fig. 3.2. Variance fractal dimension (VFD) calculation for a signal with 512 samples. The three segments in blue denote distinct time scales displacements. After [Kins020].

Considering the VFDT as a process for signal multiscale analysis of lower order moments, a generalized multiscale analysis methodology that utilizes arbitrary operators for searching properties can be derived. Such multiscale analysis methodology is introduced next.

3.7 Multiscalors: A Generalized Multiscale Analysis Methodology

This thesis introduces a novel generalized multiscale analysis methodology "*multiscalors*" capable of making arbitrary operators functional in multiscale analysis. Characterization of signals, in a given time frame, is provided by moments (*e.g.*, mean, variance, skewness, or kurtosis) classically operative in monoscale analysis only. *Multiscalors* is a methodology that has been devised for allowing a selected operator or a given signal analysis methodology of interest for being functional in the context of multiscale analysis.

The generalized multiscale analysis of a digital signal utilizing arbitrary operators requires deriving a sequence with multiscale nature from the signal $B(t)$, such a sequence is

provided by (3.8) and an arbitrary operator is utilized on it for each value of j. Hence, this multiscale analysis methodology is defined by the *multiscalor* operator

$$\Xi_{|||^n}[\bullet] \tag{3.10}$$

Where Ξ represents an arbitrary operator applied in a multiscale approach, $|||$ is a short hand notation indicating that such operator is applied in multiscale analysis, n refers to the multiscalor component (one of n data streams created as a result of applying a given operator), $\bullet$ in this case represents either a set of samples or their relationships (*e.g.*, ΔB that represents the amplitude differences of the signal $B(t)$ over the time increment Δt) from the operating frame. Hence, the term *multiscalor* is introduced in this research when referring to $\Xi_{|||^n}[\bullet]$.

Equation (3.10) stands for a generalization of (3.1), which allows for utilizing arbitrary operators in multiscale analysis. The mathematical operators that restrict (3.1) have been removed in order to create a sequence resembling the activity in the multiscale analysis domain for a given signal. The operators that have been removed are the variance and the Hurst exponent H. Hence, this generalization provides results based on raw data inherently extracted from multiscale analysis rather than providing explicit links to power laws and fractal dimensions. Nevertheless, the process for creating b-adic sequences, for the signal under analysis, is maintained intact. It is precisely this mechanism that allows access to the long-range dependencies that may be present in the signal under multiscale analysis. The availability of such a raw multiscale sequence allows the utilization of any arbitrary operator (conventionally applied in the monoscale analysis domain only) in multiscale analysis. Consequently, this thesis investigates processes, such as the Internet traffic, governed by power-law relationships.

The quantities that are crucial in the b-adic process for the creation of the multiscale signal are K_{hi}, K_{max}, K_{buf}, and K_{low}. The loop involved in the computation of these quantities has previously been used for the computation of the variance and the variance fractal dimension trajectory ([TeKi011], [TeKi013b]). Now, since the mathematical operations have been adapted, the variance can be replaced with any arbitrary and optimal operator, or even further a combination of them. Thus, multiscalors becomes the pivotal methodology empowering the search for long range dependencies in this research. It is important to note that computational

technologies supporting multiscalors methodologies introduced through this research and specific applications for DDoS detection are covered by intellectual property protection in the form of patents ([TeKi019a] and [TeKi019a]).

3.7.1 Variance

The variance reflects the dispersion degree of a probability mass function (pmf) of random variables around the mathematical expectation. It shall be noticed that the value of variance is always positive. Variance is used to describe important indicators of fluctuations in signals (*e.g.*, image denoising) [ZhWC012].

Variance is defined as:

$$\sigma^2 = \frac{1}{N_k}\sum_{i=1}^{N_k}(x_i - \mu)^2 \tag{3.11}$$

where x_i is the individual realization of the random variable X, N_k is the sample size of X, and μ is the expected value of X [ZhWC012].

In this thesis, variance is further defined as:

$$m_2 = \sigma^2 \tag{3.12}$$

3.7.2 Skewness

Skewness is the third statistical moment that characterizes a pmf by measuring its asymmetry. The values for the skewness provide important information: (i) It is zero for symmetric distributions, (ii) when it is positive, its main mode is positioned to the left and usually a long tail is positioned to the right, and (iii) when it is negative, its main mode is positioned to the right and usually a long tail is positioned to the left. For the last two cases, it should be noted that the more negative/positive the skewness is an indication that such pmf differs significantly from a Gaussian and its process has no resemblance with a symmetric population [DoSe011].

Skewness is defined as:

$$\gamma = E\left[\left(\frac{x-\mu}{\sigma}\right)^3\right] = \frac{E\left(x-\mu\right)^3}{\sigma^3} \tag{3.13}$$

where E is the expectation operator, x is the individual realization of the random variable X, μ is the mean, and σ is the standard deviation.

In this thesis, skewness is further defined as:

$$m_3 = \gamma \tag{3.14}$$

Research work related to the application of skewness in multiscalors has been published by the author of this thesis. This can be found in ([TeKi018] and [Terr020]), where the generalized multiscale analysis methodology has been introduced. This publication states that the results obtained through the skewness multiscalor applied to Gaussian random noise (GRN) are congruent in value with monoscale analysis. Nevertheless, multiscale analysis can have access to the information found in the long-range dependencies.

3.8 Selected Signal Analysis Methodologies Applied to Multiscalors Epiphenomena

When applying an arbitrary operator via multiscalors to a given signal, a form of *epiphenomena* occurs creating a number of multiscale sequences packaging the information content (potential long-range dependencies) of the original data. The number of multiscale sequences is dependent on the size of the processing frame.

It is also possible to examine further the by-products of the multiscalors epiphenomena with alternative monoscale and multiscale approaches to gain more insight concerning the information content in the original data. Nevertheless, for the purposes of this research, selected operators are applied through multiscalors in a single level only. The selected operators are described next.

3.8.1 Cumulative Sum

Consider $x[n]$, defined to be an ordered sequence of numbers, where index n is an integer in the range $-\infty$ to $+\infty$. The purpose of n is to keep track of the relative ordering of values in sequence x. When a specific time value is associated with n, such as nT seconds, then the sequence $x[n]$ becomes a discrete-time signal [LaGo018].

There are plenty of known discrete-time *signals*, signal *properties* (*e.g.*, energy and power, summable sequences, periodic sequences, or sum of periodic sequences), and signal

operations (*e.g.*, time shift, time reversal, time scaling, or cumulative sum, or backward difference). This thesis considers the cumulative sum operator of a discrete-time signal $x[n]$, which is itself a function of the independent time index n. The cumulative sum is defined as ([CPFS014], [LaGo018], and [Page954]):

$$S = \sum_{n=-\infty}^{k} x[n] \tag{3.15}$$

In this research, only the pure cumulative sum is considered and alternative complex measures (*e.g.*, scoring systems, control charts, change detection monitoring) are kept aside as the discrete sum S reflects the digital content accumulated over time.

3.8.2 Zero-Crossing Rate

Many signals are quasistationary, as is the case of speech, and their properties (*e.g.*, level-crossings, zero-crossings, energy, and information theoretic related features) are often studied by segmenting them in windows that are stationary within that specific window [ShSe012]. Even though speech is a non-stationary signal, it remains nearly unvaried for small segments (*i.e.*, for 10 to 50 ms) [JaBM013].

Stationarity ranges from *wide sense stationarity* (WSS) to *strong sense stationarity* (SSS). New and versatile approaches that do not fall in extremes have been proposed in the literature as is the case of *finite sense stationarity* (FSS) [TeKi013b].

An intuitive indication of how "busy" a signal becomes can be estimated by the number of times it crosses either the zero-activity line for alternating signals, or some other reference level for oscillating signals. The zero-crossing rate (ZCR) is defined as the number of times the signal crosses the reference within a specified interval.

In its simplest form, the frequency of a sinusoid is estimated as half the number of zero-crossing counts per second [ShSe015]. More formally, ZCR is a measure of "frequency composition" of a signal, which is more valid for narrowband signals such as sinusoids [JaBM013]. A sinusoid of frequency f_0 sampled with a frequency f_s produces f_s / f_0 samples per cycle, which possess two zero crossings per cycle. This results in the ZCR defined as [JaBM013]

$$Z = \frac{2f_0}{f_s} \tag{3.16}$$

The interpretation of the average ZCR for broadband signals is less precise. However, the use of short-time average ZCR could provide good estimates of the signal properties [JaBM013].

The definitions of ZCR for discrete computation is defined as [JaBM013]

$$Z_n = \sum_{m=-\infty}^{\infty} \left[\left| \text{sgn}(x[n]) - \text{sgn}(x[m-1]) \right| \right] w[n-m] \tag{3.17}$$

where the sgn(•) is defined as the sign function and w[•] represents the window containing a stationary segment of the signal under analysis.

The sign function is represented as [JaBM013]

$$\text{sgn}(x[n]) = \begin{cases} 1, & x[n] \geq 0 \\ -1, & x[n] < 0 \end{cases} \tag{3.18}$$

and the stationary window is [JaBM013]

$$w[n] = \begin{cases} \dfrac{1}{2N}, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise.} \end{cases} \tag{3.19}$$

where N represents the total number of samples contained in the window. An estimation of the frequency content of a signal is provided by the ZCR by the occurrences, in a given time interval/frame, of a sign change in a given signal. The rate at which zero crossings occur is a simple measure of the frequency content of a signal.

Zero crossing rate is very useful for discriminating a broadband signal from noise. Furthermore, ZCR helps in determining the beginning and the end of segments of interest in a signal [JaBM013].

Research work related to zero-crossing rate has been published at two international conferences ([TeKi016a] and [TeKi016b]) and in two journals ([TeKi016c] and [TeKi016d]). This research work relates to: (i) The generation of processes with the characteristics of Lévy walks, which reflect the dynamics of Internet traffic and its relevance in cybersecurity is portrayed, (ii) the direct computation in the time-domain for obtaining the ZCR and its advantages for real-time implementations of feature extraction of signals, (iii) the insight

provided by the ability of ZCR to identify sections in a composite signal, and (iv) the practical application of ZCR in cybersecurity by inspecting a dataset containing a documented DDoS attack where the beginning and the end of the cyberattack were identified clearly.

3.8.3 Entropy

Information theory addresses two fundamental concerns in communication theory: (i) The ultimate data compression through *entropy* measures, and (ii) the ultimate transmission rate of communication through the *channel capacity* (computed from the noise characteristics of the channel). Reliable communications are bounded between the compression limit, entropy, and the data transmission limit, channel capacity. All data compression schemes, and modulation schemes exist within these limits.

Entropy is a probabilistic measure, oriented to determine *redundancy*, of the spread of probabilities of individual symbols in the source with respect to the equal (uniform) symbol probabilities. When the source entropy is maximum due to equal probabilities of the source symbols there is no redundancy in the source alphabet. This random, patternless like, source cannot be compressed without a loss of information. The difference between entropies from the source and the code determines the quality of the code.

3.8.3.1 Self-Information

Information is interpreted as the reduction in uncertainty of the *frequency of occurrence* for a symbol representing an event. Stated in another form, uncertainty reduction causes information gain.

The Shannon's self-information I_j of the n_j event is defined as

$$I(\sigma_j) \equiv I_j \triangleq \log_b \frac{1}{p_j} = -\log_b p_j, \text{ [information unit, or u]} \tag{3.20}$$

where $p_j = p(\sigma_j)$ for brevity and denotes probability, and b in the Shannon's entropy sense is the size of the coding alphabet Γ_C required to code each symbol. Since each symbol probability is confined to the unit interval $p_j = [0,\ 1]$, the self-information is non-negative $I_j = [0,\ \infty]$. For a binary coding alphabet $\Gamma_C = \{0,\ 1\}$, $b = 2$, and $u \equiv \text{bit}$ (binary digit), while for natural base

$b = e,\ u \equiv nat$ (natural digit), and for $b = 10,$ and $u \equiv$ Hartley [Kins004]. Self-information measures the information content carried by a specific symbol provided by an information source. It is observed that the symbols with smaller probabilities, less common, carry more self-information, while the symbols with higher probabilities, more common, carry smaller self-information. Self-information is also known as *surprisal* due to its relation to surprise. The higher the self-information is, the higher is the potential for surprise. Compactly, self-information measures the importance, inversely proportional to probability, of a specific symbol provided by a source.

3.8.3.2 Shannon's Entropy

Shannon's entropy concept, now a fundamental notion through the sciences, is of particular importance for communications and cryptography [Stin006]. Shannon's theorems state how reliable communications are and how much meaningful information is conveyed over a given channel ([BBMW014], [Shan948], and [Shan949]).

Shannon's entropy also provides a measure for the average self-information, regardless of the message size. Entropy is then *a weighted average of probabilities* [Kins004]. It is then defined as the average (expected) value of self-information

$$H \triangleq \sum_{j=1}^{N_S} p(\sigma_j) I(\sigma_j) \tag{3.21}$$

$$H = -\sum_{j=1}^{N_S} p(\sigma_j) \log_b p(\sigma_j) \tag{3.22}$$

$$H = -\sum_{j=1}^{N_S} p(j) \log_b p(j) \tag{3.23}$$

$$H = -\sum_{j=1}^{N_S} p_j \log_b p_j \text{ [u/symbol]} \tag{3.24}$$

where N_s is the size of the source alphabet $\Sigma = \{\sigma_1,\ \sigma_2,\ ...,\ \sigma_{N_S}\}$ and $p(\sigma_j) \equiv p(j) \equiv p_j$ is the probability of the j^{th} symbol taken from the corresponding pmf $f_p = [p_1,\ p_2,\ ...,\ p_{N_S}]$. When all the probabilities are equal, the weighted average turns into a simple average. This entropy function is non-negative and concave in f_p ([CoTh005] and [Kins004]). It is important to

highlight that plenty of other entropic measurements exist. The classical Shannon's entropy has been considered in this research as a base case.

3.9 Features Availability

Kulikowski and Weiss [WeKu991] discussed that any learning computer system is at the mercy of the sample data and the *features quality*. A given feature ideally should be highly representative of the raw data phenomena under analysis. However in practice, this varies depending on the analysis methodologies or techniques used for such analysis. It is assumed that Kulikowski and Weiss refer to data and features processed in a *monoscale* setting. The research work presented in this thesis is *multiscale* in nature and reflects the Internet/traffic (signals with a high spiky behaviour) dynamics that permeate into all the scales due to the fractal nature of this type of traffic. Kulikowski and Weiss consider that features fall within three categories, completely noisy, somewhat noisy, and completely predictive, but no criteria for determining clearly how a feature can be classified in any of the three is provided. Nevertheless, one would expect that applying denoising techniques to the outcomes of a feature, this would acquire a more predictive nature in the sense that it would become less spiky. The specific research work introduced in this thesis for feature extraction considers an initial stage of multiscalor operators (variance or skewness) followed by a secondary operator (cumulative sum, ZCR, or Shannon's entropy). After this, denoising would help for increasing the predictive power of the features utilized, according to the point of view of Kulikowski and Weiss. Imprinting this predictive nature into the extracted features reduces the number of classes created by ART and provides a more compact output for interpreting the DDoS attacks occurrences. To the best of the author's knowledge, increasing the predictive power of features through multiscalors components would become the first reported case of applying denoising methodologies on processed data through multi- and polyscale analysis.

In order to increase the predictive power in the results yielded by multiscalor operators in the multiscalor components, two denoising techniques are utilized Donoho's denoising (multiscale based) and nonlinear filtering (median filter with monoscale nature). These denoising techniques increase the predictive capacity of multiscalor components by reducing the spiky behaviour of the extracted features.

3.9.1 Denoising

There are various wavelet based *denoising* schemes attempting to reject noise by damping or thresholding in the wavelet domain [Dono995]. A formal approach of the term denoising has been proposed by Donoho [Dono995] and has shown how wavelet transforms may be used to optimally "de-noise" in such interpretation.

An unknown function f on [0, 1] can be recovered from noisy data

$$d_i = f(t_i) + \sigma_D z_i,\ i = 0,\ ...,\ n-1 \tag{3.25}$$

where $t_i = i/n$, z_i, is a standard Gaussian white noise (independent and identically distributed (iid); denoted by $z_i \overset{idd}{\sim} N(0,\ 1)$, and σ_D is a noise level. Donoho's interpretation of "denoising" is setting as a goal the optimization of the mean-squared error

$$n^{-1} E \left\| \hat{f} - f \right\|_{\ell_n^2}^2 = n^{-1} \sum_{i=0}^{n-1} E(\hat{f}(i/n) - f(i/n))^2 \tag{3.26}$$

where ℓ^2 represents the L^2 norm (Euclidean distance), $E[\circ]$ represents expectation, and there is the condition that with high probability $\hat{f}$ is at least as smooth as f [Dono995]. Many statistical techniques optimize the mean-squared error causing a trade-off between bias and variance keeping the two terms about the same order of magnitude [Dono995].

3.9.1.1 Thresholding denoising procedure

i. Apply the interval-adapted pyramidal filtering algorithm of Cohen, Daubechies, Jawerth, and Vial (*aka* CDJV) to the measured data $\left(d_i / \sqrt{n}\right)$, obtaining empirical wavelet coefficients (e_I).

ii. Apply the soft thresholding nonlinearity

$$\eta_t(y) = \text{sgn}(y)(|y| - t)_+ \tag{3.27}$$

coordinatewise to the empirical wavelet coefficients with specially chosen threshold $t_n = \gamma_1 \cdot \sigma \cdot \sqrt{2\log(n)/n}$, and γ_1 is a constant.

iii. Invert the pyramid filtering, recovering

$$(\hat{f}_n^*)(t_i),\ i = 0,\ ...,\ n-1 \tag{3.28}$$

It has been proved that in addition to the good visual quality, the estimator has an optimality property with respect to mean-squared error for estimating functions of *unknown* smoothness at a point [Dono995].

It has been proven that in Donoho's denoising, two phenomena, smoothing and adapting, are held with considerably generality. Smoothing holds because with high probability $\hat{f}_n^*$ is at least as smooth as *f*, with smoothness measured by any of a wide range of smoothness measures. Adapting holds because $\hat{f}_n^*$ achieves almost the minimax mean-square error over every one of a wide range of smoothness classes, including many classes where traditional linear estimators do not achieve the minimax rate. Some additional properties of Donoho's denoising are: (i) The coefficient reconstruction is noise free, (ii) using thresholding or other nonlinearities in the Fourier domain cannot match its broad adaptive thresholding in the wavelet domain, (iii) it has a special optimality enjoyed by no other nonlinearity, (iv) it adapts easily to higher dimensions and to sampling operators which compute area averages rather than point samples, and (v) the noise level f does not have to be known and it suffices to apply the threshold $\hat{t}_n = \gamma_1 \hat{\sigma}\sqrt{2\log(n)/n}$, where the scale estimate $\hat{\sigma} = MAD/0.6745,$ with the $MAD,$ median absolute deviation, value of the appropriate normalized fine-scale wavelet coefficients $\left(\sqrt{n}\cdot w_{j_1-1,k}\right)_k$ [Dono995].

3.9.2 Non-Linear Filtering

Nonlinear filtering techniques provide a better trade-off between noise smoothing and the retention of fine details. The non-linear filtering technique used here is median filtering, which has proven to be useful for the suppression of impulse like or shot disturbances in images and it falls into classical digital signal processing techniques. Median filtering has interesting advantages like: (i) Avoiding blurring of edge features, (ii) causing no reduction in contrast since the output values come from the values present in the neighbourhood, (iii) non-shifting boundaries, and (iv) having less sensitivity to extreme values (outliers) causing a more efficient removal [Prat001].

In the one-dimensional form, the median filter consists of a sliding window encompassing an odd number of pixels. The center pixel in the window is replaced by the

median of the pixels in the window. The median of a discrete sequence $a_1, a_2, ..., a_N$ for N odd is that member of the sequence for which $(N-1)/2$ elements are smaller or equal in value and $(N-1)/2$ elements are larger or equal in value. Hence, the neighbouring pixels are ranked causing that the median of this sorting would become the new value for the central pixel [Prat001].

3.9.3 Quantization

Data provided to an ART neural network are required to be presented in the form of a binary array. Hence, after non-linear filtering one needs to apply a method that would create a binary array. The method that is followed here is *least squares quantization* (LSQ), which utilizes the idea of spacing quantum values closely in the voltage regions where the amplitude of a digital signal produced by a *pulse-code modulation* (PCM) system is expected [Lloy982].

The Shannon-Nyquist sampling theorem is at the core of PCM systems. The sampling theorem asserts that a voltage signal $s(t), \ -\infty < t < \infty,$ containing only frequencies less than W cycles/s or Hz can be recovered from a sequence of its sample values according to

$$s(t) = \sum_{j=-\infty}^{\infty} s(t_j) K(t - t_j), \ -\infty < t < \infty \tag{3.29}$$

where $s(t_j)$ is the value of s at the j^{th} sampling instant

$$t_j = \frac{j}{2W}, \ -\infty, j, \infty \tag{3.30}$$

and where

$$K(t) = \frac{\sin 2\pi W t}{2\pi W t}, \ -\infty, t, \infty \tag{3.31}$$

is a sinc $\sin t / t$ pulse of the appropriate width [Lloy982].

A *pulse-amplitude modulation* (PAM) system is based on the sampling theorem alone. A sequence

$$..., \ s(t_{-1}), \ s(t_0), \ s(t_1), \ ... \tag{3.32}$$

of samples of the signal $s(t)$ is sent over a channel. The receiver constructs the pulses $K(t-t_j)$ and adds them together, with the received amplitudes $s(t_j)$, as in (3.29), producing an exact representation of the original band-limited signal s [Lloy982].

In PCM, instead of sending the exact sample values (3.32), the signal voltage range is partitioned into a finite number of subsets. The information to which subset a sample happens to fall in is then transmitted to the receiver. The receiver has a source of fixed representative voltages, *quanta*, one for each of the subsets. When the receiver is informed about a sample falling into a subset, it uses its *quantum* for that subset as an approximation to the true sample value and constructs an approximated band-limited signal [Lloy982].

The *noise signal* is the difference between the receiver-output signal and the original signal. The *noise power* is the average square of the noise signal. PCM considers the given number of quanta and certain statistical properties of the signal for determining the subsets and quanta that are best in minimizing the noise power [Lloy982].

Formally, a **quantization** scheme consists of a class of sets $\{Q_1, Q_2, ..., Q_v\}$ and a set of quanta $\{q_1, q_2, ..., q_v\}$. The $\{Q_\alpha\}$ are any v disjoint subsets of the voltage axis which, taken together, cover the entire voltage range. The $\{q_\alpha\}$ are any v finite voltage values. The number v of quanta is to be regarded throughout as a fixed preassigned number [Lloy982].

A partition $\{Q_\alpha\}$ is associated with a label function $\gamma(x)$, $-\infty < x < \infty$, defined for all (real) voltages x by

$$\begin{aligned} &\gamma(x) = 1 \quad \text{if} \quad x \quad \text{lies in} \quad Q_1, \\ &\gamma(x) = 2 \quad \text{if} \quad x \quad \text{lies in} \quad Q_2, \\ &\qquad\qquad \vdots \\ &\gamma(x) = v \quad \text{if} \quad x \quad \text{lies in} \quad Q_v, \end{aligned} \tag{3.33}$$

Therefore, the label a_j of the set that a sample $s(t_j)$ falls in is defined as

$$a_j = \gamma(s(t_j)), \quad -\infty < j < \infty \tag{3.34}$$

Consequently, the PCM based signal when sent over is then a sequence of labels

$$..., a_{-1}, a_0, a_1, ... \tag{3.35}$$

where each a_j is one of the integers {1, 2, ..., v} [Lloy982].

These labels contain information of the multiscalors based features in a compressed form and then become the binary code that is utilized as input to the ART neural network. Only the description of the quantization conceptual framework relevant to this research is provided.

3.10 Summary

The need for multiscale signal analysis is extensively discussed during this chapter. The discussion is built up around the monoscale analysis limitation for accessing information present in the long-range dependencies of processes, which is something that multiscale analysis, by its fractal nature, can surpass. The discussion for signal analysis based on multiscalors focuses in network security by aiming to detect the presence of DDoS attacks.

This chapter provided an in depth description of the distinct required subsystems for feature extraction. The sensing of the Internet/network traffic by sniffing packets is the means of signal acquisition. Context about monoscale analysis and its limitations is provided by highlighting that energy based metrics are inadequate for resolving information present in long-range dependencies. The critical importance of multiscale analysis for sifting information available in long range dependencies then becomes apparent. The novel methodology "multiscalors", allowing arbitrary operators and signal analysis methodologies to be functional in the multiscale analysis context, is introduced as the main contribution of this research work. The statistical moments (variance and skewness) used as multiscalors operators and the signal analysis methodologies (cumulative sum, zero-crossing rate, and Shannon entropy) applied to the multiscalors components are presented in detail. Lastly, the methodologies applied to the extracted features (Donoho's denoising, non-linear filtering, and quantization) in order to make them readily available for to further by machine learning stages (*i.e.*, ART). The next chapter elaborates deeply on computational intelligence methodologies considered in this research.

CHAPTER IV

COMPUTATIONAL INTELLIGENCE APPROACHES UTILIZED FOR DISTRIBUTED DENIAL OF SERVICE DETECTION

In machine learning, *supervised*, *semi-supervised* or *hybrid*, and *unsupervised* are three ways to classify anomalous packets from normal packets. Supervised methods have the privilege of differentiating anomalous and normal data from a labelled dataset. Unsupervised methods, on the other hand, segment a dataset into different clusters where the strength of the clustering usually lies within the algorithm itself [DaVS020]. Appendix C covers computational intelligence approaches in depth.

Ensemble learning, *i.e.*, combining multiple classifiers to form a more powerful classifier, has been well-studied in the machine learning community and it has been proposed for some cybersecurity applications in DDoS detection. Ensembles are selected in some cases because they usually provide better results than a single classifier and many classification problems have benefited from the idea of combining multiple classifiers. Appendix D provides more information about ensemble classifiers [DaVS020].

Both surveys of machine learning based intrusion detection approaches (*e.g.*, [HOHR015] and [AZZS019]), and a systematic literature review and taxonomy of DDoS attacks [YuUS019] are necessary to know the machine learning landscape for both IDS and DDoS. An interesting survey about DDoS mechanisms against DDoS is found in [ZaJT013].

Yusof and Selamat [YuUS019] perform an in-depth analysis on DDoS attack types as well as on existing DDoS detection and attack prediction techniques. Also, factors behind the DDoS attacks are identified. Moreover, they have classified and ranked 53 articles from different

digital libraries (*e.g.*, Science Direct, ACM Digital Library, IEEE Xplore, Springer, and Web of Science) related to DDoS detection and prevention. It was found that 30% of these articles use machine learning techniques as their detection or prevention strategy [DaVS020].

Alessa et al. [AZZS019] reviews and analyses the research landscape for IDSs, considering DDoS and other cyberattacks, based on DL techniques. Alessa et al. [AZZS019] focus on 68 articles with the keywords 'deep learning', 'intrusion' and 'attack' and their variations in four major databases, namely Web of Science, Science Direct, Scopus, and IEEE Xplore. Three proportions are found: Developing an approach for evaluating or identifying intrusion detection techniques using the DL approach (72.06%), studying/applying articles to the DL area (22.06%), and discussing frameworks/models for running or adopting IDSs (5.88%) [AZZS019]. Three phases are proposed for detecting DDoS: Data collection and training, feature extraction and selection, and DL detection [AZZS019]. This phases, even though not available at the beginning of the research proposed here and being recently proposed, fit into research work carried in this thesis.

Research work that considers plenty of machine learning approaches are available and is well documented in the literature. Detection performance on some of the approaches already surpass 95% of precision [HOHR015]. However, some of the training datasets used in the literature contain 99% of normal data and 1% of anomalous data to make their model run efficiently and accurately in detecting anomaly by learning from normal behaviour [HOHR015]. In perspective, the dataset studied here contains 46% of anomalous data as a DDoS attack.

It is important to reiterate that the research herein developed and presented focuses on the development of deep robust features and on the usage of ART, as core machine learning approach. This premise aids mainly for understanding the mechanisms behind ART in-depth, which could allow envisioning methods for tuning the vigilance parameter. This knowledge gain creates the possibility of extrapolating the lessons learned regarding alternate machine learning approaches and also further improves the precision and recall results that would be obtained throughout this research.

The research presented in this thesis explores DL from the following views: (i) The *depth of the computing stages*, a novel viewpoint in DL investigated here, (ii) the extraction of *robust features* via *multiscalors*, a new methodology, through applying multi- and polyscale analysis,

and (iii) performing classification of the *abstract features* obtained via *adaptive resonance theory*, an unsupervised machine learning approach used in the context of multi- and polyscale analysis based DL for the first time.

The features obtained from the deep architecture by multiscalors (capable of extracting robust and refined abstract features) are used in the detection of classes of interest through ART as the machine learning approach.

This could be seen as a novel approach in DL, and might be even considered as an element of a new cognitive approach to DL. This argument appears to be justified because DL could be safely regarded as the study of models that involve a greater amount of ***composition*** of either learned ***functions*** or learned ***concepts*** than traditional machine learning [GoBC016]. The novel composition of "features of features" (FOF), as a cognitive element of DL, carried throughout this research captures a clearer and more refined view on the behaviour of the Internet traffic under study. Two machine learning models based on ART, ART1 and FuzzyART, are implemented throughout the thesis and a comparison of performance between them is established.

4.1 Adaptive Resonance Theory

Human memory has the ability to learn new things without forgetting things learned in the past (*e.g.*, recognizing parents after not seeing them for some time while learning new faces in the interim). This capability is highly desirable in ANNs as many of them tend to forget old information when incrementally adding new information [FrSk991].

An ANN usually performs pattern-classification operations by being trained with a set of exemplars or patterns. Training allows the encoding of information in the ANN by adjustment of the weight values. Once the training is deemed adequate, the ANN is put into production and no additional weight modification is permitted. This operational scenario is acceptable provided the problem domain has well-defined boundaries and is stable. Unfortunately, in many realistic situations, the environment is neither bounded nor stable [FrSk991].

If an ANN is presented with a previous unseen input pattern, there is generally no built-in mechanism for the network to be able to recognize the novelty of the input. The ANN does not know that it does not know the input pattern [FrSk991].

The previous discussion describes practically the **stability-plasticity dilemma (SPD)** coined by Stephen Grossberg. The SPD can be stated as a series of questions: How can a learning system remain adaptive (plastic) in response to significant input, yet remain stable in response to irrelevant input? How does the system know to switch between its plastic and its stable modes? How can the system retain previously learned information while continuing to learn new things? In response to such questions, Grossberg, Carpenter, and other colleagues developed **adaptive resonance theory**, which seeks to provide answers. In fact, an approach to solve the SPD is to add a feedback mechanism between a competitive layer and the input layer of a network. This feedback mechanism facilitates: (i) *Learning* of new information without destroying old information, (ii) *automatic switching* between stable and plastic modes, and (iii) *stabilization of encoding* of the classes done by the nodes. This feedback mechanism is exploited by ART and variants, (input vectors in a binary, analog or grayscale, or fuzzy form), that are suitable for pattern-classification problems in realistic environments [FrSk991].

ART gets its name from the way in which *learning* and *recall* interplay in the network. In physics, *resonance* occurs when a small-amplitude vibration of the proper frequency causes a large-amplitude vibration in an electrical or mechanical system. In an ART network, information in the form of processing-element outputs reverberates back and forth between layers. If the proper patterns develop, a stable oscillation ensues, which is the neural-network equivalent of resonance. During this *resonant period*, learning, or adaptation, can occur. Before the network has achieved a resonant state, no learning takes place, because the time required for changes in the processing element weights is much longer than the time that it takes the network to achieve resonance [FrSk991].

A resonant state is attained if: (i) The network had learned to recognize a previous input vector, then a resonant state would be achieved quickly when that input vector is presented. During resonance, the adaptation process reinforces the memory of the stored pattern; (ii) the input vector is not immediately recognized, and then the network searches through its stored patterns looking for a match. If no match is found, the network enters into resonant state whereupon the new pattern is stored for the first time [FrSk991].

Thus, the ART neural networks respond quickly to previously learned data, yet remains able to learn when novel data are presented. Grossberg has focused mainly on modelling actual

macroscopic processes that occur within the brain in terms of the average properties of collections of the microscopic components of the brain (neurons). Thus, a Grossberg processing element may represent one or more actual neurons [FrSk991].

ART was developed from the observation of biological phenomena, regarding vision, speech, cortical development, and cognitive-emotional interactions. This theory is based on three biological principles highlighted by the following characteristics: (i) *Signal normalization*. Ability of biological systems to adapt themselves to environments that change all the time (*e.g.*, the human vision system can adapt to different amounts of light; (ii) *Contrast intensification*. Capability of identifying subtle details in the environment through successive observations (*e.g.*, the respiratory system can perceive, almost instantly, a clean environment that began being polluted with carbon monoxide); and (iii) *Short-term memory*. Capacity to temporarily store sensorial information from the contrast intensification mechanism, before it can be decoded for decision-making [NHAB017].

One of the main features of ART networks is the ability to learn new patterns, when new samples are presented, without destructing previously extracted knowledge. This characteristic is associated with the plasticity/stability dilemma, where the system is flexible, *adaptive*, enough to incorporate environmental changes, whereas it must also be *stable* to preserve the knowledge already gained. This distinctive quality makes it one of the best ANNs architectures, which can deal with the stability/plasticity dilemma in a coherent and systematic way. Other distinctive characteristics are the following: (i) ART architecture has biological plausibility, which is principled by the signal normalization, contrast intensification, and short-term memory principles; (ii) the network training is always stable and after this stabilization (convergence), the presentation of a pattern that fits one category already created, directly activates the neuron corresponding to that group, with no need to initiate the search phase. In this case, the network works as an autonomous associative memory; (iii) the selection of the winner neuron in the recognition phase is also always stable. Once an input vector is associated with a group represented by a neuron in the *recognition layer*, this same unit always wins the competition, regardless of some eventual posterior adjustment in the forward or backward weights when a new training sample is presented; (iv) the occurrence of adaptive resonance depends, mainly, on how close the input sample is to the vector that represents the cluster, indicated by the winner

neuron. If the distance, weighted by the *vigilance parameter*, is acceptable, then, an adaptive resonance state is achieved, what in biological terms, corresponds to the gain and extension of neural activity; and (v) the level of details in each new class included in the network structure is based on the vigilance parameter value. The larger the value, the finer details and distinctive characteristic are considered from the patterns to be inserted [NHAB017].

Unlike some other neural architectures, it is verified that both the training phase and the operating phase of an ART network are included in the same algorithm, since its topology always needs to perform a similarity test to categorize the input sample. Furthermore, this learning is processed in an unsupervised manner, allowing the inclusion of knowledge inside classes that are already represented by existing neurons, or, evaluating if there is a necessity for the inclusion or enablement of other neurons as the samples bring new relevant knowledge [NHAB017].

4.1.1 ART: Equations Descriptions

4.1.1.1 Type-1: STM and LTM States Solved with Differential Equations

When the ART1 architecture was first reported, it was presented as a **biologically inspired** neural system described by a set of *short-term memor*y (STM) and a set of *long-term memory* (LTM) nonlinear and coupled time domain *differential equations*. STM equations described the instantaneous activation evolution for the neurons as a function of the externally applied inputs and the present set of interconnection weights, while LTM equations described the time evolution of the adaptive interconnection weights, which store the knowledge and experience of the complete system. In ART, STM equations settle much faster than LTM equations [SeLA012].

4.1.1.2 Type-2: STM States Solved with Algebraic Equations and LTM States Solved with Differential Equations

In an ART system, if the input patterns are held stable long enough so that STM equations reach their steady state, this steady state can be computed directly without solving the STM differential equations. The STM steady state can be obtained by solving a set of algebraic equations, properly sequenced. Hence, in this description of an ART system, the *STM state is*

computed by solving algebraic equations and the *LTM evolution is computed by solving the corresponding differential equations* [SeLA012].

4.1.1.3 Type-3: STM and LTM States Solved with Algebraic Equations

If input patterns are held constant long enough so that both STM and LTM equations settle to their respective states, then the *ART system operation can be described by solving properly sequenced algebraic equations only*. This description corresponds to the particular case called *Fast Learning* in the original ART1 paper. When the FuzzyART, ARTMAP, and FuzzyARTMAP algorithms were reported, they were described in their Type-3 or *Fast Learning* version, or, at the most, a *Slow Learning* LTM update was considered in which finite difference equations instead of differential equations are used [SeLA012].

Most of the reported work on ART architectures and their applications is developed as software algorithms, running on conventional sequential computers. However, the parallel nature of these architectures and the simplicity of its components calls in a natural way for hardware implementations, similar to what nature has done with brains in living beings. Also, the fact that these ART, its variants, and other architectures can be combined hierarchically to build higher level **cognitive systems** that solve complicated engineering problems (*e.g.*, robotics, vision systems and speech recognition), makes it even more attractive to develop a set of hardware components to be used in more complicated and hierarchically structured systems. There have been some attempts in the past to implement in hardware some of the aspects of ART architectures. However, they were intended to emulate Type-l or Type-2 descriptions of ART, and the results were bulky and inefficient pieces of hardware that could only realize part of the functionality of the powerful ART algorithms [SeLA012].

The author of this thesis gathers that the Type-3 description of ART, as it is behavioural, is preferred in most implementations targeting conventional sequential hardware, and specialized applications in very large integrated circuits (VLSI) or application-specific integrated circuits (ASIC).

4.1.2 ART: Topological Distinctions

4.1.2.1 ART1

The first ART neural network model appeared in the open literature in 1987 and is known as ART1. It is an unsupervised learning neural clustering architecture whose inputs are patterns composed of binary values and it groups them into categories according to a similarity criterion based on Hamming distances, modulated by a variable coarseness vigilance criterion. As a result, a set of extraordinary mathematical properties arises, rarely present in other algorithms of similar functionality [SeLA012].

4.1.2.2 ART2

Additionally in 1987, almost simultaneously to the ART1 publication, a similar algorithm named ART2 was published intended to cluster input patterns composed of analog valued (continuous) values. It is relatively more complicated than ART1 [SeLA012].

4.1.2.3 ART3

Presented in 1990, this ART topology is configured with binary inputs or analog inputs (continuous) and unsupervised training that uses multilevel topology and the chemical properties of neurotransmitters for the searching process of a better solution [NHAB017].

4.1.2.4 ARTMAP

Introduced in 1991, ARTMAP are supervised learning architectures that can be trained to learn the correspondence between an input pattern and the class to which it belongs, analogously to the popular backpropagation (BP) algorithm. The advantage of these ARTMAP architectures with respect to BP are mainly that they converge in a few training epochs (while BP converges in the order of thousands to even hundreds of thousands) and they are able to learn more complicated problems for which BP is inadequate [SeLA012]. ARTMAP is configured with both binary inputs or analog inputs and supervised training in real time and it requires two ART networks in its structure to function [NHAB017].

4.1.2.5 FuzzyART

Similarly introduced in 1991, a FuzzyART architecture was reported which extended the original ART1 functionality by generalizing its operators using fuzzy set theory concepts. The result is that FuzzyART can take analog valued patterns as input while keeping the original mathematical properties present in ART1 [SeLA012].

4.2 Computational Intelligence Algorithms Applied

4.2.1 ART1

The ART1 architecture is a massively parallel neural network pattern recognition machine which self organizes recognition codes in response to a sequence of binary valued input patterns. The system receives a sequence of binary valued input patterns clustering them into a set of categories in an unsupervised way [SeLA012].

4.2.1.1 ART1 Architecture

The topological structure of an ART1 system is shown in Fig. 4.1. It consists of two layers of neurons or processing cells named *F1*, *comparison layer*, and *F2*, *recognition layer*. Each neuron in layer *F1* receives the binary value of an input pattern. There are *N* neurons in layer *F1*. Hence, the input pattern I_i $(i = 1, ..., N)$ has N binary values '0' or '1' [SeLA012].

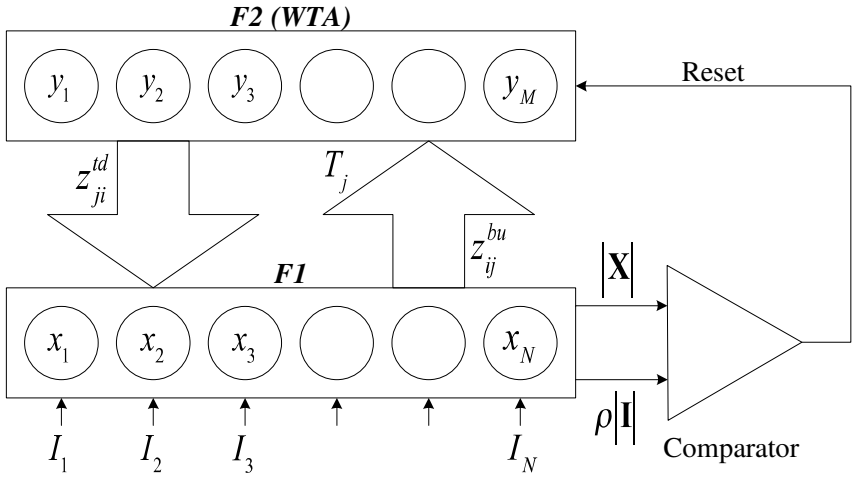


Fig. 4.1. Topological structure of the ART1 architecture. From [SeLA012].

Input patterns presented to layer *F1* cluster into categories, and a neuron in layer *F2* represents a possible category. Each neuron in layer *F1* is connected to all neurons in layer *F2* through *bottom-up synaptic connections* of strength z_{ij}^{bu}. Index i indicates that the connection goes from the i^{th} neuron in layer *F1* to the j^{th} neuron in layer *F2*. Bottom-up weights z_{ij}^{bu} are of continuous nature and they may take any value within the bounded interval [0, 1] [SeLA012].

The input to the j^{th} *F2* neuron is given by $\cap$

$$T_j = \sum_{i=1}^{N} z_{ij}^{bu} I_i, \text{ for } j = 1, \ ..., \ M \tag{4.1}$$

where M denotes the number of neurons in the *F2* layer. The terms T_j, known as *choice functions,* represent a certain "distance" between the stored pattern $z_j^{bu} \equiv z_{1j}^{bu}, \ z_{2j}^{bu}, \ ..., \ z_{Nj}^{bu}$, and input pattern $I \equiv (I_1, \ I_2, \ ..., \ I_N)$ [SeLA012].

Neurons in layer *F2* operate in such a way that their output y_j is always '0', except for the neuron receiving the largest T_j input from the *F1* layer. This *F2* neuron, let us call it *J*, has output '1' [SeLA012],

$$\begin{aligned} &y_J = 1 \text{ if } T_J = \max_j \{T_j\} \\ &y_{j \neq J} = 0 \end{aligned} \tag{4.2}$$

Each *F2* neuron is connected to all *F*1 neurons through *top-down synaptic connections* of strength z_{ji}^{td}, which are binary-valued. Thus, the i^{th} *F1* neuron input from the *F2* layer is [SeLA012]

$$V_i = \sum_{j=1}^{M} z_{ji}^{td} y_j = z_{Ji}^{td}, \text{ for } i = 1, \ ..., \ N \tag{4.3}$$

A *vigilance subsystem*, denoted as the comparator in Fig. 4.1, verifies the appropriateness of the *F2* neuron designating the active category. This vigilance subsystem compares the norm of vector $\mathbf{X} \equiv (x_1, \ x_2, \ ..., \ x_N)$, defined as [SeLA012]

$$x_i = V_i I_i \text{ or } \mathbf{X=V} \cap \mathbf{I=z}_J^{td} \cap \mathbf{I} \tag{4.4}$$

4.2.1.2 ART1 Dynamics

The time evolution of the state of all *F1* and *F2* neurons is governed by a set of time domain nonlinear and coupled differential equations, called *short-term memory equations*, and the present state of *F1* and *F2* neurons is called *short-term memory*. The time domain evolution of the set of weights z_{ij}^{bu} and z_{ji}^{td} is governed by another set of time domain nonlinear differential equations called *long-term memory equations*, and the set of values stored in weights z_{ij}^{bu} and z_{ji}^{td} is called *long-term memory* [SeLA012].

The time constant associated to the LTM equations is much slower than that of the STM equations. Consequently, if an input pattern **I** is presented to the *F1* layer, the STM settles first. If the input pattern **I** is held constant at the *F1* layer input until all STM equations and the vigilance subsystem settle, it is possible to describe the STM dynamics using an algebraic description of the steady state of the STM differential equations. Furthermore, if the input pattern **I** is held constant until LTM settles, then it is also possible to describe the LTM dynamics using algebraic equations that define the steady state of the LTM differential equations. It is in this case that the dynamic description of the ART1 architecture is referred to as "Fast Learning ART1". This type of description is the one used in this thesis. The Fast Learning algorithmic description of the ART1 architecture is shown in Fig. 4.2. Note that only two parameters are needed, ρ which is called the *vigilance parameter* and takes a value in the interval [0, 1], and parameter *L* which takes a value larger than '1' [SeLA012].

First, all interconnection weights are initialized. These weights store the knowledge or experience of the ART1 system. Therefore, after initialization they do not hold any information on categories, clusters, nor past input patterns provided. Bottom-up weights are initialized to $z_{ij}^{bu} = L/(L-1+N)$ and top-down weights to $z_{ji}^{td} = 1$. Now the system is ready to receive its first input pattern $\mathbf{I} = (I_1, I_2, ..., I_N)$ where I_i may be either '0' or '1'. At this point, the input to each neuron in the *F2* layer is computed [SeLA012],

$$T_j = \sum_{i=1}^{N} z_{ij}^{bu} I_i, \text{ for } j = 1, ..., M \tag{4.5}$$

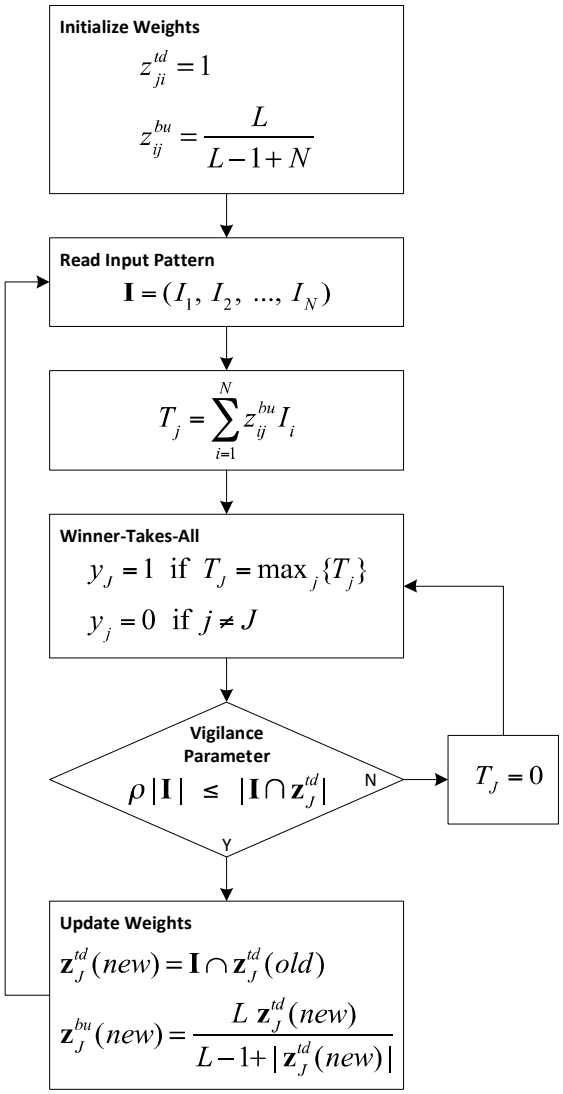


Fig. 4.2. Algorithmic description of ART1 functionality. From [SeLA012].

The neuron receiving the largest input T_j is activated in the *F2* layer, while all others are deactivated. Thus, if T_j is the maximum of all T_j inputs, then $y_J = 1$ and $y_{j \neq J} = 0$. Once an *F2* node is active, the vigilance subsystem checks if it is appropriate. The vigilance subsystem action is characterized by the vigilance parameter ρ set to a value in the interval [0, 1]. If the following condition is satisfied [SeLA012]

$$\rho \le \frac{|\mathbf{I} \bigcap \mathbf{z}_J^{td}|}{|\mathbf{I}|} \tag{4.6}$$

where

$$\begin{aligned} |\mathbf{I}| &= \sum_{i=1}^{N} I_i, \\ |\mathbf{I} \bigcap \mathbf{z}_J^{td}| &= \sum_{i=1}^{N} I_i z_{Ji}^{td} \end{aligned} \tag{4.7}$$

The active *F2* category neuron *J* is selected for LTM update. Otherwise, the *F2* neuron *J* is shut off, by making $T_J = 0,$ and a new *F2* neuron becomes active. The active neuron is checked by the vigilance subsystem and it is deactivated if it does not satisfy its condition. This process continues until an active *F2* neuron meets the vigilance criterion. Once a neuron in *F2* is found the bottom-up and top-down connection weights related to this node are updated according to [SeLA012]

$$\begin{aligned} z_{iJ}^{td}(new) &= z_{iJ}^{td}(old) I_i \\ z_{iJ}^{bu}(new) &= \frac{L\, z_{iJ}^{td}(new)}{L - 1 + |\, z_{iJ}^{td}(new)|} \end{aligned} \tag{4.8}$$

or, in vector notation [SeLA012]

$$\begin{aligned} \mathbf{z}_J^{td}(new) &= \mathbf{I} \bigcap \mathbf{z}_J^{td}(old) \\ \mathbf{z}_{iJ}^{bu}(new) &= \frac{L\, \mathbf{z}_J^{td}(new)}{L - 1 + |\, \mathbf{z}_J^{td}(new)|} \end{aligned} \tag{4.9}$$

Now the system is ready to receive the next input pattern [SeLA012].

If an *F2* category neuron *j* has not yet been chosen for category storage it is considered as an *uncommitted* neuron and its weights z_{ij}^{bu} and z_{ji}^{td}, $i = (1,\ 2,\ ...,\ N),$ still preserve their initialized values. On the other hand, if an *F2* neuron has already been selected, at least once, for storage, it is referred to as a *committed* neuron. Note that initially, since all weights are equal, the first time all *F2* inputs T_j are computed with equation (4.5), they are identical, and it is not possible to choose a maximum among them. This can be solved by making $M = n_c + 1,$ where n_c is the number of committed neurons. This way, initially $n_c = 0$ and $M = 1,$ which means that

only the first *F2* neuron is available for competition. As soon as this neuron is chosen for storage $n_c = 1$ and $M = 2,$ so that next time the competition is between one committed and one uncommitted neurons. Once the second neuron is chosen for storage $n_c = 2$ and $M = 3,$ and so on. The competition in the *F2* layer is always between the n_c committed neurons and one uncommitted neuron. Note also that an uncommitted neuron always satisfies the vigilance criterion for any $\rho \in [0,\ 1]$ because [SeLA012]

$$\frac{|\mathbf{I} \bigcap z_{j_{uncommited}}^{td}|}{|\mathbf{I}|} = \frac{|\mathbf{I}|}{|\mathbf{I}|} = 1 \geq \rho \tag{4.10}$$

Therefore if, at a given time, the maximum T_J corresponds to an uncommitted neuron, this neuron would be chosen for storage and become committed [SeLA012].

4.2.1.3 ART1 Properties

The ART1 architecture possesses interesting properties, which set it apart favourably from other clustering algorithms. Some of these properties are listed and explored subsequently [SeLA012].

4.2.1.3.1 Vigilance or Variable Coarseness

One of the most important characteristic features of ART1 is the possibility to externally tune the coarseness with which categories should be formed. ART1 contains a *vigilance subsystem*, controlled by a *vigilance parameter* $\rho,$ that provides this functionality [SeLA012].

The vigilance parameter ρ can be set to any value in [0, 1]. The smaller ρ is set, the more input patterns are clustered together into the same category, which means a higher generalization capability. The higher ρ gets causes more attention to be paid to differences among input patterns, which produces a larger number of categories [SeLA012].

By looking at the algorithmic description of ART1 in Fig. 4.2, one may wonder what the purpose of the vigilance subsystem might be. Why do one need to check whether or not

$$\rho |\mathbf{I}| \ \leq \ |\mathbf{I} \bigcap \mathbf{z}_J^{td}| \tag{4.11}$$

if category J has already been selected by the Winner-Takes-All? The answer is related to the *self-scaling* property discussed next [SeLA012].

4.2.1.3.2 Self-Scaling

This property allows features to be treated as either noise or signal, according to context. For example, consider two patterns $\mathbf{I}_a$ and $\mathbf{z}_a^{td}$ differing only in one feature, but it is a feature out of three. Therefore, it seems reasonable to classify each pattern as belonging to two different categories. In this case, the feature that makes the difference is considered a 'critical feature' or signal. Now, consider two patterns $\mathbf{I}_b$ and $\mathbf{z}_b^{td}$ differing in one feature, but now it is a mismatch of one feature out of 15. Therefore, it seems reasonable to classify them into the same category. In this case, the feature that makes the difference is considered as 'noise' [SeLA012].

Patterns $\mathbf{I}_a$ and $\mathbf{I}_b$ are considered new input patterns and patterns $\mathbf{z}_a^{td}$ and $\mathbf{z}_b^{td}$ characterize stored categories. If pattern $\mathbf{I}_a$ is presented to the *F1* layer and the category characterized by weight template $\mathbf{z}_a^{td}$ received the largest input T_J, then the vigilance subsystem would accept this category for pattern $\mathbf{I}_a$ if [SeLA012]

$$\rho \leq \frac{|\mathbf{I}_a \cap \mathbf{z}_a^{td}|}{|\mathbf{I}_a|} = \frac{2}{3} = 0.6666 \tag{4.12}$$

Consequently, if parameter ρ is initially set to a value higher than 0.666, pattern $\mathbf{I}_a$ would not be stored into the category characterized by weight template $\mathbf{z}_a^{td}$ [SeLA012].

For pattern $\mathbf{I}_b$ the condition would change to [SeLA012]

$$\rho \leq \frac{|\mathbf{I}_b \cap \mathbf{z}_b^{td}|}{|\mathbf{I}_b|} = \frac{14}{15} = 0.9333 \tag{4.13}$$

Hence, if for example the vigilance parameter is set to $\rho = 0.85$ patterns $\mathbf{I}_b$ and $\mathbf{z}_b^{td}$ would have been clustered together, but patterns $\mathbf{I}_a$ and $\mathbf{z}_a^{td}$ would not. These examples also illustrate the role of vigilance parameter ρ. The larger or closer it is to '1', the more categories are formed, and more attention is paid to the 'details' that distinguish the input patterns. The smaller ρ is, or closer to '0', less attention is be paid to 'small details' and more input patterns are be clustered into the same categories, resulting in a smaller number of categories and more generalization capability, for the same sequence of input patterns [SeLA012].

4.2.1.3.3 Self-Stabilization in a Small Number of Iterations

All the interconnection weights in the system that are subject to learning reach a *stationary* value after a finite number of presentations of a sequence of arbitrarily many and arbitrarily complex binary input patterns [SeLA012].

Initially, when a category *j* is uncommitted, it stores a template with all its elements $z_{ji}^{td} = 1$. When category *j* becomes committed by an input pattern **I**, it loses the elements z_{ji}^{td} such that the corresponding $\mathbf{I}_i = 0,$ [SeLA012]

$$z_{ji}^{td}(new) = z_{ji}^{td}(old) I_i \tag{4.14}$$

Hence, each category *j* progressively loses elements. However, the number of elements a category can lose is at most *N*. Consequently, if we denote as n_j the number of times the weight template $\mathbf{z}_j^{td}$ of category *j* is submitted to change, then [SeLA012]

$$n_j \leq N \tag{4.15}$$

If there are *M* categories in the system, the number of times the weights in the system are changed is limited to a maximum of $N \times M$. In practice, the system weights always stabilize in a reduced number of input pattern presentations [SeLA012].

4.2.1.3.4 On-line Learning

For many clustering algorithms, given a set of exemplars or input patterns, the clusters or categories are computed *off-Line*. If a new exemplar needs to be added, then the system knowledge has to be erased and retrained with the updated database. *off-Line* learning means that *learning phase* and *performing phase* are separate phases. In ART1 this does not happen. ART1 can be trained *on-Line*, in other words, it learns while it performs. Every time a new input pattern is given, ART1 answers with a category (either committed or uncommitted) and updates the weights that trigger this category to incorporate the new knowledge. This on-line learning property makes the ART1 an ideal candidate for *real-time* clustering [SeLA012].

4.2.1.3.5 Capturing Rare Events

The ART1 algorithm is able to learn and form clusters with input exemplars that appear very rarely. Thanks to its *on-line* learning capability, ART1 can learn a rare input pattern with

only a single exemplar presentation. Since the pattern is rare it solicits an uncommitted node. The rest of patterns, since they differ significantly from this rare one, would never choose its category for update, and consequently cause no alterations onto it [SeLA012].

4.2.1.3.6 Direct Access to Familiar Input Patterns

An input pattern **I** is said to have *direct access* to a stored category *j*, if this category is the first one chosen by the winner takes all (WTA) in the *F2* layer, and the vigilance criterion is met. As the human cognitive system, ART1 has the ability to quickly recognize an object which is familiar to the system. No matter how many recognition codes (or categories) the system may store, after stabilization the system always directly accesses the code of patterns that have been learned, or which are very similar to other input patterns learned previously [SeLA012].

4.2.1.3.7 Direct Access to Subset and Superset Patterns

Suppose that a learning process has produced a set of categories in the *F2* layer. Suppose that two of these categories, j_1 and j_2, are such that $z_{j_1}^{td} \subset z_{j_2}^{td}$ (this means that if $z_{j_1 i}^{td} = 1$ then it must be $z_{j_2 i}^{td} = 1$, but if $z_{j_1 i}^{td} = 0$ then $z_{j_2 i}^{td}$ can either be '0' or '1'). In this case $\mathbf{z}_{j_1}^{td}$ is a *subset template* of $\mathbf{z}_{j_2}^{td}$, or equivalently $\mathbf{z}_{j_2}^{td}$, is a *superset template* of $\mathbf{z}_{j_1}^{td}$. Mathematically, in vector notation, this can be expressed as [SeLA012],

$$\mathbf{z}_{j_1}^{td} \bigcap \mathbf{z}_{j_2}^{td} = \mathbf{z}_{j_1}^{td} \tag{4.16}$$

Consider two input patterns $\mathbf{I}^{(1)}$ and $\mathbf{I}^{(2)}$ such that [SeLA012],

$$\begin{aligned} \mathbf{I}^{(1)} &= \mathbf{z}_{j_1}^{td} \equiv (\mathbf{z}_{j_1 1}^{td},\ \mathbf{z}_{j_1 2}^{td},\ ...,\ \mathbf{z}_{j_1 N}^{td}) \\ \mathbf{I}^{(2)} &= \mathbf{z}_{j_2}^{td} \equiv (\mathbf{z}_{j_2 1}^{td},\ \mathbf{z}_{j_2 2}^{td},\ ...,\ \mathbf{z}_{j_2 N}^{td}) \end{aligned} \tag{4.17}$$

The *direct access to subset and superset* property assures that input $\mathbf{I}^{(1)}$ has *direct access* to category j_1 and that input $\mathbf{I}^{(2)}$ has Direct Access to category j_2 [SeLA012].

First, suppose input $\mathbf{I}^{(1)}$ is presented to the system. Computing the values of T_{j_1} and T_{j_2} [SeLA012],

$$T_{j_1} = \frac{L\sum_{i=1}^{N} I_i^{(1)} z_{j_1 i}^{td}}{L-1+\left|\mathbf{z}_{j_1}^{td}\right|} = \frac{L\left|\mathbf{I}^{(1)} \cap \mathbf{I}^{(1)}\right|}{L-1+\left|\mathbf{I}^{(1)}\right|} = \frac{L\left|\mathbf{I}^{(1)}\right|}{L-1+\left|\mathbf{I}^{(1)}\right|}$$
$$T_{j_2} = \frac{L\sum_{i=1}^{N} I_i^{(1)} z_{j_2 i}^{td}}{L-1+\left|\mathbf{z}_{j_2}^{td}\right|} = \frac{L\left|\mathbf{I}^{(1)} \cap \mathbf{I}^{(2)}\right|}{L-1+\left|\mathbf{I}^{(2)}\right|} = \frac{L\left|\mathbf{I}^{(1)}\right|}{L-1+\left|\mathbf{I}^{(2)}\right|} \tag{4.18}$$

Since $|\mathbf{I}^{(1)}| < |\mathbf{I}^{(2)}|$, it follows that $T_{j_1} > T_{j_2}$, (recalling that $L > 1$) and therefore category j_1 is selected by the *F2* layer. This category consequentially is also accepted by the vigilance subsystem because [SeLA012]

$$\frac{\left|\mathbf{I}^{(1)} \cap \mathbf{z}_{j_1}^{td}\right|}{\mathbf{z}_{j_1}^{td}} = \frac{\left|\mathbf{I}^{(1)}\right|}{\left|\mathbf{I}^{(1)}\right|} = 1 \geq \rho \tag{4.19}$$

On the other hand, if input pattern $\mathbf{I}^{(2)}$ is presented at the input [SeLA012],

$$T_{j_1} = \frac{L\sum_{i=1}^{N} I_i^{(2)} z_{j_1 i}^{td}}{L-1+\left|\mathbf{z}_{j_1}^{td}\right|} = \frac{L\left|\mathbf{I}^{(2)} \cap \mathbf{I}^{(1)}\right|}{L-1+\left|\mathbf{I}^{(1)}\right|} = \frac{L\left|\mathbf{I}^{(1)}\right|}{L-1+\left|\mathbf{I}^{(1)}\right|}$$
$$T_{j_2} = \frac{L\sum_{i=1}^{N} I_i^{(2)} z_{j_2 i}^{td}}{L-1+\left|\mathbf{z}_{j_2}^{td}\right|} = \frac{L\left|\mathbf{I}^{(2)} \cap \mathbf{I}^{(2)}\right|}{L-1+\left|\mathbf{I}^{(2)}\right|} = \frac{L\left|\mathbf{I}^{(2)}\right|}{L-1+\left|\mathbf{I}^{(2)}\right|} \tag{4.20}$$

Since function $Lx/(L-1+x)$ is an increasing function with x, it results that $T_{j_2} > T_{j_1}$ and category j_2 is then chosen by the *F2* layer, and accepted by the vigilance subsystem since [SeLA012]

$$\frac{\left|\mathbf{I}^{(2)} \cap \mathbf{z}_{j_2}^{td}\right|}{\mathbf{z}_{j_2}^{td}} = \frac{\left|\mathbf{I}^{(2)}\right|}{\left|\mathbf{I}^{(2)}\right|} = 1 \geq \rho \tag{4.21}$$

4.2.1.3.8 Biasing the Network to Form New Categories

Independent of the vigilance parameter ρ, parameter '*L*' biases the tendency of the network to form a smaller or larger number of categories. In particular, parameter *L* biases the tendency of the network to select a new uncommitted category before a committed one. When an input pattern **I** is presented, an uncommitted neuron is chosen before a committed one *j* if [SeLA012]

$$\frac{\left|\mathbf{I}\cap\mathbf{z}_j^{td}\right|}{L-1+\left|\mathbf{z}_j^{td}\right|}<\frac{\left|\mathbf{I}\right|}{L-1+N} \tag{4.22}$$

This inequality is equivalent to [SeLA012]

$$L-1>\frac{N\left|\mathbf{I}\cap\mathbf{z}_j^{td}\right|-\left|\mathbf{z}_j^{td}\right|\left|\mathbf{I}\right|}{\left|\mathbf{I}\right|-\left|\mathbf{I}\cap\mathbf{z}_j^{td}\right|} \tag{4.23}$$

Therefore, increasing L increases the tendency to select an uncommitted neuron before a committed one [SeLA012].

4.2.2 FuzzyART

FuzzyART is a clustering neural network architecture which self-organizes recognition codes in response to sequences of analog or binary input patterns [SeLA012].

4.2.2.1 FuzzyART Architecture

The FuzzyART architecture is shown in Fig. 4.3. It has the same structure as the ART1 system shown in Fig. 4.1. It consists of two layers of computing cells or neurons *F1* and *F2*, and a vigilance subsystem controlled by an adjustable vigilance parameter $\rho\in[0,\ 1]$ [SeLA012].

Layer *F1* is the input layer composed of N input cells. Each input cell receives a component $I_i\in[0,\ 1]$ of the continuous input vector $\mathbf{I}=(I_1,\ I_2,\ ...,\ I_N)$. Layer *F2* is the category layer. It is composed of M cells, each one representing a possible category. Each category cell receives an input T_j. Each *F1* layer neuron i is connected to each *F2* layer neuron j by a synaptic connection of weight z_{ij}^{bu}. Each *F2* layer neuron j is connected to each *F1* layer neuron i by a synaptic connection of strength z_{ji}^{td}. In FuzzyART $z_{ij}^{bu}=z_{ji}^{td}$. Consequently, from now on the weights as $z_{ij}=z_{ij}^{bu}=z_{ji}^{td}$ [SeLA012].

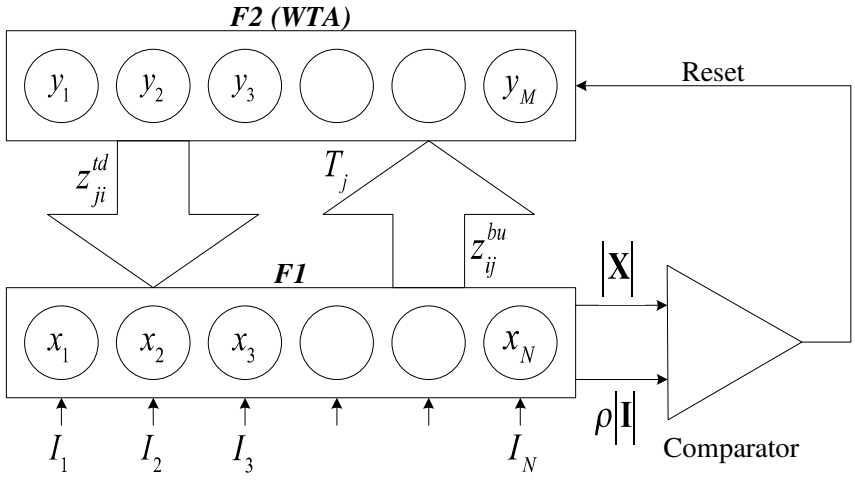


Fig. 4.3. Topological structure of the FuzzyART architecture. From [SeLA012].

The main differences between the ART1 and FuzzyART architectures are: (i) The input vectors are continuous in nature. That is $\mathbf{I} = (I_1,\ I_2,\ ...,\ I_N)$ is an *N* dimensional vector with each component $I_i \in [0,\ 1]$, (ii) there is only one set of analog valued weight vectors $\mathbf{z}_j = (z_{1j},\ z_{2j},\ ...,\ z_{Nj})$ for $j = 1,\ 2,\ ...,\ M$, and (iii) in the computation of the choice functions T_j, the learning rule, and the vigilance criterion, the intersection operation $\cap$ (binary AND) is substituted by the fuzzy MIN operator $\wedge$ (continuous AND) [SeLA012].

4.2.2.2 FuzzyART Operation

Figure 4.4 shows the flow diagram of FuzzyART. Initially in FuzzyART, all the interconnection weights z_{ij} are set to '1'. When a continuous input vector $\mathbf{I} = (I_1,\ I_2,\ ...,\ I_N)$ is applied to the system, each *F1* layer neuron receives a component $I_i \in [0,\ 1]$. Then each *F2* layer category neuron receives an input T_j, which is a measurement of the similarity between the continuous-valued input pattern $\mathbf{I}$ and the continuous-valued weight template $\mathbf{z}_j = (z_{1j},\ z_{2j},\ ...,\ z_{Nj})$ stored in category j [SeLA012],

$$T_j = \frac{|\mathbf{I} \wedge \mathbf{z}_j|}{\alpha_{FA} + |\mathbf{z}_j|} \tag{4.24}$$

where $\wedge$ is the fuzzy MIN operator defined by $(\mathbf{X} \wedge \mathbf{Y})_i = \min(X_i, Y_i)$, $|\mathbf{X}|$ is the ℓ^1 norm $|\mathbf{X}| = \sum_{i=1}^{N} |X_i|$, and α_{FA} is a positive parameter called 'choice parameter' [SeLA012]. The choice parameter α_{FA} takes values in the interval $(0, \infty)$ [GFBH996]. A small choice parameter of $\alpha_{FA} = 0.01$ is sometimes used, as it has been shown that the clustering performance is generally robust to this parameter [MeTW013]. However, a value of $\alpha_{FA} = 1$ is also suggested. The j^{th} *F2* cell gives an output y_j either with a value of '1' if this cell is receiving the largest T_j input or '0' otherwise [SeLA012].

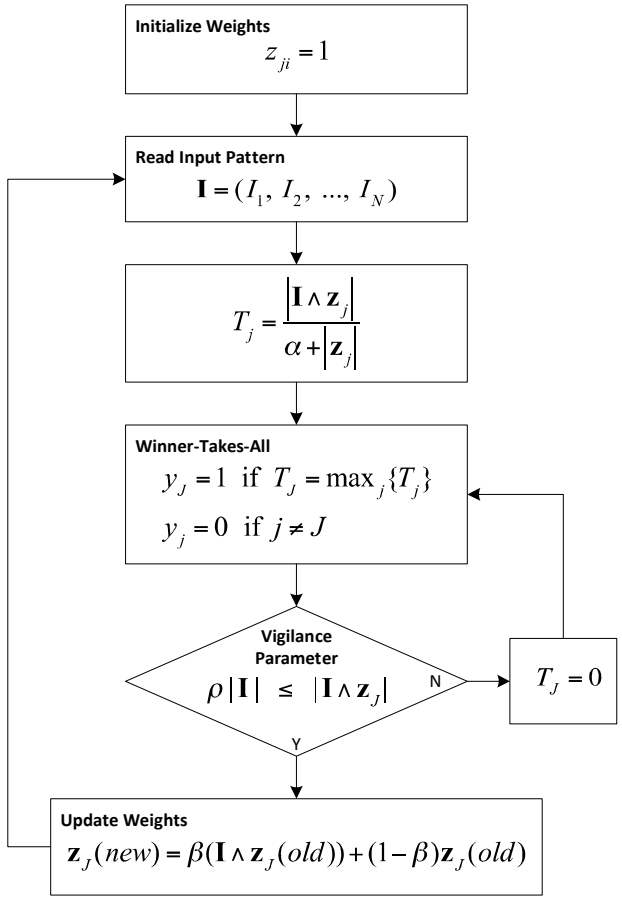


Fig. 4.4. Algorithmic description of FuzzyART functionality. From [SeLA012].

$$y_J = 1 \text{ if } T_j = \max_j \{T_j\}$$
$$y_{j \neq J} = 0 \text{ otherwise} \tag{4.25}$$

In this way, the *F2* or WTA layer selects the category *J* whose stored template $\mathbf{z}_J$ most closely resembles input pattern $\mathbf{I}$ according to the similarity criterion $T_j = |\mathbf{I} \wedge \mathbf{z}_j| / (\alpha + |\mathbf{z}_j|)$ [SeLA012].

For the winning category *J*, the vigilance subsystem checks the condition [SeLA012],

$$\rho |\mathbf{I}| \leq |\mathbf{I} \wedge \mathbf{z}_J| \tag{4.26}$$

If this condition is not true, category *J* is discarded by making $T_J = 0$. In the next iteration, layer *F2* selects the category with maximum T_j, and the vigilance criterion defined in the equation above is verified again. The search process continues until layer *F2* finds a winning category capable of fulfilling the vigilance criterion [SeLA012].

When a category *J* meeting the vigilance criterion is activated, its weights $\mathbf{z}_J$ are updated according to the rule [SeLA012]

$$\mathbf{z}_J(new) = \beta_{FA}(\mathbf{I} \wedge \mathbf{z}_J(old)) + (1 - \beta_{FA})\mathbf{z}_J(old) \tag{4.27}$$

where β_{FA} is the parameter known as 'learning rate', which is confined to $\beta_{FA} \in [0, 1]$ [SeLA012].

4.2.2.2.1 Fast-Commit Slow-Recode Option

For efficient coding of noisy input sets, it is useful to set $\beta_{FA} = 1$ when the learning category *J* is an uncommitted neuron (fast-commit) and $\beta_{FA} < 1$ after the category is committed (slow-recode) [SeLA012].

With this option, the first-time category *J* becomes active $\mathbf{z}_J(new) = \mathbf{I}$, allowing an adequate response to inputs that may occur only rarely and in response to which a quick and accurate performance may be needed [SeLA012].

When a committed category needs to be updated $\beta_{FA} < 1$, thus preventing features that have been incorporated into it from being deleted when a noisy or partial input appears. Only a persistent change in a feature allows deleting it from a category template [SeLA012].

4.2.2.2.2 Input Normalization Option

A category proliferation problem may occur in some ART systems when the norm of input vectors $|\mathbf{I}|$ can be made arbitrarily small. This problem of category proliferation is avoided in the FuzzyART system when the input vectors are normalized before being processed by the system [SeLA012].

An input vector $|\mathbf{I}|$ is said to be normalized when there exists a constant $\gamma_{no} > 0$, such that $|\mathbf{I}| = \gamma_{no}$ for all input vectors $\mathbf{I}$. One way of normalizing the input vectors could be to divide each incoming vector $\mathbf{a}$ by its norm $|\mathbf{I}| = \mathbf{a} / |\mathbf{a}|$. However, this method may lose the information about the input amplitude. Consider, for example, the two-dimensional incoming vectors $\mathbf{a}_1 = (1,\ 1)$ and $\mathbf{a}_2 = (0.1,\ 0.1)$. The first vector $\mathbf{a}_1$ indicates a high value of the two vector components, while the second vector $\mathbf{a}_2$ indicates a low value of both vector components. However, both vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ produce the same normalized input vector $\mathbf{I} = (1/2,\ 1/2)$ and is treated by the system in the same way [SeLA012].

To avoid loss of information, Grossberg and Carpenter proposed the complement coding rule for the normalization of the input vectors. This rule consists of expanding an *N*-dimensional incoming vector $\mathbf{a} = (a_1,\ a_2,\ ...,\ a_N)$ to a 2*N*-dimensional vector defined by [SeLA012]

$$\mathbf{I} = (\mathbf{a},\ \mathbf{a}^c) = (a_1,\ a_2,\ ...,\ a_N,\ a_1^c,\ a_2^c,\ ...,\ a_N^c) \tag{4.28}$$

where $a_i^c = 1 - a_i$ for $i = 1,\ 2,\ ...,\ N$. This way, all the input vectors $\mathbf{I}$ are normalized [SeLA012]

$$|\mathbf{I}| = |\mathbf{a},\ \mathbf{a}^c| = \sum_{i=1}^{N} a_i + \sum_{i=1}^{N} a_i^c = \sum_{i=1}^{N} a_i + N - \sum_{i=1}^{N} a_i = N \tag{4.29}$$

but the amplitude information is preserved. The two vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ discussed above produce two different input vectors $\mathbf{I}_1 = (1,\ 1,\ 0,\ 0)$ and $\mathbf{I}_2 = (0.1,\ 0.1,\ 0.9,\ 0.9)$ and are treated by the system in a different way [SeLA012].

In the case of a FuzzyART system with the complement coding option, the weight vectors $\mathbf{z}_j$ are also expanded to 2*N*-dimensional vectors [SeLA012],

$$\mathbf{z}_j = \left(\mathbf{u}_j,\ \mathbf{u}_j^c\right) \quad j = 1,\ 2,\ ...,\ M$$

(4.30)

which are initially set to $\mathbf{z}_j = (1, ..., 1)$, so that $\mathbf{u}_j = (1, ..., 1)$, and $\mathbf{v}_j = (1, ..., 1)$. The same learning rule defined by $\mathbf{z}_J(new) = \beta(\mathbf{I} \wedge \mathbf{z}_J(old)) + (1-\beta)\mathbf{z}_J(old)$ is still valid for updating the $\mathbf{z}_j$ vectors [SeLA012]. Appendix E includes an explanation about the geometrical interpretation of complement coding.

4.3 Summary

The adaptive resonance theory underpinnings have been introduced. Specific equation models are discussed. The topological distinctions for different generations of ART are presented. The computational intelligence algorithms, ART1 and FuzzyART, applied in this research are explored in depth.

CHAPTER V

DESIGN OF EXPERIMENTS

This chapter is dedicated for describing the experiments details. This description follows the theoretical framework introduced in the previous chapters. The experiments are described with the required detail so that they are reproducible. The experimental environment available is also described. This description previous to executing the experiments provides completeness for the specific route taken in this research to this point. The results of experiments after conducting them are described in the next chapter.

5.1 Experimental Platform

The experiments are executed in two computer systems: (i) A server with an Intel Xeon dual (2.93 GHz) microprocessor, 48 GB of RAM memory, Windows Server 2008 R2 Enterprise, and MATLAB R2019a; and (ii) A Mac Pro with a 2 Quad-Core Intel Xeon (2.4 GHz) microprocessors, 12 GB of RAM memory, MacOS X High Sierra (10.13.2), and MATLAB R2019a.

5.2 Experiments Design

The distinct experiments conducted in this section follow directly from the research core. These experiments are prepared in a modular and incremental fashion so that the relevant points are serially connected in their respective descriptions. A high degree of care has been put into the experiments design for having a concrete and very complete analysis framework in which the research questions would be covered extensively. The experiments addressed here are conducted in a dataset that contains a real DDoS attack. The dataset utilized in this research contains enough features (*e.g.*, precursors, attack flows, aggregated attack flows, genuine traffic flows, and an undisclosed attack) to be considered an example of a preliminary "gold" standard for detection of anomalies caused by DDoS. Additionally, this dataset is understood and one DDoS

attack is annotated and extensively described by the source. The full description of the details of such DDoS attack is included in Appendix F [Depa013].

The notation introduced in Ch. II is followed in the experiments design. The DDoS dataset analysed in this research with a traffic length N (60 million). The total stream flows are compounded whether they are either attack flows or genuine flows as defined by

$$S_F = A_i + G_i \tag{5.1}$$

where S_F denotes all streams present in the traffic under analysis, A denotes attack flows, G denotes genuine flows, and $i \geq 1$ represents multiple attack or genuine flows. All flows, total stream, attack, or genuine, are sampled within an arbitrary small time interval.

5.2.1 Dataset Insight

5.2.1.1 Dataset Packet Count Integration

The number of packets in the total flows stream are integrated within a time interval of 100 ms [Yu014]. All packets falling into this time interval are accumulated into a packet count. S_F is further simplified to S when singling out a specific component of the traffic as illustrated in the next case. The number of packets is then represented by the data sequence $S.p[n]$, where n denotes the n^{th} element in the sampled data sequence. Hence, a realization of the total stream flows packet count is represented by

$$S.p[n] = \{s.p[1],\ s.p[2],\ ...\} \tag{5.2}$$

5.2.1.2 Dataset Packet Length Integration

Similarly, from the dataset [Depa013], the packets lengths can be integrated within a 1 s interval. The length of the number of packets falling into these intervals is accumulated. This accumulated value is utilized to specify the data rate in B/s in this time series. A realization of the total stream flows packet length is represented by

$$S.l[n] = \{s.l[1],\ s.l[2],\ ...\} \tag{5.3}$$

5.2.1.3 DDoS Attack Packet Count Integration

The number of DDoS attack packets, considering all the attack flows being aggregated,

are integrated within a time interval of 100 ms [Yu014]. All packets falling into this time interval are accumulated into an attack packet count. The number of all aggregated DDoS attack packets is then represented by the data sequence $A.p[n]$, where n denotes the n^{th} element in the sampled data sequence. Hence, a realization of the DDoS attack packet count is represented by

$$A.p[n] = \{a.p[1],\ a.p[2],\ ...\} \tag{5.4}$$

5.2.1.4 DDoS Attack Packet Length Integration

Likewise, the DDoS attack packets lengths can be integrated within a 1 s interval. The length of the number of DDoS attack packets falling into these intervals is accumulated. This accumulated is utilized to specify the data rate in B/s in the DDoS attack time series. A realization of the DDoS attack flows packet length is represented by

$$A.l[n] = \{a.l[1],\ a.l[2],\ ...\} \tag{5.5}$$

5.2.1.5 DDoS Attack Flows Packet Count Integration

The number of packets per DDoS attack flow, a per bot contribution to the overall DDoS attack, are integrated within a time interval of 100 ms [Yu014]. All packets falling into this time interval are accumulated into a DDoS attack flow packet count. The number of packets per DDoS attack flow is then represented by the data sequence $A_i.p[n]$, where n denotes the n^{th} element in the sampled data sequence and i denotes the i^{th} DDoS attack flow. Hence, a realization of a DDoS attack flow packet count is represented by

$$A_i.p[n] = \{a_i.p[1],\ a_i.p[2],\ ...\} \tag{5.6}$$

For the particular dataset subject of this study, the botmaster had the IP address: 145.233.157.236. This botmaster coordinated six bots (IP addresses: 145.233.157.224, 145.233.157.228, 145.233.157.232, 145.233.157.233, 145.233.157.234, and 145.233.157.235) contributing DDoS attack flows to the DDoS attack. The DDoS attack flows packet counts are identified by $A_1.p[n]$, $A_2.p[n]$, $A_3.p[n]$, $A_4.p[n]$, $A_5.p[n]$, and $A_6.p[n]$, which is in correspondence with the IP addresses listed beforehand. For more information on the specifics of the DDoS dataset, please refer to Appendix F. The six bot DDoS attack flows contributions are isolated from the dataset traffic in order to verify the botnet fingerprint.

5.2.1.6 DDoS Attack Flows Length Integration

Also, the DDoS attack flows packets lengths are integrated within a 1 s interval. The length of the number of DDoS attack flows packets falling into these intervals is accumulated. This accumulation is utilized to specify the data rate in B/s for the DDoS attack flows time series. Following the same criteria for the IPs list described previously, a realization of the DDoS attack flows packet length is represented by

$$A_i.l[n] = \{a_i.l[1],\ a_i.l[1],\ ...\} \tag{5.7}$$

5.2.2 Implementation and Validation of VFD as Reference Methodology

5.2.2.1 VFD Validation through White Noise

In order to verify the correct implementation of the VFD algorithm, it is necessary to validate that it produces results that correspond to known signals. Two white noise signals are generated with uniform and Gaussian probability distribution functions. The length for both signals is 10 million samples. After these signals are generated, they are subjected to determine their fractal dimension with the VFD. Since white noise is considered a space-filling curve, it is expected that a fractal dimension with a value of two would be provided upon verification. Obtaining a fractal dimension of two would confirm that the VFD algorithm is correctly implemented.

The importance of validating the VFD implementation is necessary because it is planned to be used in the continuing stages of the proposed research. Hence, the need for verification of this focal algorithm.

5.2.3 Internet/Network Traffic Pipeline: Signal Conditioning, Analysis, Feature Extraction, and Classification via Adaptive Resonance Theory

Utilizing the Protected Repository for the Defense of Infrastructure Against Cyber Threats (PREDICT) DDoS dataset, having a minimal sampling period of 1 μs, a signal $S.l[n]$ with a specific integration time of $n = 1.0486$ s is created. This integration time of 1.0486 s is derived from a frame containing 256 samples, which are derived from frames of 4,096 of original DDoS dataset sampled at 1 μs (256 frames times 4,096 frames sampled at 1 μs cause an

integration time of 1.0486 s). The necessary number of samples in a time frame is 4,096 for allowing the proposed multiscalors methodology to work properly. Additionally, highlighting the presence of a hit and run (H&R) DDoS attack in this dataset requires a value close to 1 s for achieving a better visual perception. The signal $S.l[n]$ is then digitally processed for denoising, non-linear filtering, and quantization. At this stage, once the multiscalors epiphenomena have been quantized a feature vector is set for classification through ART.

5.2.4 Feature Extraction

5.2.4.1 Selected Operators Applied through Multiscalors

Two statistical moments, variance and skewness, are used with the methodology "multiscalors" previously introduced in this thesis. Since statistical moments provide relevant statistical properties about a signal stemming from a given process, the variance and skewness reflect the spread and how biased a pmf is to either side. Since Internet traffic resembles the probability distribution function (pdf) of Lévy walks, the two referred statistical moments are appropriate. Extensive publications, related to the resemblance between Internet traffic pmfs and Lévy walks, by the author of this thesis are available in the literature ([TeKi016a], [TeKi016b], [TeKi016c], and [TeKi016d]). Hence, the multiscalors methodology utilizing two operators, variance and skewness, is applied to frames of 4,096 samples in size from the DDoS dataset with a sampling period of 1 μs. From each frame of 4,096 samples, the multiscalors would provide seven points, which correspond to the vels size from 2^1 to 2^7.

These experiments are very important because they show the application of two operators, which have been traditionally used only in monoscale analysis, in multiscale analysis. Once the multiscalors are obtained, secondary methodologies are utilized to analyse the multiscalor components. These are described next.

5.2.4.2 Experiments with Selected Signal Analysis Methodologies Applied to Multiscalors Components

The successive application of multiscalors in frames, 4,096 samples in size, creates seven components (streams) of data. Each of these multiscalors components (streams) is further sliced

into frames of 256 samples in length so that a secondary operator is utilized. The three secondary operators selected are cumulative sum, ZCR, and Shannon entropy. Publications, by the author of this thesis, portraying the use of ZCR are available in the literature ([TeKi016a], [TeKi016b], [TeKi016c], and [TeKi016d]).

The selection of the sizes of the primary frame (where multiscalors is applied to a 4,096 samples frame in size, which are sampled at $1\ \mu s$) and the secondary frame (where the three secondary operators are applied to a 256 samples frame in size) reflect the dynamics of the Internet traffic in a time of 1.0486 s. A major significance of the usage of the linked frames (by subsequently applying multiscalors and secondary operator) is achieving a compression factor of 1.0486×10^{6} for the extracted features prior to using machine learning. It is important to highlight that bigger sizes (*i.e.*, 1,024 and 4,096 samples) for the secondary frame were considered and experiments were also conducted on them, but these are not included in this document. The only size that is documented in this thesis is of 256 samples in the secondary frame for detecting the presence on a DNS amplification DDoS attack and these reasons are considered: (i) It is the worst-case scenario considered in this research due to the smaller number of samples involved; (ii) this frame size enables the ART neural networks to achieve a detection time close to one second depending on the computing power required to make a decision. ART based neural networks are favoured in this research due to the inherent unsupervised nature and overperforming alternate supervised neural networks which typically require hundreds of thousands of epochs for training and good quality training sets *a priori*. Hence, ART neural networks are suitable candidates for real-time applications; and (iii) considering that bigger frame sizes would provide smoother and even more compressed features to the ART neural networks at the expense of delaying detection (*e.g.*, four seconds if a frame size of 1,024 samples is used and 16 seconds with a frame size of 4,096 samples).

5.2.5 Feature Classification

5.2.5.1 Preparation of Feature Vector for ART

Upon achieving the compression of the raw signal through multiscale and polyscale analysis, the features extracted from the multiscalors epiphenomena are assembled into a feature

vector. The vector would then be subject for classification through computational intelligence utilizing ART. In order to have this feature vector ready for ART, each feature is exposed to denoising, non-linear filtering, and quantization.

The feature vector for ART bundles 42 quantized scalars (each in a four bits long binary representation) entirely, which are identified in detail as: Seven scalars from the cumulative sum applied to the variance multiscalor, seven scalars from the ZCR applied to the variance multiscalor, seven scalars from entropy applied to the variance multiscalor, seven scalars from the cumulative sum applied to the skewness multiscalor, seven scalars from the ZCR applied to the skewness multiscalor, and seven scalars from entropy applied to the skewness multiscalor.

5.2.5.2 Preparation of Feature Vector for FuzzyART

An alternate computational intelligence algorithm that can be used in the context of ART is known as FuzzyART. FuzzyART spares quantizing the signal, which saves time in the preparation of a feature vector. The feature vector for FuzzyART bundles 42 non-quantized scalars entirely. This arrangement is described previously for ART.

For both ART1 and FuzzyART the order of the scalars is inconsequential as the sensitivity parameter can be set to fingerprint minute details in distinct patterns. From the two vectors containing 42 scalars each, alternate vectors can be derived. Such refinement has not been considered within this research as it is a research problem that falls outside the scope defined.

5.2.5.3 Classification Through ART

Once the feature vector for the neural network ART1 is prepared and its scalar components are translated into binary representation, it is run through the MATLAB implementation of ART1. The ART1 neural network has been fully described previously in section 5.7.1. This experiment aims to determine a quantitative insight about the Internet/network traffic dataset under analysis.

Since ART1 has only one parameter, the *vigilance parameter* ρ, a rule of thumb is followed to set its value to $\rho = 0.9$. If this value would yield no successful classification results, alternate values in the interval [0, 1] would be chosen and experimented with until a conclusive

result is achieved.

5.2.5.4 Classification Through FuzzyART

Correspondingly, the feature vector for the neural network FuzzyART is readied and value representation of its scalar components is kept unaltered because FuzzyART is capable of handling continuous value representations. Afterwards, the corresponding feature vector is subjected to the FuzzyART neural network implementation in MATLAB. Similarly, this trial targets getting quantitative insight about the Internet/network dataset and providing an alternative point of comparison with a neural network based on ART.

FuzzyART also requires only one parameter, the *vigilance parameter* ρ, for creating classes for recognizing the patterns within the traffic in this specific application. A similar criterion is followed for setting the initial value of the vigilance parameter and finding an optimal value in the interval [0, 1] that provides conclusive results about the presence of malicious traffic.

Since the raw Internet traffic has been compressed significantly by a factor of 1.0486×10^{6}, it is expected that the ART1 and FuzzyART neural networks (already capable of surpassing alternative neural networks, which in comparison require a high number of epochs for their training) are a good fit for running in real-time.

5.3 Summary

The available computing resources available to analyze the PREDICT dataset have been described. An in depth approach to further delve into details of the dataset has been provided. This approach starts with the raw traffic (mixture of genuine and attack traffic), the compound attack traffic, and the isolation of individual attack flows. For these three cases, the integration of the packet counts and the packet lengths over periods of time of 100 ms and 1 s are considered respectively. The variance fractal dimension validation is carried by using white noise with two distinct distributions (*i.e.*, Uniform and Gaussian). The Internet traffic analysis pipeline is presented, where the key elements are feature extraction and classification through ART. The two statistical moments, variance and skewness, application as operators through multiscalors is described. The application of three secondary operators, cumulative sum, ZCR, and Shannon

entropy, onto the multiscalors components is proposed. The preparation of the feature vectors for the machine learning models, ART1 and FuzzyART, are addressed. The expected classification mechanism for both models is also discussed.

CHAPTER VI

EXPERIMENTAL RESULTS AND DISCUSSION

The previous chapter presented the design of experiments considered in this study. Eight experiments are performed and shown throughout this chapter. Six of these experiments embrace the approach for revealing the individual contributions of bots. It shall be recalled that in order to achieve this, the raw traffic, the compound attack traffic, and the individual attack flows are analyzed. The two cases of the compound attack traffic and individual attack flows require isolating attack packets from the dataset. The individual attack flows packet counts, and data rates are analyzed to see if a contribution fingerprint to the overall attack traffic is observed. This is the utmost goal pursued in this segmentation of traffic. Lastly, two experiments for validating the variance fractal dimension are shown at the end. These experiments consider the usage of known signals as is the case of white noise, which was generated using two distinct probability distribution functions (*i.e.*, Uniform and Gaussian).

6.1 Dataset Packet Count Integration

The traffic packets represented by the data sequence $S.p[n]$ are integrated over a time interval of 100 ms. Each point in Fig. 6.1 represents the packet count over this interval of time. This figure resembles normal behaviour of Internet traffic, which is noise like. By integrating the traffic further through a moving average filter with 128 coefficients, a smoother curve is seen in blue. It is difficult to identify the DDoS attack taking place at this point by visually inspecting Fig 6.1. It is known *a priori* that the DDoS attack takes place between the 446.9 seconds and 1369.3 seconds according to the PREDICT dataset information. Knowing this and paying attention to the figure it is possible to see a hump in the averaged packet count between the start and end of the DDoS attack.

Almost at the end of the waveform there are two spikes that do not belong to the DDoS

attack, but these are apparently part of genuine traffic. This provides an idea of how challenging it is to identify DDoS attacks based on methods that could consider energy alone and not ITB approaches.

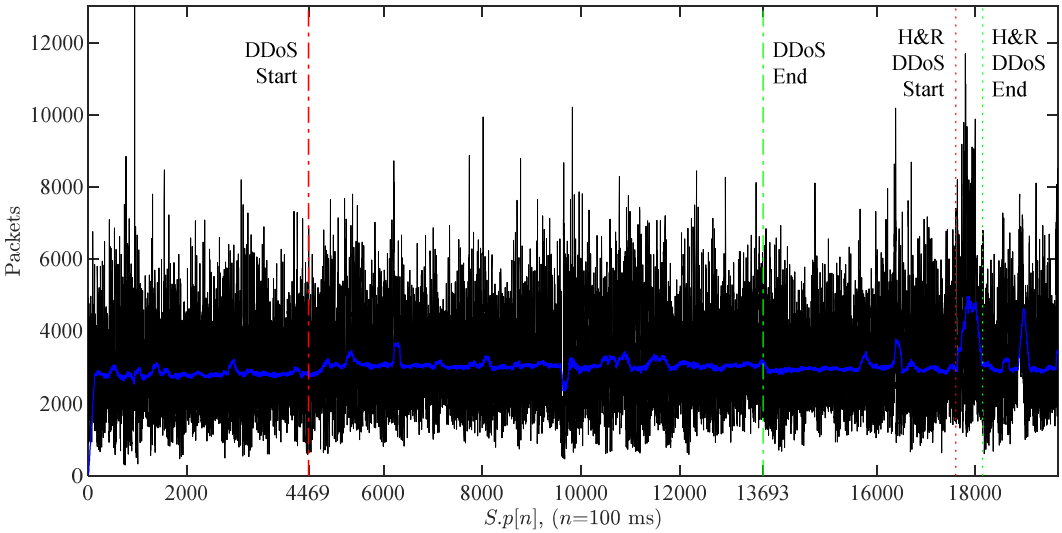


Fig. 6.1. Packets count of traffic in 100 ms time intervals. The packets counts averages are shown as blue coloured waveform.

6.2 Dataset Packet Length Integration

The integration of the packets data rate $S.l[n]$ over a time interval of one second is shown in Figure 6.2. The protuberance between the start and end of the DDoS attack is more visible and the spikes belonging to normal traffic are even more pronounced. Some other minor spikes start to pop up in different positions as well.

6.3 DDoS Attack Packet Count Integration

Figure 6.3 depicts the traffic belonging only to the DDoS attack $A.p[n]$. This is why there is no traffic before the start and after the end of the DDoS attack. It is noticeable in this figure of the regularity of the packets count, which is steady at rate of 120 packets per 100 ms. It is also noticeable that at the beginning of the DDoS attack there is some sort of preliminary signature and then the packet count for the attack becomes steady.

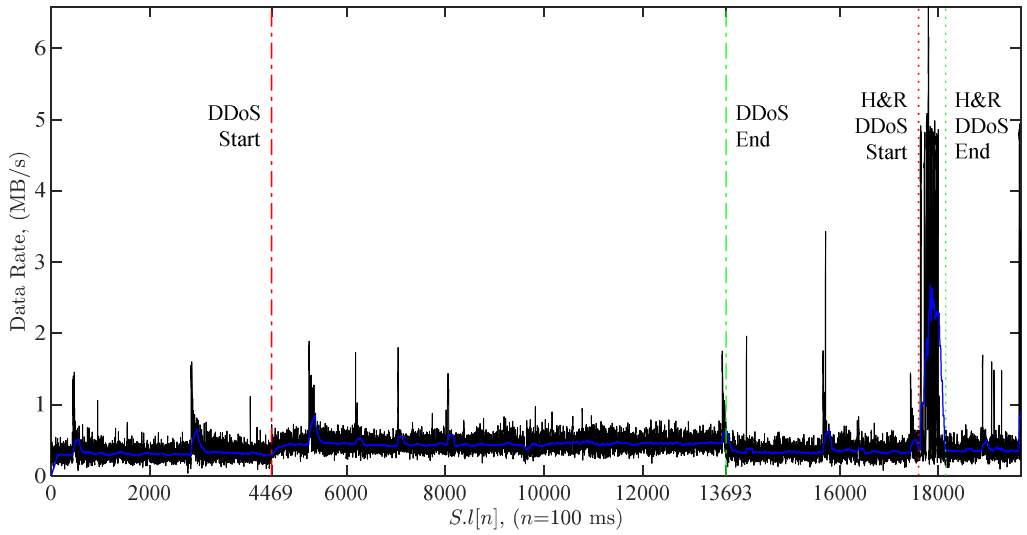


Fig. 6.2. Traffic data rate in 1 s time intervals. The traffic data rate averages are shown as a blue coloured waveform.

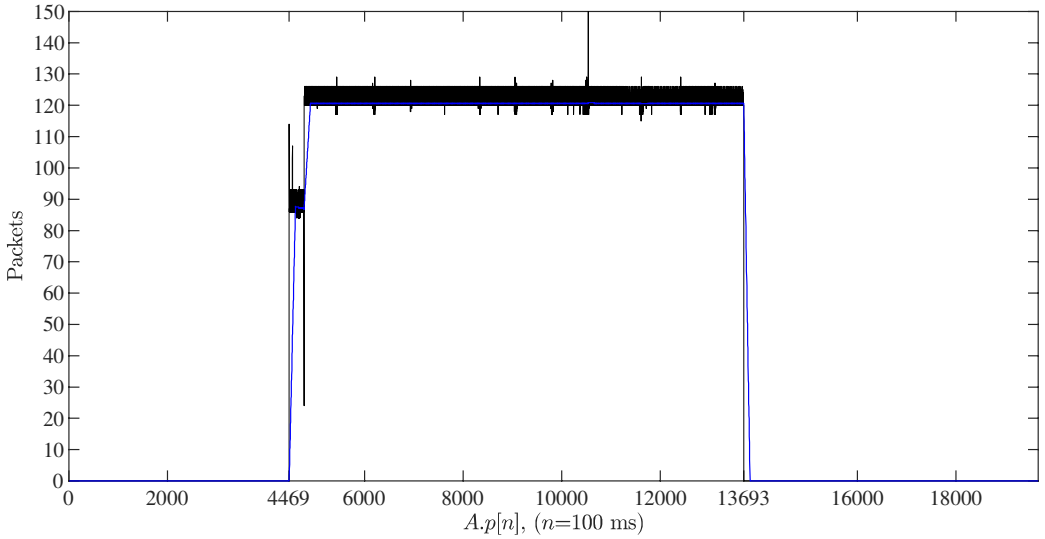


Fig. 6.3. DDoS attack packets counts in 100 ms time intervals. The DDoS attack packets counts averages are shown as a blue coloured waveform.

6.4 DDoS Attack Packet Length Integration

The integration of the attack packets lengths $A.l[n]$ is showcased in Fig. 6.4. A similar behaviour can be seen as in the integration of the attack packets counts waveform discussed just

previously. Also seen is a preliminary transient signature before the attack becomes steady. Once this transient goes by the data rate becomes steady in a value of 0.1296 MB/s. Similarly, the absence of data before and after the attacks takes place, follows from the fact that the genuine traffic has been removed.

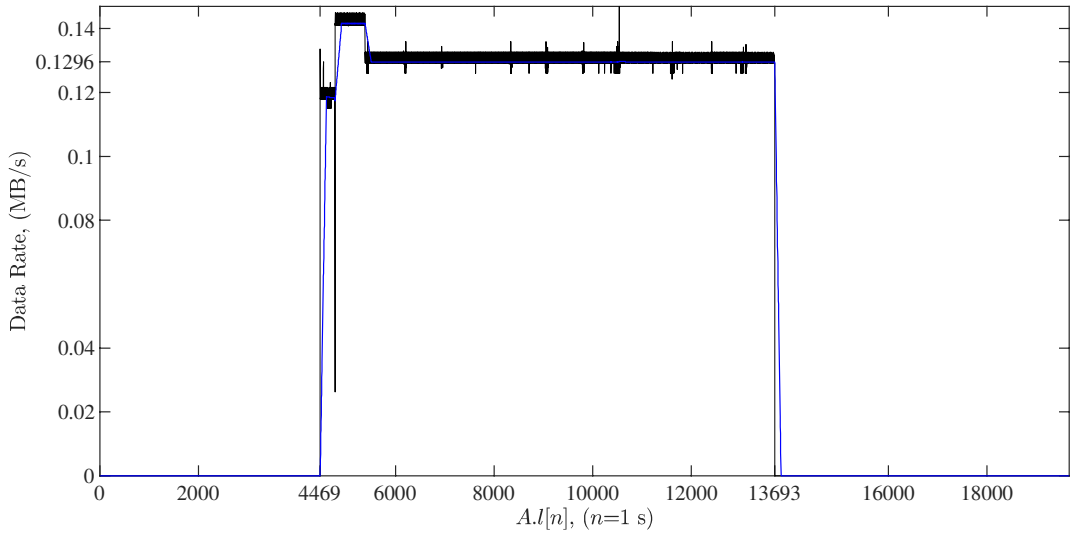


Fig. 6.4. DDoS attack data rate in 1 s time intervals. The DDoS attack data rate averages are shown as a blue coloured waveform.

6.5 DDoS Attack Flows Packet Count Integration

The attack flows $A_i.p[n]$ that each of the six bots is contributing towards the DDoS attack are isolated by filtering very specifically the source and destination IPs. The 6 attack flows are integrated over time intervals of 100 ms. The six figures 6.5-10 show the waveform of these distinct integrations correspondingly. Once the bot becomes steady it is observed that each attack flow is contributing 20 packets each 100 ms for the overall attack. Zooming into the contribution of single bots towards the overall DDoS attack is relevant because it confirms that the agents, part of a botnet, have very similar behaviour. It is important to highlight that not all botnets have the same behaviour and not all of them are expected to have a fingerprint that is easily identifiable. Concluding with the observations from Figs 6.5-10, it is seen that each of the six

bots contributes 20 packets to the overall DDoS attack depicted in Fig 6.1.

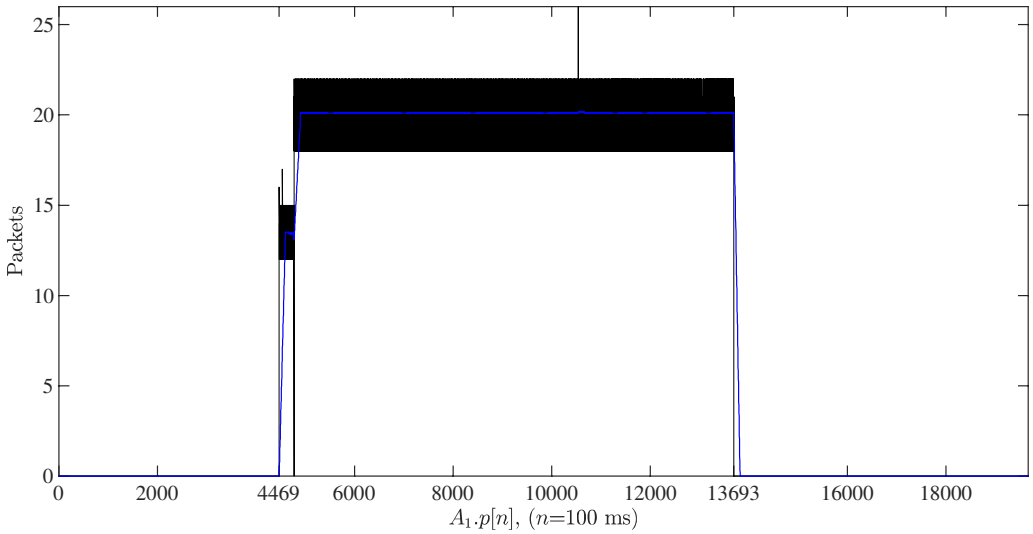


Fig. 6.5. DDoS attack flow packets counts $A_1.p[n]$ in 100 ms intervals. The DDoS attack flow packets counts averages are shown as a blue coloured waveform.

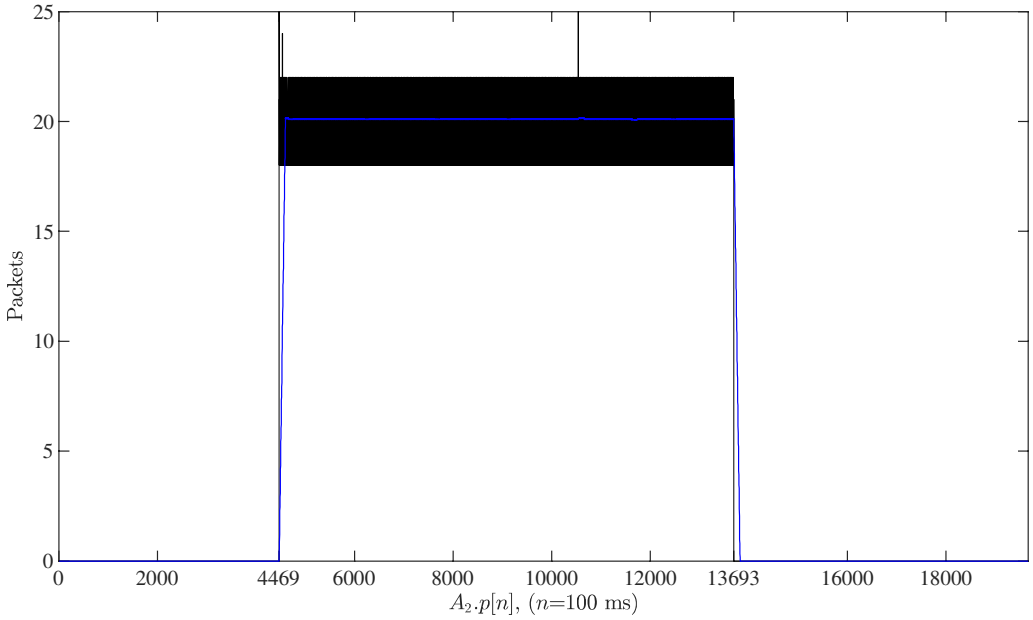


Fig. 6.6. DDoS attack flow packets counts $A_2.p[n]$ in 100 ms intervals. The DDoS attack flow packets counts averages are shown as a blue coloured waveform.

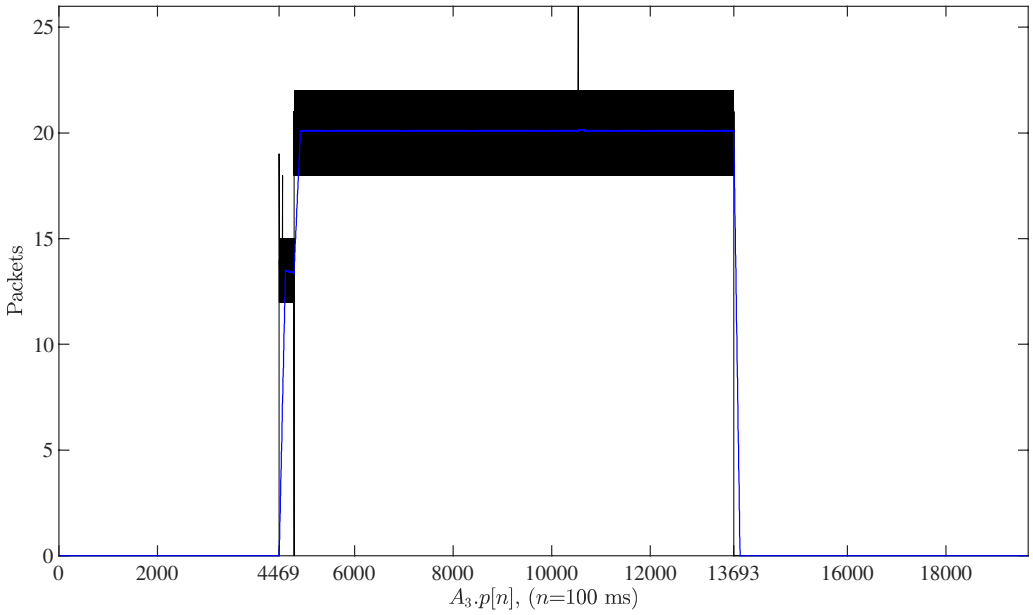


Fig. 6.7. DDoS attack flow packets counts $A_3.p[n]$ in 100 ms intervals. The DDoS attack flow packets counts averages are shown as a blue coloured waveform.

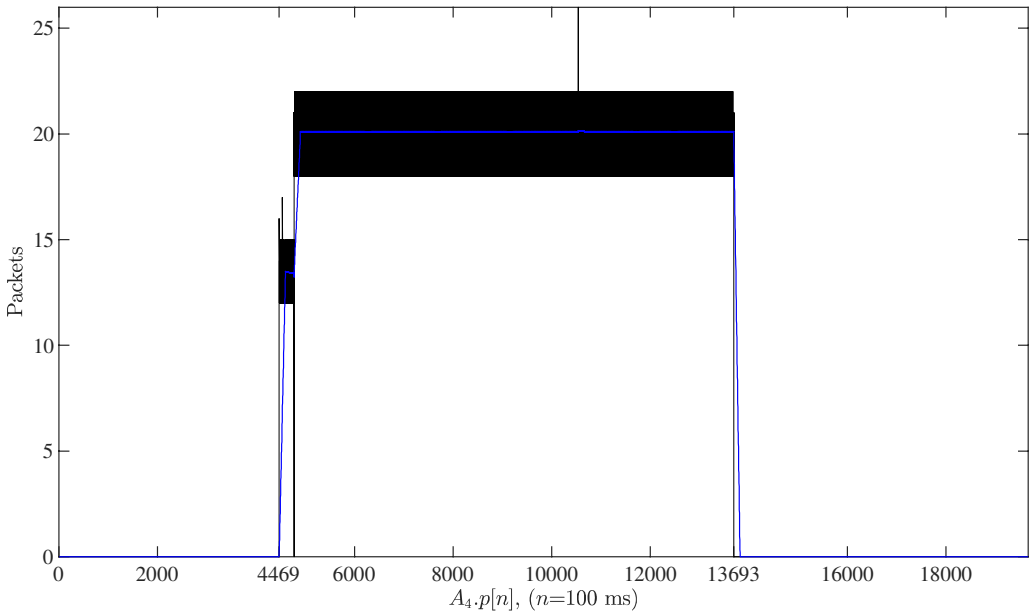


Fig. 6.8. DDoS attack flow packets counts $A_4.p[n]$ in 100 ms intervals. The DDoS attack flow packets counts averages are shown as a blue coloured waveform.

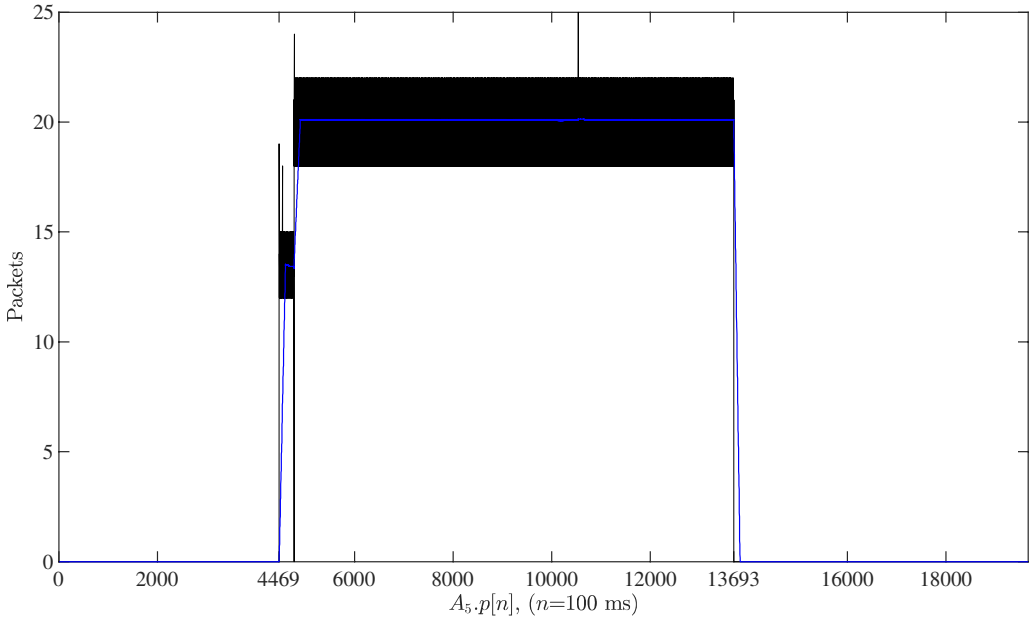


Fig. 6.9. DDoS attack flow packets counts $A_5.p[n]$ in 100 ms intervals. The DDoS attack flow packets counts averages are shown as a blue coloured waveform.

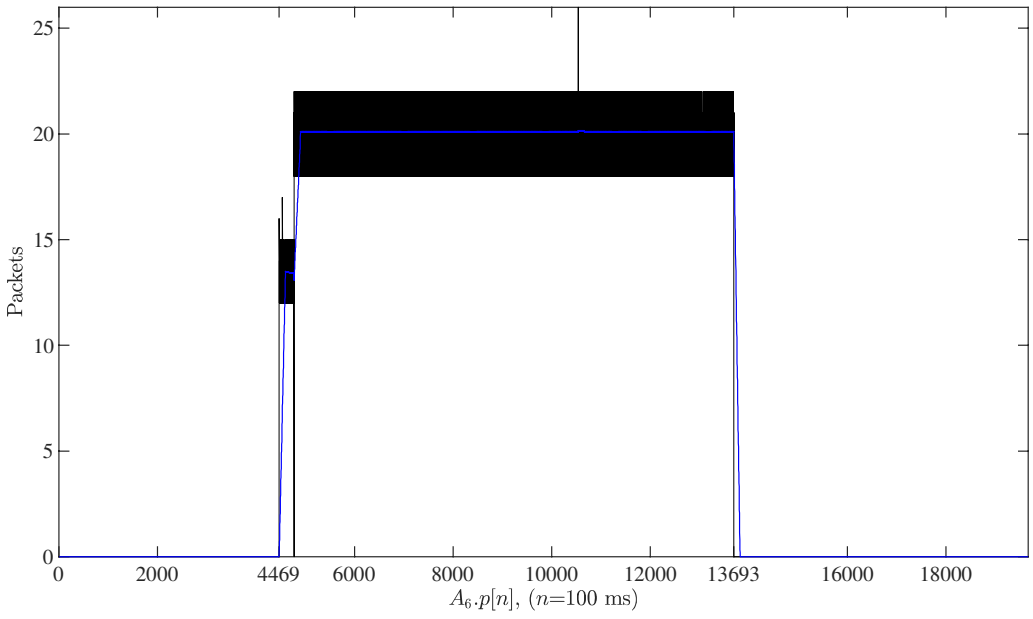


Fig. 6.10. DDoS attack flow packets counts $A_6.p[n]$ in 100 ms intervals. The DDoS attack flow packets counts averages are shown as a blue coloured waveform.

6.6 DDoS Attack Flows Length Integration

The six bots contributions to the composite DDoS attack, in terms of the packets lengths behaviours $A_i.l[n]$, ae observed in Figs. 6.11-16. The packets lengths in the six attack flows are integrated over a time interval of one second. It is observed that in all six cases the contributions of the bots, in terms of data rates, also become steady shortly after the bots have been instructed to launch the attack. This agrees with the previous results in terms of the packets count per attack flow. All figures 6.11-16 show that each bot contributes 0.0216 MB/s when attacking the victim. The accumulation of these steady contributions adds up to the data rate of 0.1296 MB/s belonging the composite DDoS attack and shown previously in Fig. 6.4. A bot contributing a negligible data rate for attacking a victim seems inconsequential, but one has to consider that the number of agents in a botnet is in the range of thousands or hundreds of thousands. It is these very high data rates which are extremely dangerous for businesses and dedicated services that our society relies on.

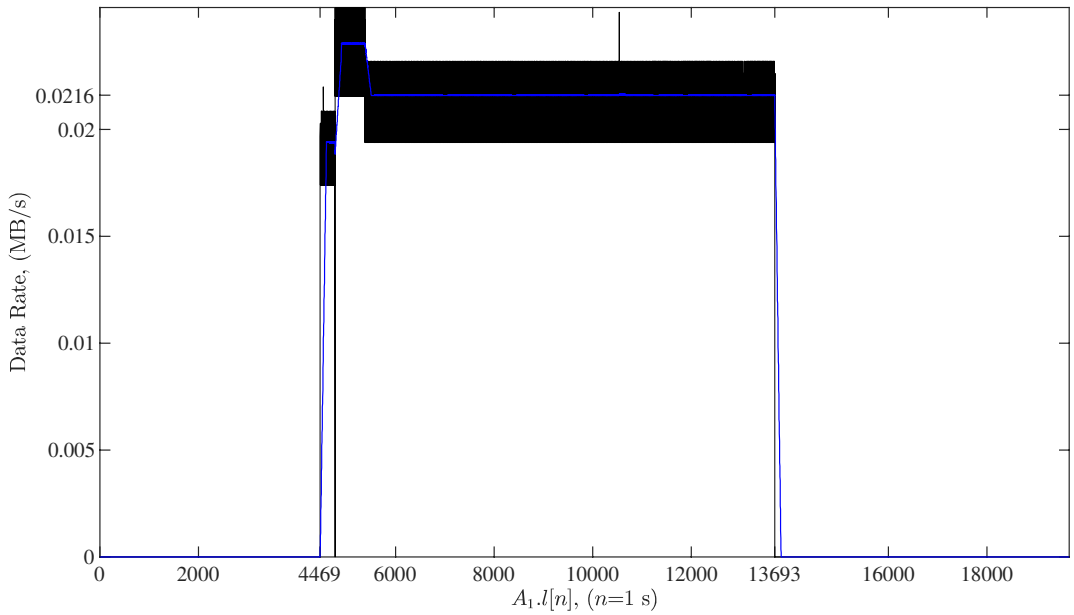


Fig. 6.11. DDoS attack flow data rate $A_1.l[n]$ in 1 s time intervals. The DDoS attack flow data rate averages are blue coloured.

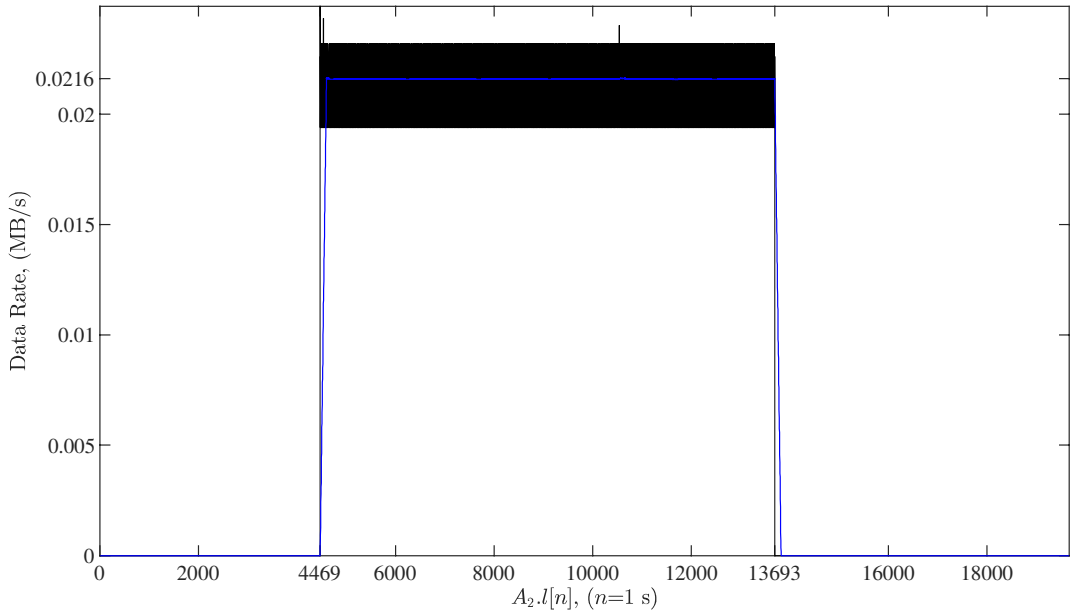


Fig. 6.12. DDoS attack flow data rate $A_2.l[n]$ in 1 s time intervals. The DDoS attack flow data rate averages are shown as a blue coloured waveform.

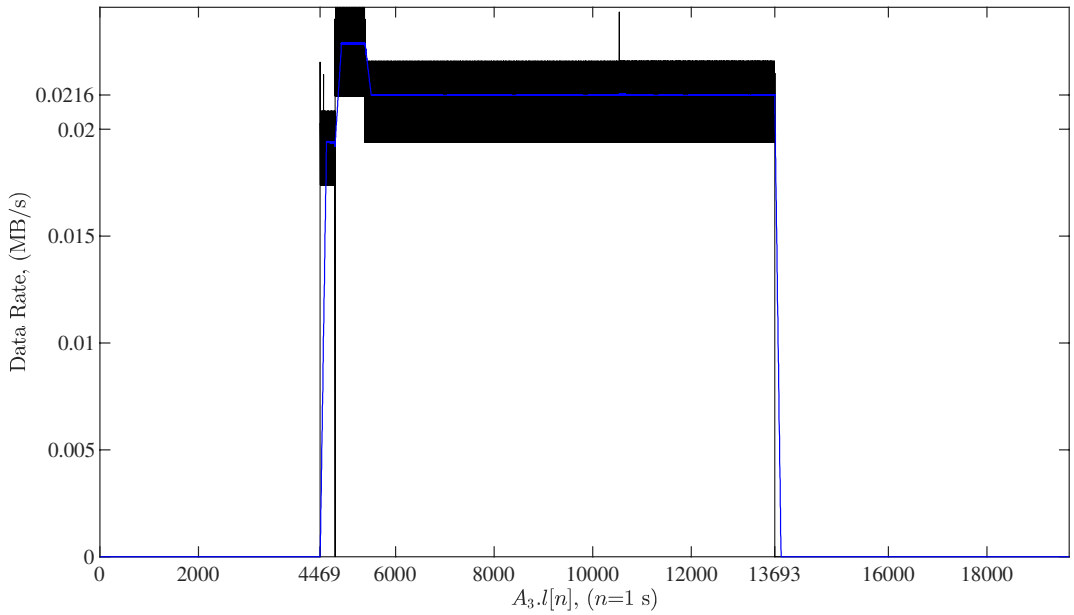


Fig. 6.13. DDoS attack flow data rate $A_3.l[n]$ in 1 s time intervals. The DDoS attack flow data rate averages are shown as a blue coloured waveform.

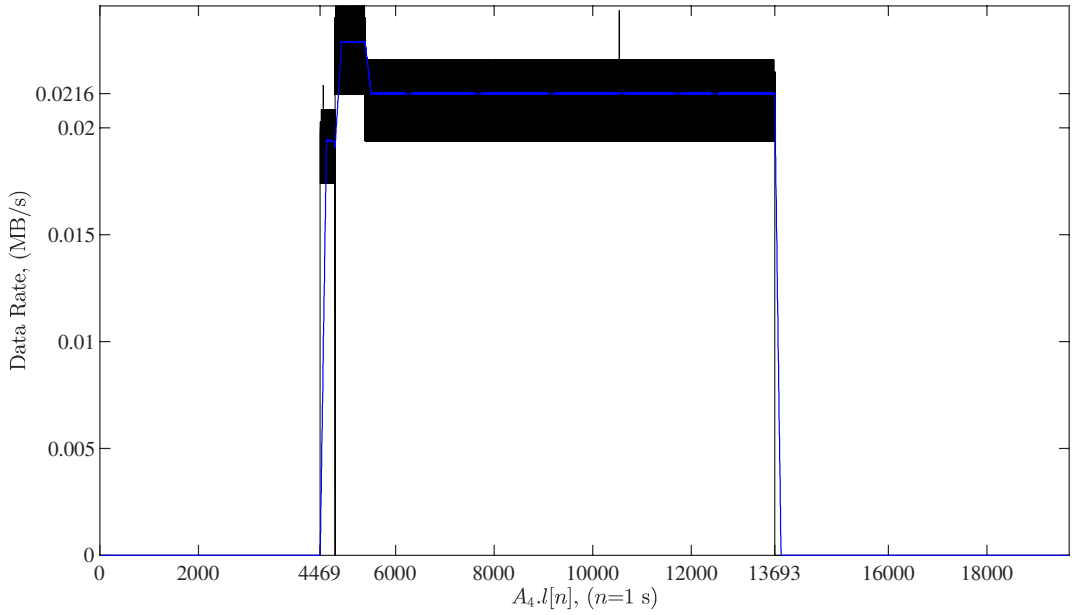


Fig. 6.14. DDoS attack flow data rate $A_4.l[n]$ in 1 s time intervals. The DDoS attack flow data rate averages are shown as a blue coloured waveform.

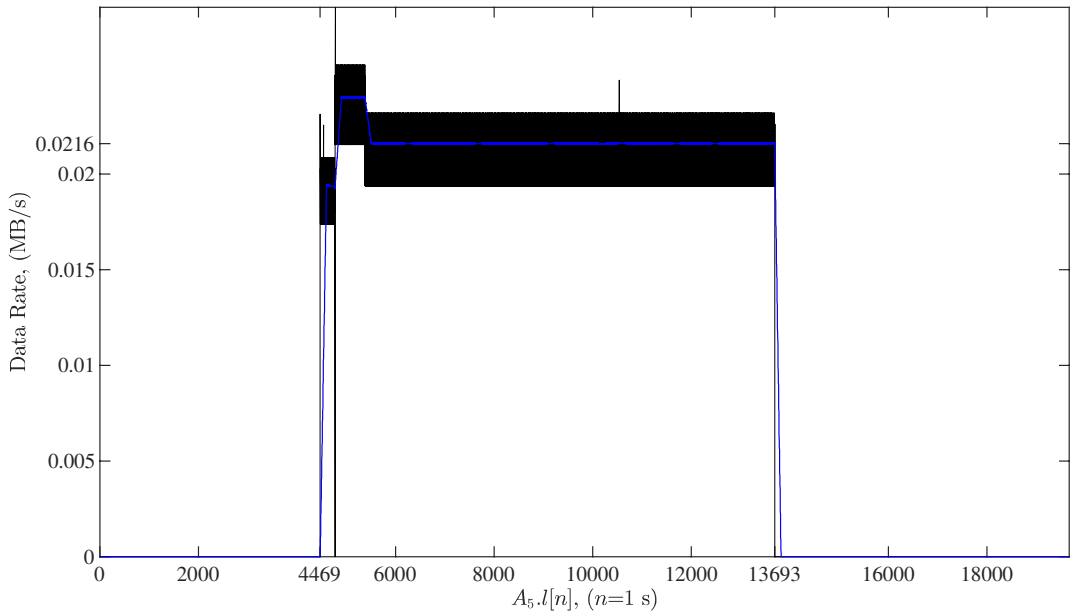


Fig. 6.15. DDoS attack flow data rate $A_5.l[n]$ in 1 s time intervals. The DDoS attack flow data rate averages are shown as a light coloured waveform.

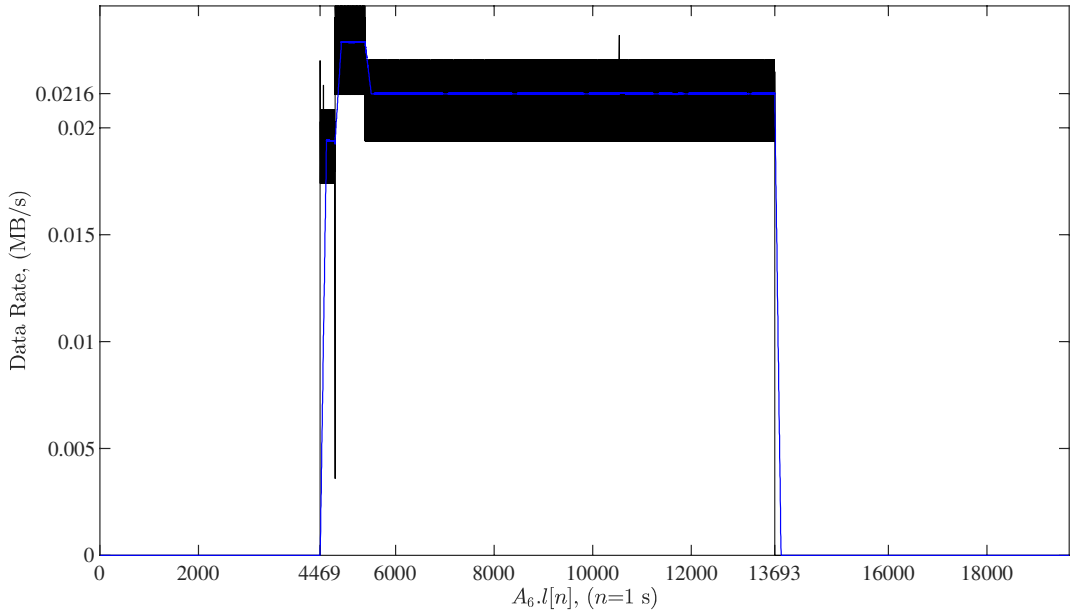


Fig. 6.16. DDoS attack flow data rate $A_6.l[n]$ in 1 s time intervals. The DDoS attack flow data rate averages are shown as a light coloured waveform.

6.7 VFD Validation

6.7.1 VFD Validation through White Noise with Uniform Distribution

Given the significance of the variance fractal dimension in this research, its algorithm implementation is verified utilizing Uniform white noise (UWN) in this experiment. Figure 6.17 shows the first 500 samples of the signal with the characteristics of white noise with Uniform distribution. This figure is included here for completeness and to provide a graphical description of the nature of this signal. Subjecting the UWN to the VFD analysis so that its fractal dimension is obtained determines that it has a value of two as shown in Fig. 6.18. The value of the log base two corresponds to the variance of 1/12, which is characteristic of the Uniform distribution. Such value is found to be constant in all the different scales for which it was calculated. Hence, the implementation of the VFD to this point is verified to be correct.

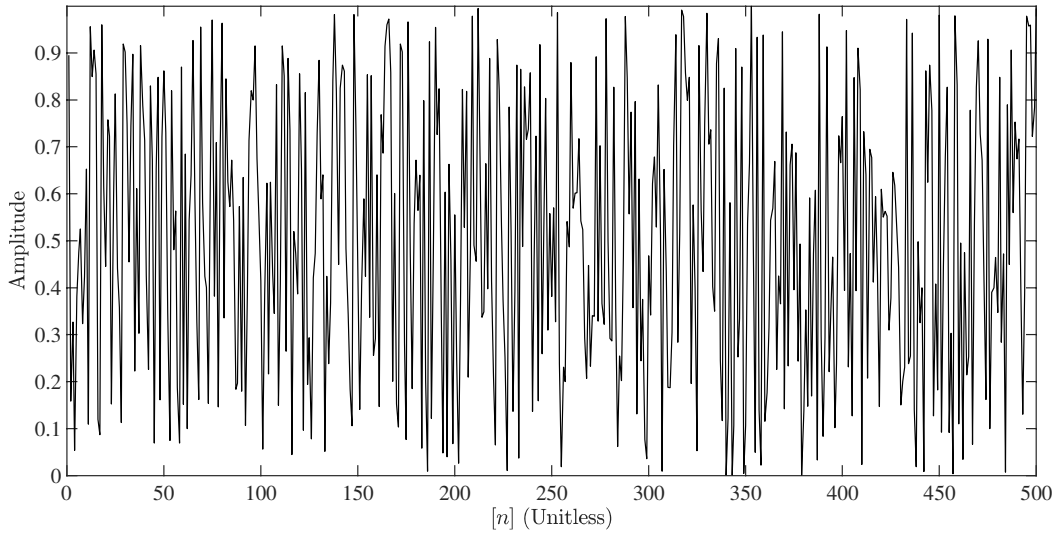


Fig. 6.17. White noise with Uniform distribution.

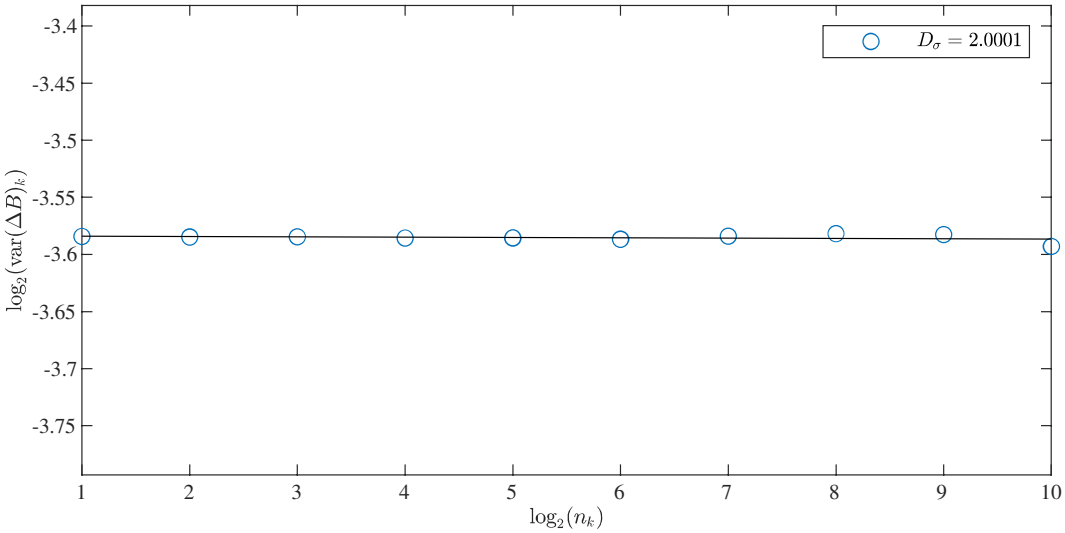


Fig. 6.18. Variance fractal dimension applied to a sequence (10 million samples long) of white noise with Gaussian distribution. The ten most significant variance values in the log-log plot are shown.

6.7.2 VFD Validation through White Noise with Gaussian Distribution

Including another known space-filling curve, as is the case of GWN, this experiment extends the validation of the VFD. Figure 6.19 shows the first 500 samples of the signal with the

characteristics of white noise with Gaussian distribution. Similarly to the previous experiment, it is found that its fractal dimension has a value of two as shown in Fig. 6.20. The value of the log base two corresponds to the variance of one (derived from the characteristic squared standard deviation equals to one in a normal Gaussian), which is characteristic of the Gaussian distribution. Such value of zero is also found to be constant in all the different scales for which it was calculated as shown in Fig. 6.20. This fact restates that the implementation of the VFD is correct.

6.8 Results of Selected Primary Analysis Operators Applied through Multiscalors

The results obtained by subjecting the variance and skewness, as primary analysis operators, to multiscalors can be found in Appendix G. No higher order moments were utilized due to increase of computing error. It is important to highlight that the results obtained by the variance multiscalor visually resembles in all its components (from first ($m_{2|||^1}$) to seventh ($m_{2|||^7}$)) both of the DDoS attacks, the DNS amplification and the H&R, that are present in the dataset and can be seen in Figs. G.1 to G.7 respectively located in Appendix G.

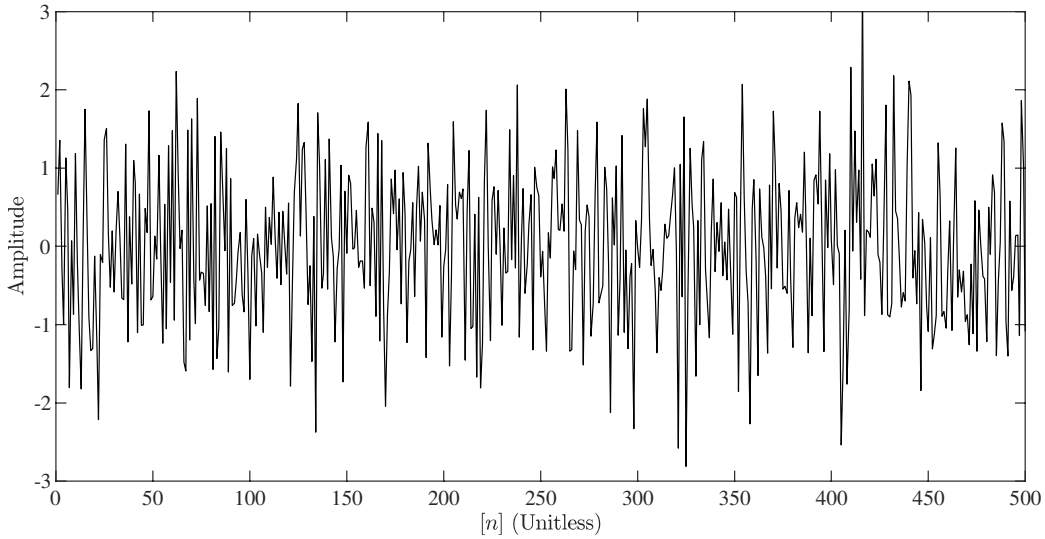


Fig. 6.19. White noise with Gaussian distribution.

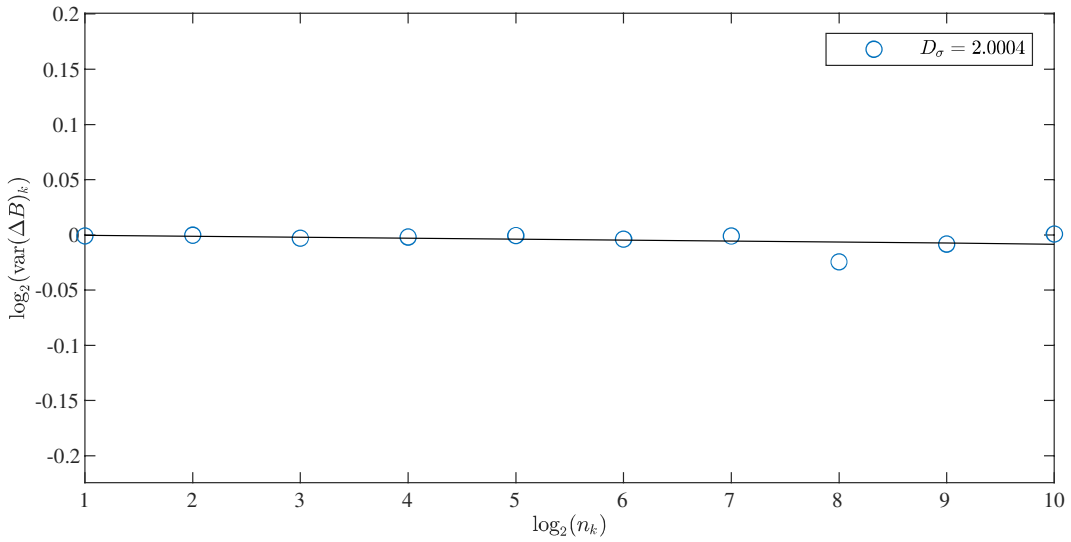


Fig. 6.20. Variance fractal dimension applied to a sequence (10 million samples long) of white noise with Gaussian distribution. The ten most significant variance values in the log-log plot are shown.

It is found that the skewness multiscalor visual insights are not as clear as the ones provided by the variance multiscalor previously. Specifically, the DNS Amplification DDoS attack is visibly identifiable in the first ($m_{3\mathrm{III}^1}$), second ($m_{3\mathrm{III}^2}$), and third ($m_{3\mathrm{III}^3}$) skewness multiscalor components while the H&R DDoS attack is remarkably weak in all components of the skewness multiscalor. Figures G.8 to G.14 found in Appendix G compile the results pertaining the skewness multiscalor.

It is expected that upon application of secondary operators to both variance and skewness multiscalors components, both DDoS attacks would become visibly stronger. This enhancement is expected prior to preparing the feature vectors intended for the machine learning models.

Finding no visual resemblance hinting at the presence of DDoS attacks (as is the case for the skewness multiscalor components) is no limitation for recognition of patterns through machine learning, which are computing implementations capable of providing quantifiable conclusions. This is one of the reasons for subjecting the individual components of both variance and skewness multiscalors to a successive analysis stage. This *secondary analysis stage* is designed for fitting secondary operators as cumulative sum, ZCR, and Shannon's entropy. This secondary analysis has a twofold purpose, *collecting* more robust and diverse insights into the

dynamics of the Internet/network traffic and achieving a *higher data compression* for the feature vector used in the classification stage through the ART variants implemented.

The author is acquainted with distinct areas of signal analysis in domains such as time, time-frequency (*e.g.*, short-time Fourier transform, short-time cepstrum), multiscale (*e.g.*, wavelets and fractal analysis), and polyscale. It is the latter that is the focus in this research. In order to achieve real-time processing, it is best to have computing methodologies and techniques with a low impact, which in an aggregated form are capable of extracting robust features. Hence, the reason why the three distinct methodologies, cumulative sum, ZCR, and Shannon entropy, have been selected for further analysis of the components of both the variance and skewness multiscalors.

For each secondary operator applied to the multiscalors components (one, Ξ_{III^1}, to seven, Ξ_{III^7} in this case) in the next subsection, four figures (6.21, 6.23-25) show the results of the secondary operator applied to a given multiscalor component, a denoising stage based on Donoho's method, a median filtering stage, and quantization stage. The results yielded by secondary operators lacking a significant or visible resemblance of the DDoS attack have been removed for brevity. Nonetheless, experiments with the machine learning algorithms selected with a feature vector incorporating all the results of the secondary operators applied to the multiscalors components have been conducted and this is highlighted accordingly. All plots are captioned properly to identify the secondary operator in question (cumulative sum, ZCR, or Shannon entropy), the multiscalor component (from first, Ξ_{III^1}, to seventh, Ξ_{III^7}), and the analysis stage (outcome of the secondary operator, denoising, non-linear filtering, and quantization). A concrete description and discussion about the dynamics seen within the analysis stage done for each component of both the variance and skewness multiscalors is elaborated on with the corresponding figure provided.

6.9 Availability of Signals for Adaptive Resonance Theory

This subsection presents the DDoS dataset (a signal $S\!I[n]$ with a specific integration time of $n = 1.0486\ \text{s}$) pipelined through Donoho's denoising, median non-linear filtering, and Lloyd's quantization. These three methodologies have been already described extensively in this

thesis. The DDoS dataset used in this pipelining is depicted in Fig. 6.21.

In figure 6.21 a traffic burst is observed between $n = 1,600$ and $n = 1,800$. This traffic burst has all the characteristics of a hit and run DDoS, which is a special form of DDoS that is activated and deactivated periodically. In Figure 6.22, the details of this burst are presented.

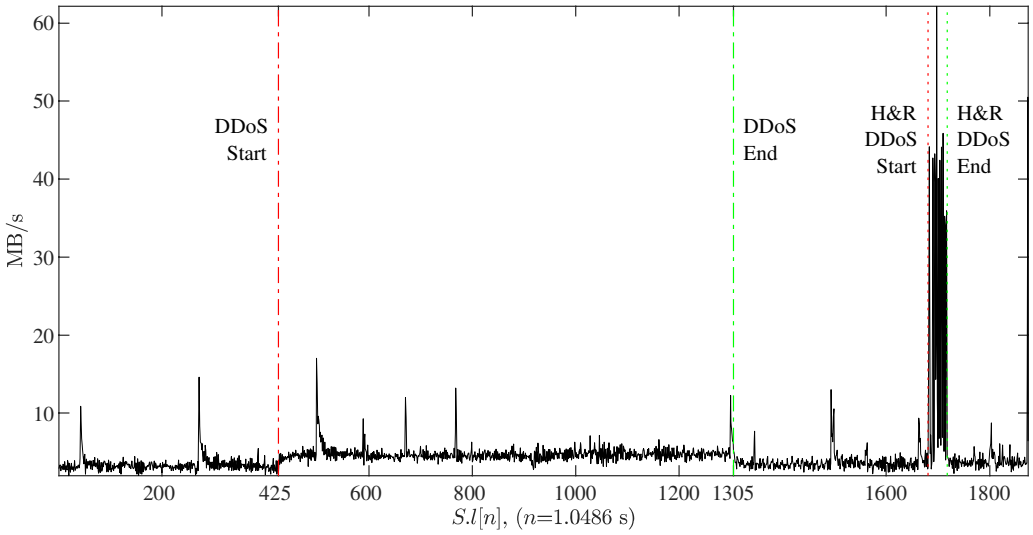


Fig. 6.21. Traffic data rate with an integration time of 1.0486 s. The DDoS attack start (marked with a red dashed and dotted line) and end (marked with a green dashed and dotted line) are seen at $n = 425$ and $n = 1,305$ respectively. Also, a hit and run DDoS attack is seen between $n = 1,681$ and $n = 1,718$.

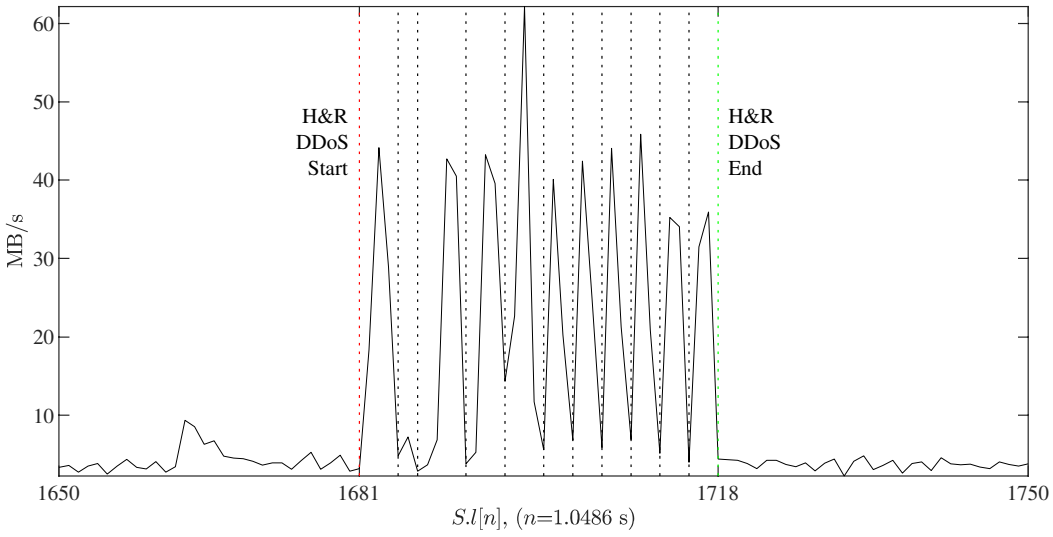


Fig. 6.22. Hit and run DDoS attack start (marked with a red dotted line) and end (marked with a green dotted line) are seen at $n = 1,681$ and $n = 1,718$ respectively. Eleven peaks are seen during the duration of this attack.

6.9.1 Denoising

The DDoS dataset is firstly exposed to the Donoho's denoising methodology for which a Coiflet wavelet with scaling factor of five has been used. This specific wavelet has been selected because Coiflets resemble best the shape of Internet traffic. Figure 6.23 shows the results of processing the DDoS dataset with Donoho's denoising, which achieves a smother waveform.

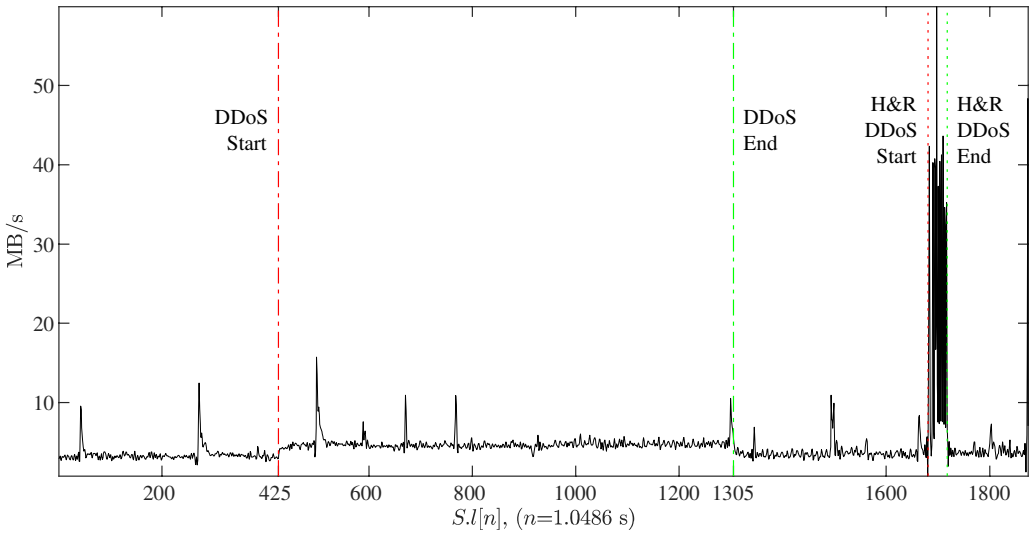


Fig. 6.23. DDoS dataset processed with Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack start (marked with a red dashed and dotted line) and end (marked with a green dashed and dotted line) are seen at $n = 425$ and $n = 1{,}305$ respectively. Also, a hit and run DDoS attack is seen between $n = 1{,}681$ and $n = 1{,}718$.

6.9.2 Non-Linear Filtering

In Figure 6.23 one still observes small peaks throughout the waveform, which are possible to remove with a non-linear technique, which in this case median filtering has been chosen. Processing the smoothed waveform with median filtering as seen in Fig. 6.24 smooths the waveform even further. The result of the non-linear median filtering is seen to cause an amplification (having a bigger impact) in the section where the hit and run DDoS attack is found.

6.9.3 Quantization

Figure 6.25 shows the DDoS dataset quantized with Lloyd's methodology. Proving that a

complex waveform, as is the case of Internet traffic, can be quantized with Lloyd's methodology provides assurance that the distinct features proposed in this thesis can be processed by the ART1 neural network. One has to recall that the ART1 can only process binary input vectors while FuzzyART can process vectors containing continuous values.

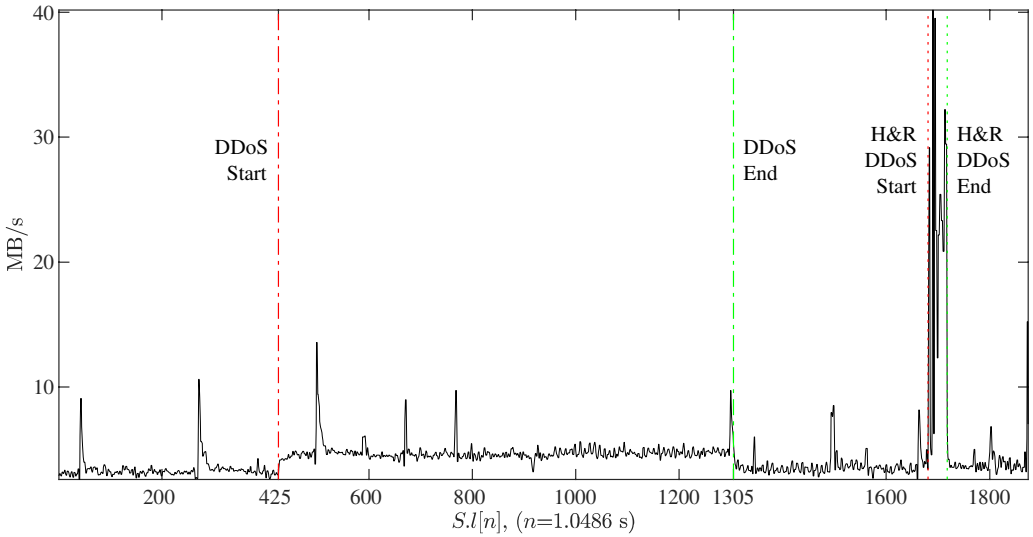


Fig. 6.24. DDoS dataset processed with median filtering once denoised with Donoho's methodology. The DDoS attack start (marked with a red dashed and dotted line) and end (marked with a green dashed and dotted line) are seen at $n = 425$ and $n = 1,305$ respectively. Also, a hit and run DDoS attack is seen between $n = 1,681$ and $n = 1,718$.

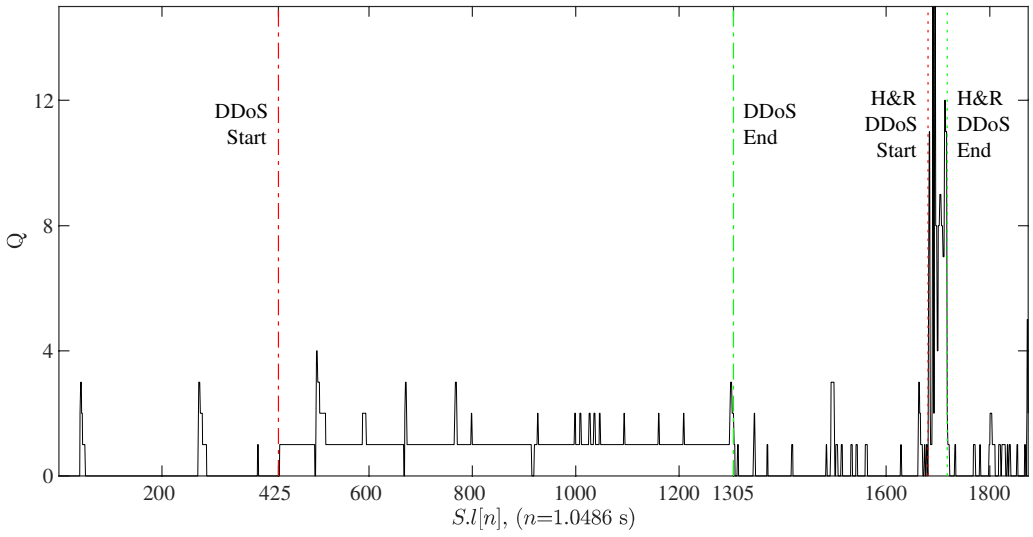


Fig. 6.25. Quantized DDoS dataset with Lloyd's methodology. The DDoS attack start (marked with a red dashed and dotted line) and end (marked with a green dashed and dotted line) are seen at $n = 425$ and $n = 1,305$ respectively. Also, a hit and run DDoS attack is seen between $n = 1,681$ and $n = 1,718$.

6.10 Findings About the Quality Detection of Variance Multiscalor Features

Table I condenses results for the quality detection of the secondary operators applied towards the variance multiscalor. The description for each case is fully provided and documented in Appendix H, where 72 corresponding plots and their thorough descriptions are found.

TABLE I
QUALITY DETECTION OF VARIANCE MULTISCALOR

Multiscalor	Secondary Operator	Component	Detection Quality					
			DNS DDoS			H&R DDoS		
			High	Medium	Low	High	Medium	Low
Variance $m_{2\|\|^n}$	Cumulative Sum (S)	$m_{2\|\|^1}$	•			•		
		$m_{2\|\|^2}$	•			•		
		$m_{2\|\|^3}$	•			•		
		$m_{2\|\|^4}$	•			•		
		$m_{2\|\|^5}$	•			•		
		$m_{2\|\|^6}$	•			•		
		$m_{2\|\|^7}$	•			•		
	Zero Crossing Rate (Z_n)	$m_{2\|\|^1}$		•		•		
		$m_{2\|\|^2}$	•			•		
		$m_{2\|\|^3}$	•			•		
		$m_{2\|\|^4}$	•			•		
		$m_{2\|\|^5}$	•			•		
		$m_{2\|\|^6}$	•			•		
		$m_{2\|\|^7}$	•			•		
	Shannon's Entropy (H)	$m_{2\|\|^1}$			•			•
		$m_{2\|\|^2}$			•			•
		$m_{2\|\|^3}$			•			•
		$m_{2\|\|^4}$		•				•
		$m_{2\|\|^5}$		•			•	
		$m_{2\|\|^6}$		•		•		
		$m_{2\|\|^7}$			•	•		

6.11 Findings About the Quality Detection of Skewness Multiscalor Features

Results for the quality detection of the secondary operators applied onto the skewness multiscalor is condensed in Table II. The description for each case is fully provided and

documented in Appendix I, where corresponding plots and their thorough descriptions are found.

TABLE II
QUALITY DETECTION OF SKEWNESS MULTISCALOR

Multiscalor	Secondary Operator	Component	Detection Quality					
			DNS DDoS			H&R DDoS		
			High	Medium	Low	High	Medium	Low
Skewness m_{3III^n}	Cumulative Sum (S)	m_{3II^1}	•					•
		m_{3II^2}		•				•
		m_{3II^3}		•				•
		m_{3II^4}	•					•
		m_{3II^5}		•				•
		m_{3II^6}			•	•		
		m_{3II^7}			•	•		
	Zero Crossing Rate (Z_n)	m_{3II^1}	•			•		
		m_{3II^2}			•		•	
		m_{3II^3}			•		•	
		m_{3II^4}		•			•	
		m_{3II^5}		•			•	
		m_{3II^6}			•	•		
		m_{3II^7}			•	•		
	Shannon's Entropy (H)	m_{3II^1}	•			•		
		m_{3II^2}			•		•	
		m_{3II^3}			•			•
		m_{3II^4}			•	•		
		m_{3II^5}			•	•		
		m_{3II^6}			•	•		
		m_{3II^7}			•	•		

6.12 Preparation of Feature Vector for ART1

The different stages performing the multiscale analysis of the dataset containing DDoS attacks have been presented. This particular segment reveals details about how the feature vector needs to be shaped for further processing by ART1 in order to obtain classification outcomes based on the relevant descriptors.

Once achieving the compression of the raw signal through multiscale and polyscale analysis, the features extracted from the multiscalors components are assembled into a feature

vector. Such vector would then be subject for classification through computational intelligence utilizing ART1. In order to have this feature vector ready for ART1, each feature is exposed to denoising, non-linear filtering, and quantization.

The feature vector for ART1 bundles 42 quantized scalars (each in a four bits long binary representation) entirely, which are identified in detail as: Seven scalars from the cumulative sum applied to the variance multiscalor, seven scalars from the ZCR applied to the variance multiscalor, seven scalars from entropy applied to the variance multiscalor, seven scalars from the cumulative sum applied to the skewness multiscalor, seven scalars from the ZCR applied to the skewness multiscalor, and seven scalars from entropy applied to the skewness multiscalor.

6.12.1 Features Stemming from Cumulative Sum Applied to Variance Multiscalor

Figures 6.26 and 6.27 show clear details about strong features generated through the variance multiscalor and that are further analysed with the cumulative sum. Both of the DDoS attacks are noticeable with clear beginnings and ends that are persistent in all variance multiscalor components (from the first $m_{2^{III^1}}$ to seventh $m_{2^{III^7}}$). Figure 6.26 shows a central wide band corresponding to the DNS DDoS attack starting and finishing in the processing frames 425 and 1,305 respectively, and a narrow band in the right corresponding to the H&R DDoS attack starting and finishing in the processing frames 1,681 and 1,718 respectively. There are narrow bands present, which correspond to minor spikes across the traffic. The cumulative sum visually registers in Fig. 6.26 high quality detection features when applied onto the seven variance multiscalor components as previously listed in Table I.

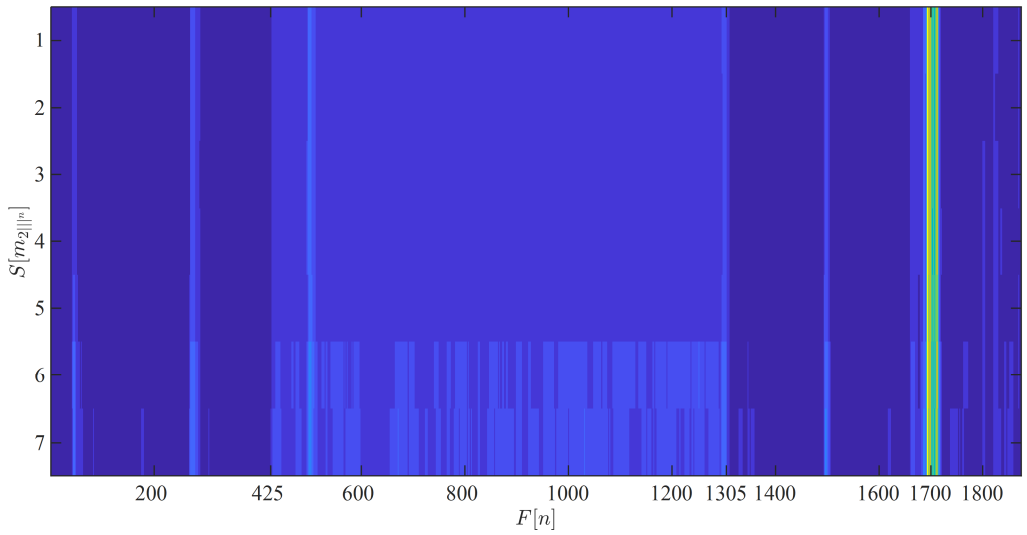


Fig. 6.26. Segment of quantized feature vector corresponding to the cumulative sum *S* applied to the variance multiscalor components ($m_{2|||^1}$ to $m_{2|||^7}$) .

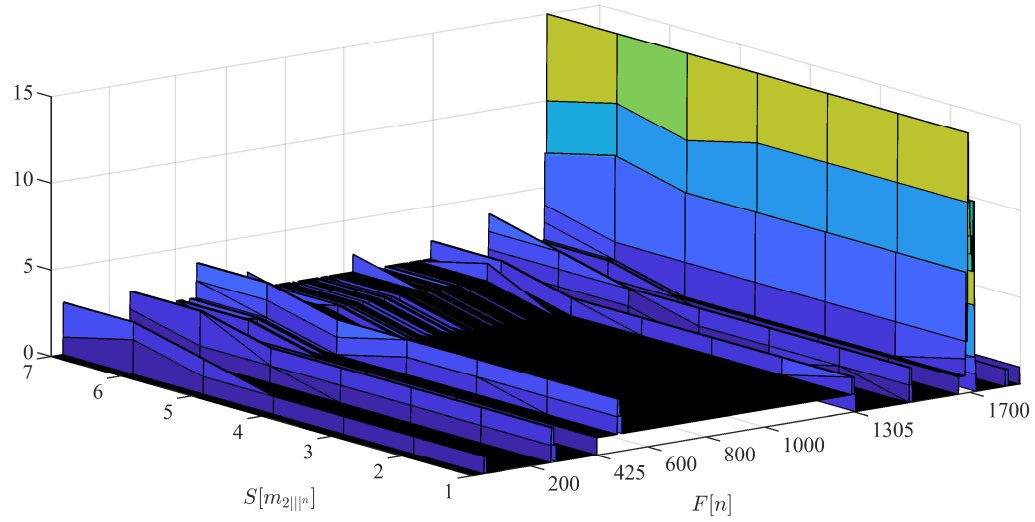


Fig. 6.27. Segment of quantized feature vector corresponding to the cumulative sum *S* applied to the variance multiscalor components ($m_{2|||^1}$ to $m_{2|||^7}$) .

6.12.2 Features Stemming from ZCR Applied to Variance Multiscalor

When the variance multiscalor components are processed through ZCR, as shown in Figs. 6.28 and 6.29, features with good detection quality are generated. The DNS DDoS attack appears well segmented from the second $m_{2\|\|^2}$ to seventh $m_{2\|\|^7}$ components, while the H&R DDoS attack appears well segmented for from the first $m_{2\|\|^1}$ to seventh $m_{2\|\|^7}$ components. Figure 6.28 shows a central wide band corresponding to the DNS DDoS attack starting and finishing in the processing frames 425 and 1,305 respectively, and a narrow band in the right corresponding to the H&R DDoS attack starting and finishing in the processing frames 1,681 and 1,718 respectively. The ZCR visually registers in Fig. 6.28 six high quality detection features for the DNS DDoS attack and seven high quality detection features for the H&R DDoS attack when applied onto the seven variance multiscalor components as listed earlier in Table I.

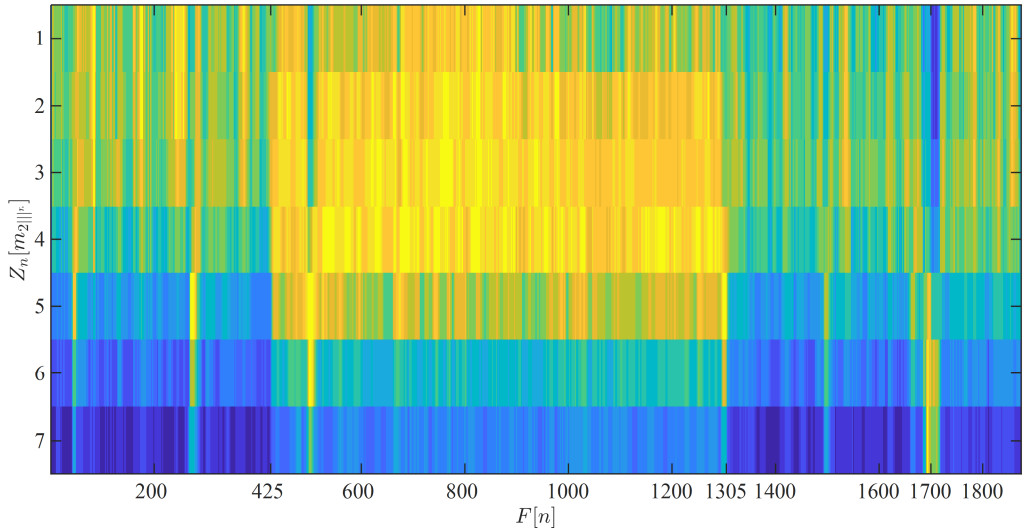


Fig. 6.28. Segment of quantized feature vector corresponding to the ZCR Z_n applied to the variance multiscalor components ($m_{2\|\|^1}$ to $m_{2\|\|^7}$).

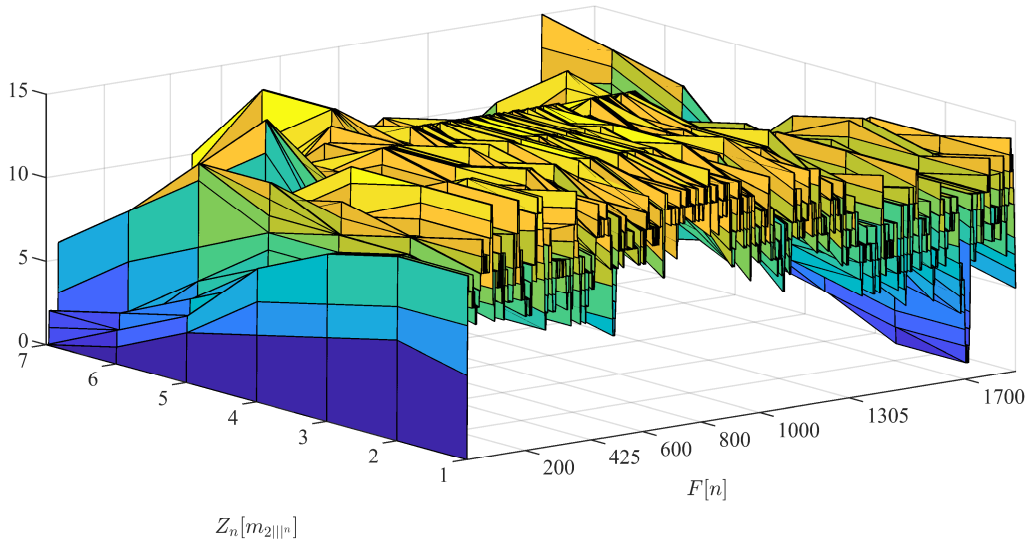


Fig. 6.29. Segment of quantized feature vector corresponding to the ZCR Z_n applied to the variance multiscalor components ($m_{2|||^1}$ to $m_{2|||^7}$).

6.12.3 Features Stemming from Shannon's Entropy Applied to Variance Multiscalor

From the secondary operators applied towards the variance multiscalor, Shannon's entropy is the one that brings fewer promising outcomes as observed in Figs. 6.30 and 6.31 where clear details of the DDoS attacks are missing. Nevertheless, the H&R DDoS attack is perceptible in the sixth $m_{2|||^6}$ and seventh $m_{2|||^7}$ component in the narrow band in the right starting and finishing in the processing frames 1,681 and 1,718, which appears prominently for the cumulative sum and ZCR cases just introduced. It shall be observed that the outcomes from the first and second multiscalor components lack insights about the traffic in general as observed in Fig. 6.30. Shannon's entropy then only provides two high quality detection features for the H&R DDoS case, which have been also pinpointed in Table I.

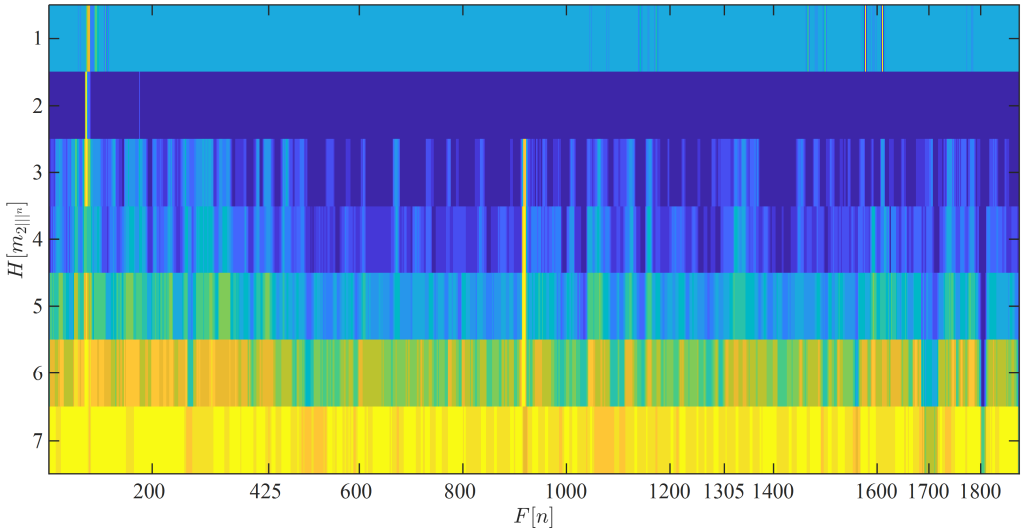


Fig. 6.30. Segment of quantized feature vector corresponding to the Shannon's entropy *H* applied to the variance multiscalor components ($m_{2|||^1}$ to $m_{2|||^7}$) .

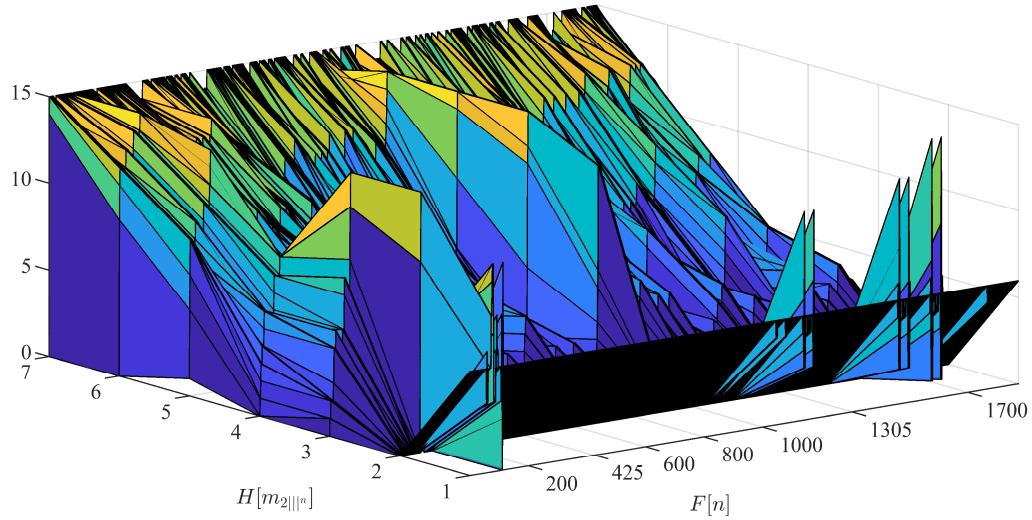


Fig. 6.31. Segment of quantized feature vector corresponding to the Shannon's entropy *H* applied to the variance multiscalor components ($m_{2|||^1}$ to $m_{2|||^7}$) .

6.12.4 Ensemble of Features Stemming from Secondary Operators Applied to Variance Multiscalor

Figure 6.32 brings into visual perspective the outcomes from the secondary operators

(cumulative sum, ZCR, and Shannon's entropy) when processing the variance multiscalor components. From the horizontal rows in Fig. 6.32, the first to the seventh describe the cumulative sum outcomes, the eight to the 14th describe the ZCR outcomes, and the 15th to the 21st describe the Shannon's entropy outcomes. A surface tridimensional plot is provided in Fig. 6.33 where the amplitude differences among the three sets of secondary operators outcomes is easier to grasp.

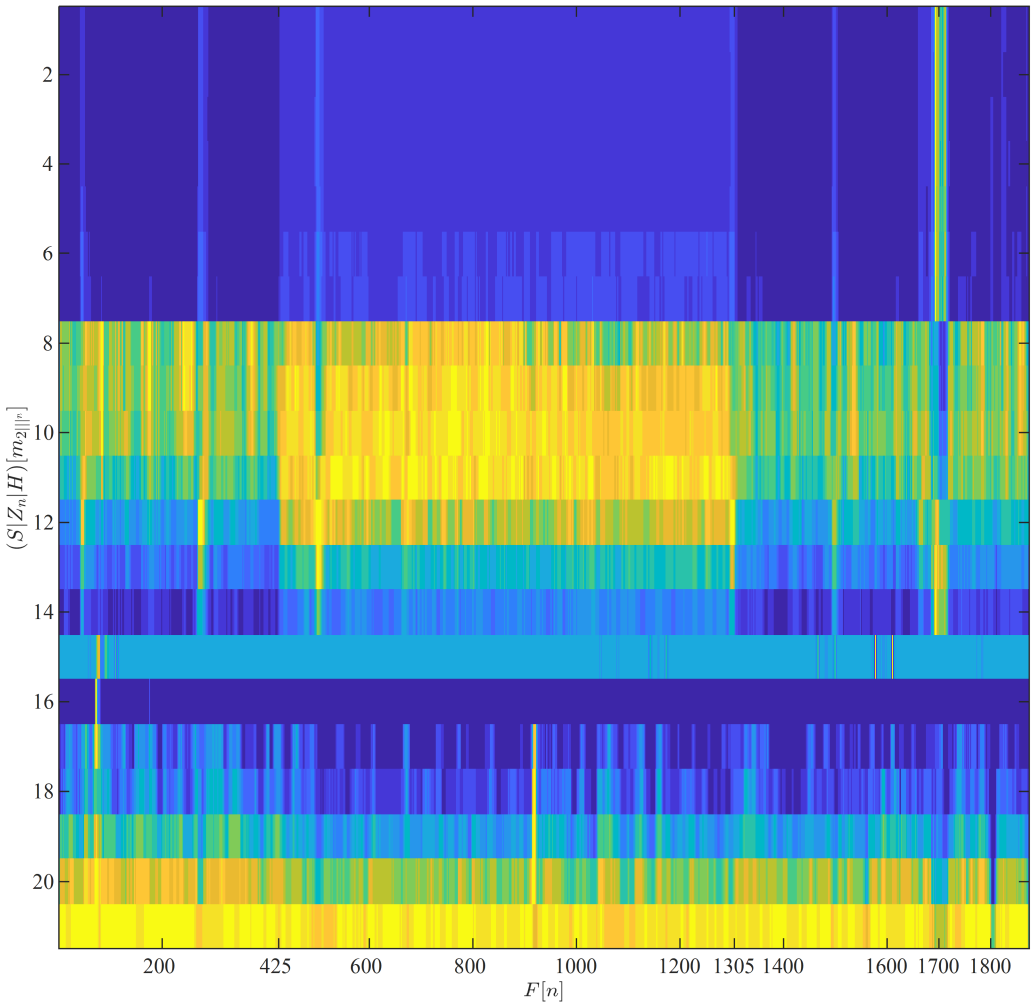


Fig. 6.32. Segment of quantized feature vector corresponding to the secondary operators applied to the variance multiscalor components ($m_{2\|\|^1}$ to $m_{2\|\|^7}$).

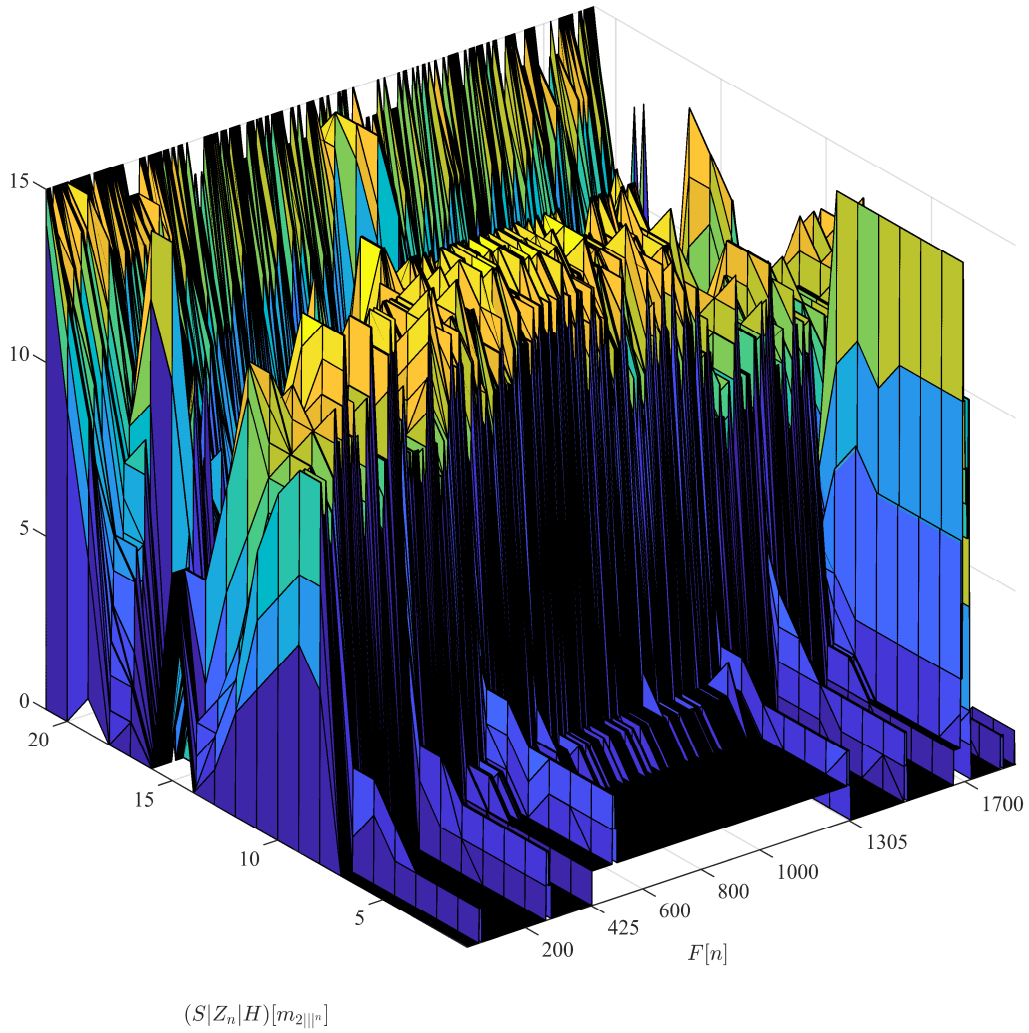


Fig. 6.33. Segment of quantized feature vector corresponding to the secondary operators applied to the variance multiscalor components ($m_{2|||^1}$ to $m_{2|||^7}$).

6.12.5 Features Stemming from Cumulative Sum Applied to Skewness Multiscalor

Features generated by the cumulative sum applied to the skewness multiscalor are shown in Figs. 6.34 and 6.35. The DNS DDoS attack is present in the first $m_{3|||^1}$ and the fourth $m_{3|||^4}$ skewness multiscalor components, while the H&R DDoS attack is present in the sixth $m_{3|||^6}$ and

the seventh $m_{3|||^7}$ skewness multiscalor components. The skewness multiscalor components observed in Fig. 6.34 are of lesser quality than the variance multiscalor components shown in Fig. 6.26, which show more defined bands containing the DDoS attacks. The four high detection quality skewness multiscalors are listed in detail in Table II.

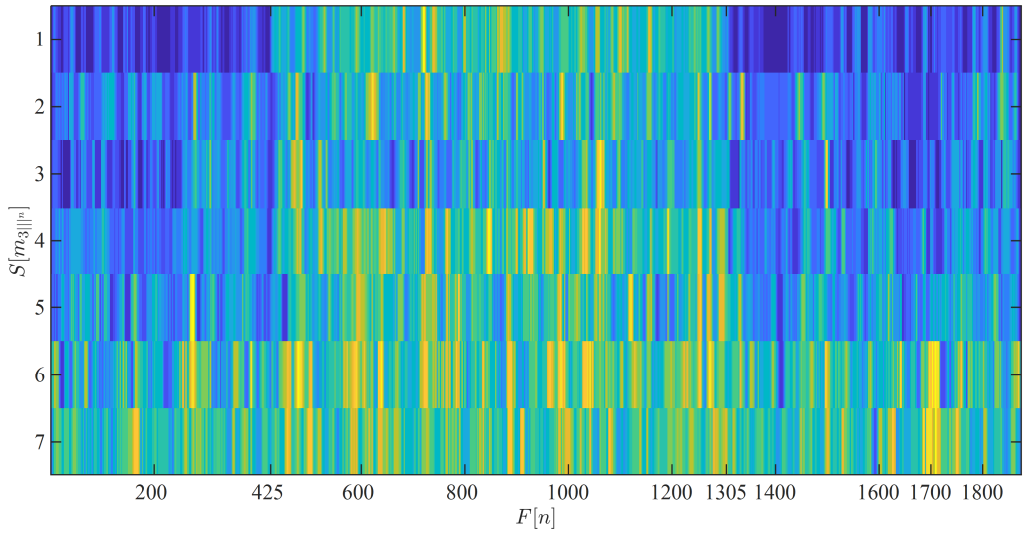


Fig. 6.34. Segment of quantized feature vector corresponding to the cumulative sum *S* applied to the skewness multiscalor components ($m_{3|||^4}$ to $m_{3|||^7}$).

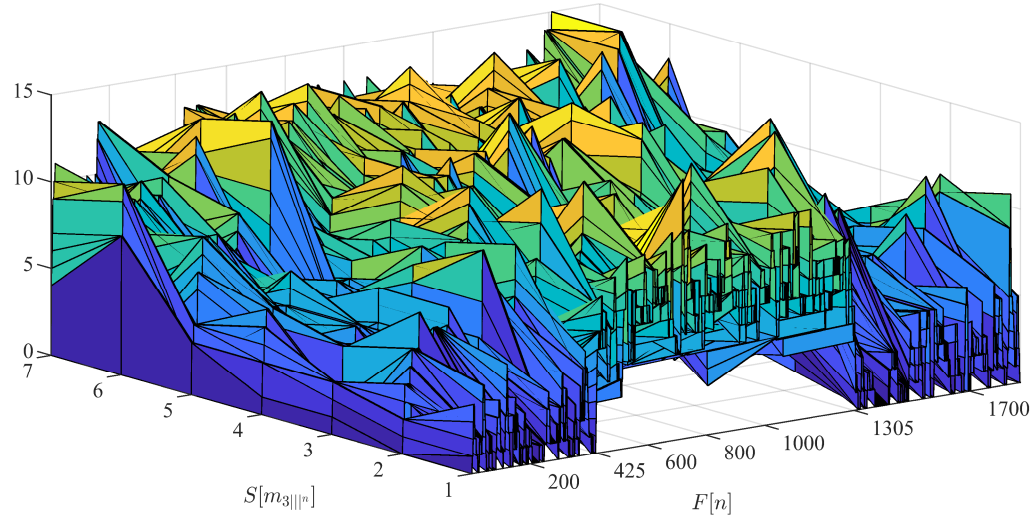


Fig. 6.35. Segment of quantized feature vector corresponding to the cumulative sum *S* applied to the skewness multiscalor components ($m_{3|||^4}$ to $m_{3|||^7}$).

6.12.6 Features Stemming from ZCR Applied to Skewness Multiscalor

Figures 6.36 and 6.37 depict the features yielded by the ZCR applied to the skewness multiscalor. The first $m_{3|||^1}$ skewness multiscalor component denotes the feature with the highest quality detection capacity for the DNS DDoS attack. The first $m_{3|||^1}$, sixth $m_{3|||^6}$, and seventh $m_{3|||^7}$ skewness multiscalor components show strong quality detection to resolve the H&R DDoS attack. Table II shows details about the detection quality for the secondary operators applied to the skewness multiscalors.

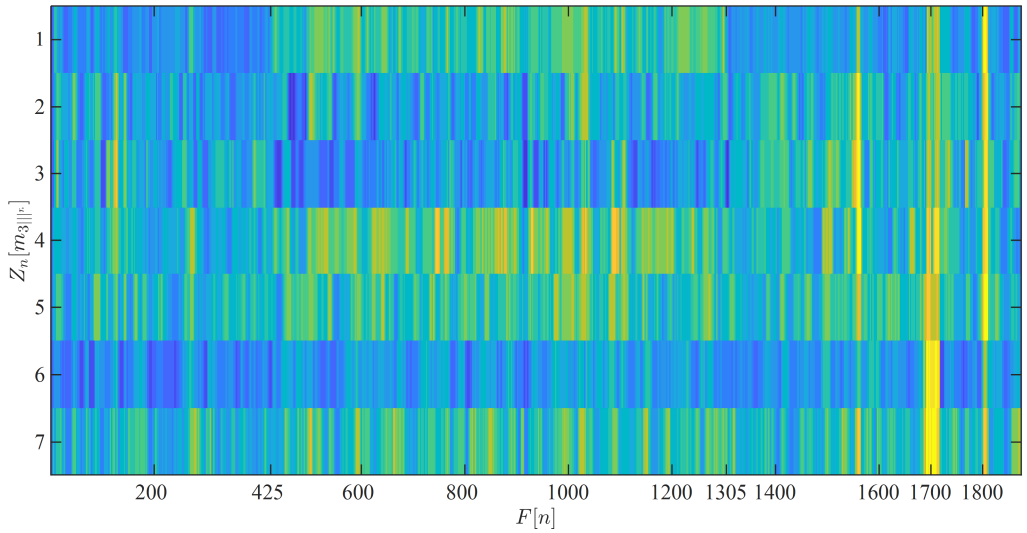


Fig. 6.36. Segment of quantized feature vector corresponding to the ZCR Z_n applied to the skewness multiscalor components ($m_{3|||^1}$ to $m_{3|||^7}$).

6.12.7 Features Stemming from Shannon's Entropy Applied to Skewness Multiscalor

Figures 6.38 and 6.39 depict the features produced by the Shannon's entropy applied to the skewness multiscalor. The DNS DDoS is detectable only by the first $m_{3|||^1}$ skewness multiscalor component, while the H&R DDoS attack is detectable by the first $m_{3|||^1}$ and from the fourth $m_{3|||^4}$ to the seventh $m_{3|||^7}$ skewness multiscalor components. The particular details about the Shannon's entropy detection quality for resolving both DDoS attacks are listed in Table II.

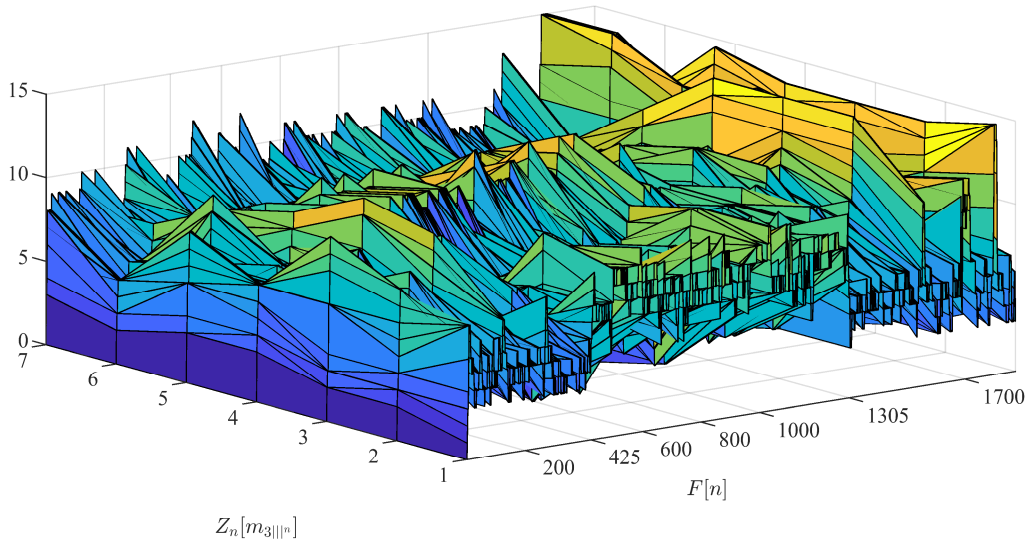


Fig. 6.37. Segment of quantized feature vector corresponding to the ZCR Z_n applied to the skewness multiscalor components ($m_{3|||^4}$ to $m_{3|||^7}$).

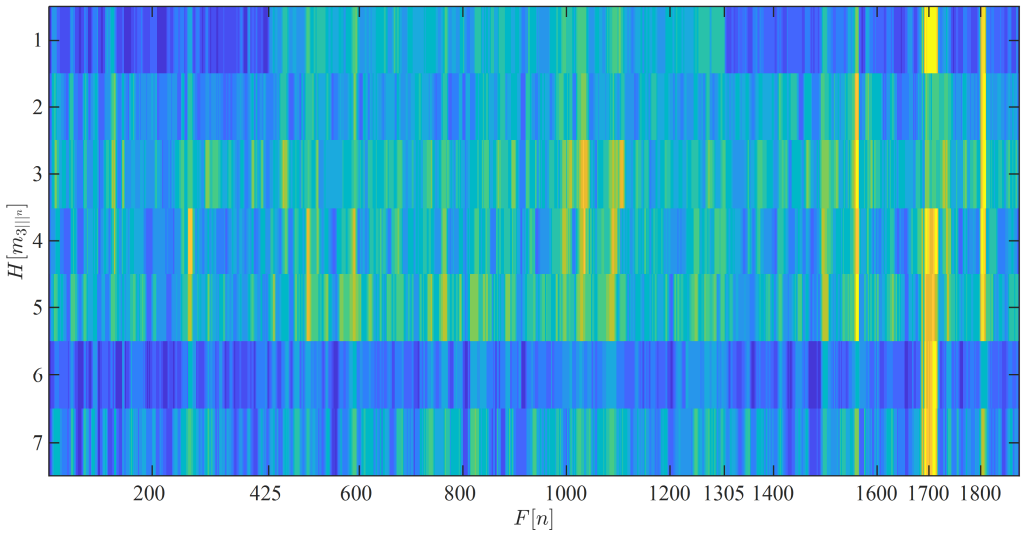


Fig. 6.38. Segment of quantized feature vector corresponding to the Shannon's entropy H applied to the skewness multiscalor components ($m_{3|||^4}$ to $m_{3|||^7}$).

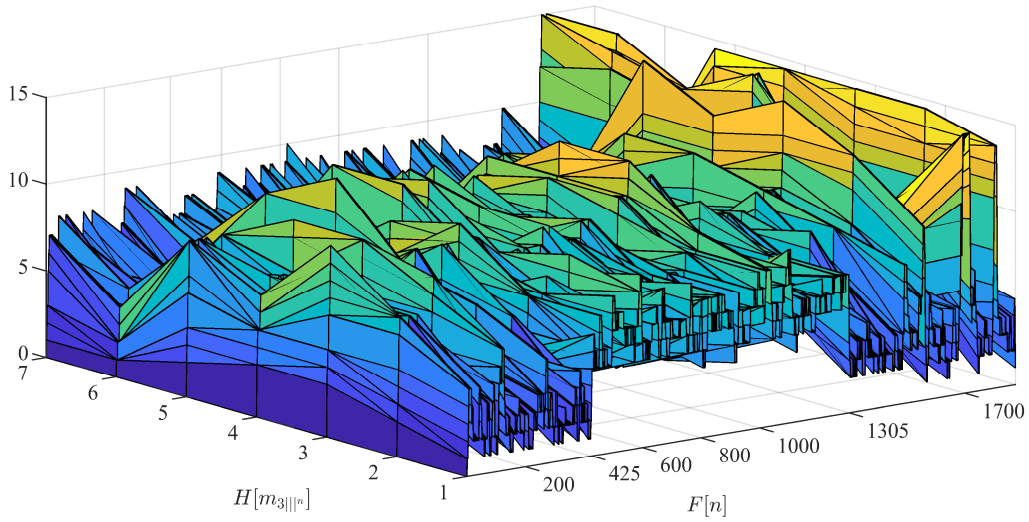


Fig. 6.39. Segment of quantized feature vector corresponding to the Shannon's entropy *H* applied to the skewness multiscalor components ($m_{3|||^1}$ to $m_{3|||^7}$).

6.12.8 Ensemble of Features Stemming from Secondary Operators Applied to Skewness Multiscalor

All the secondary operators, cumulative sum, ZCR, and Shannon's entropy, outcomes when processing the skewness multiscalor components are shown in Fig. 6.40 where in the horizontal rows shown, the first to the seventh describe the cumulative sum outcomes, the eight to the 14th describe the ZCR outcomes, and the 15th to the 21st describe the Shannon's entropy outcomes. A complementary surface tridimensional plot is displayed in Fig. 6.41 where the amplitude differences among the three sets of outcome operators are registered.

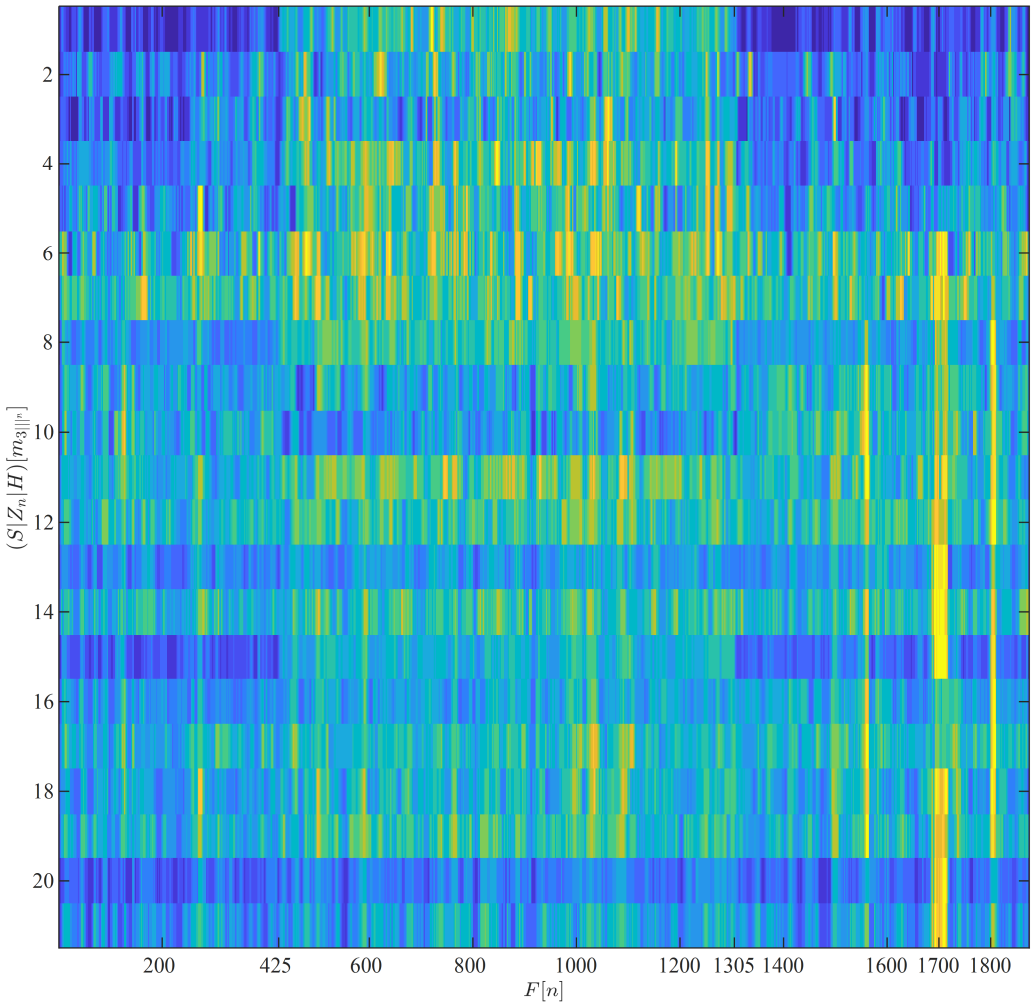


Fig. 6.40. Segment of quantized feature vector corresponding to the secondary operators applied to the skewness multiscalor components $(m_{3|||^4}$ to $m_{3|||^7})$.

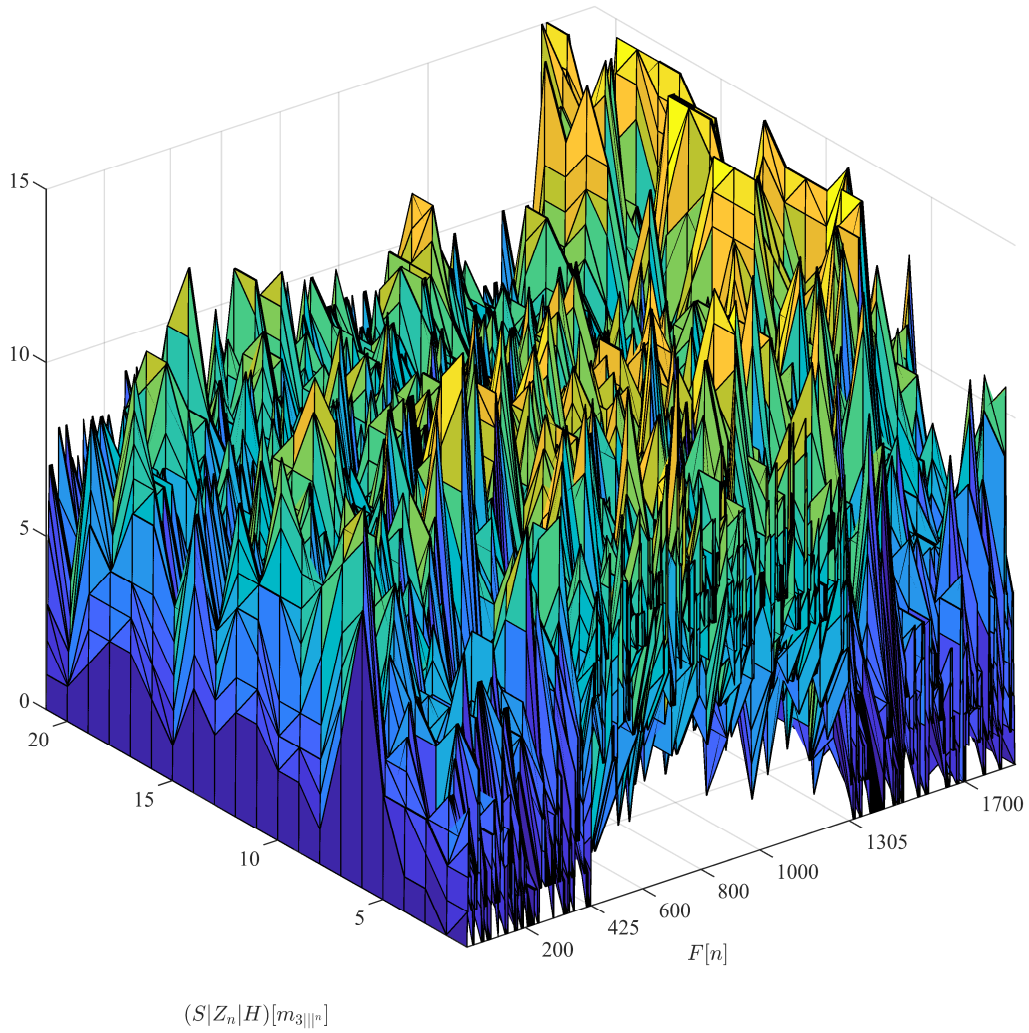


Fig. 6.41. Segment of quantized feature vector corresponding to the secondary operators applied to the skewness multiscalor components ($m_{3|||^4}$ to $m_{3|||^7}$).

6.13 Preparation of Feature Vector for FuzzyART

From the presented multiscale analysis of the dataset including the DDoS attacks, one of the variations of the ART machine learning can be fitted. This approach is FuzzyART, which has been documented previously in this thesis background. FuzzyART can provide classification outcomes utilizing real valued feature vectors. Hence, the quantization stage, after utilizing the secondary operators, is omitted and the essential feature vector for FuzzyART is then produced.

Nevertheless, the features in the vector are subject to denoising and non-linear filtering.

The feature vector for FuzzyART subsequently packages 42 real valued scalars: Seven scalars from the cumulative sum applied to the variance multiscalor, seven scalars from the ZCR applied to the variance multiscalor, seven scalars from entropy applied to the variance multiscalor, seven scalars from the cumulative sum applied to the skewness multiscalor, seven scalars from the ZCR applied to the skewness multiscalor, and seven scalars from entropy applied to the skewness multiscalor.

Plots for visualizing the specific details of each secondary operator applied to the multiscalor operators and creating the features for the FuzzyART feature vectors are omitted to condense the size of this thesis. However, specifics about every feature vector are covered extensively in section 6.12 and particulars about their detection quality are encapsulated in Table II. Furthermore, plots visualizing the ensembles of the three secondary operators applied to both multiscalors are included.

6.13.1 Ensemble of Features Stemming from Secondary Operators Applied to Variance Multiscalor

Figure 6.42 visualizes the real-valued outcomes from the secondary operators (cumulative sum, ZCR, and Shannon's entropy) when processing the variance multiscalor components. From the horizontal rows in Fig. 6.42, the first to the seventh describe the cumulative sum outcomes, the eight to the 14th describe the ZCR outcomes, and the 15th to the 21st describe the Shannon's entropy outcomes. The surface tridimensional plot in Fig. 6.43 show real valued amplitude differences among the three sets of secondary operators outcomes.

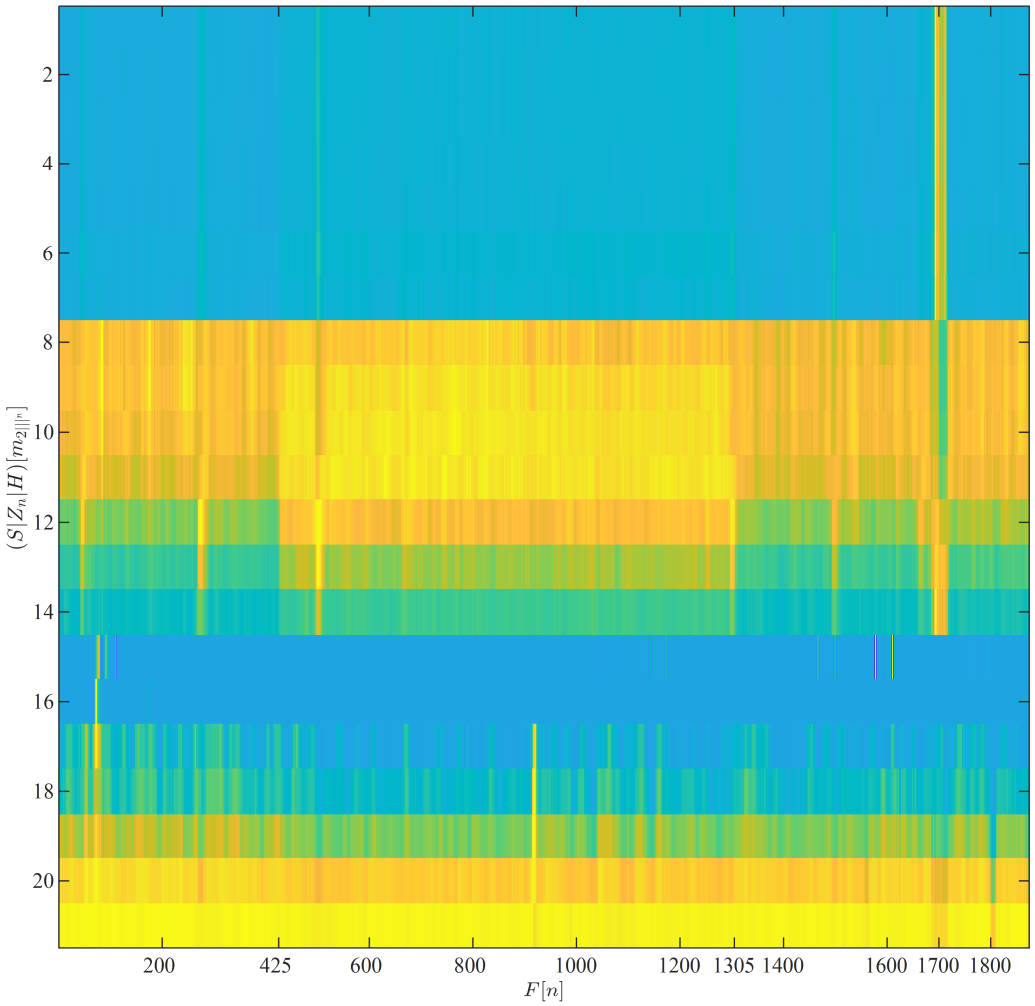


Fig. 6.42. Segment of normalized feature vector corresponding to the secondary operators applied to the variance multiscalor components $(m_{2|||^1}$ to $m_{2|||^7})$.

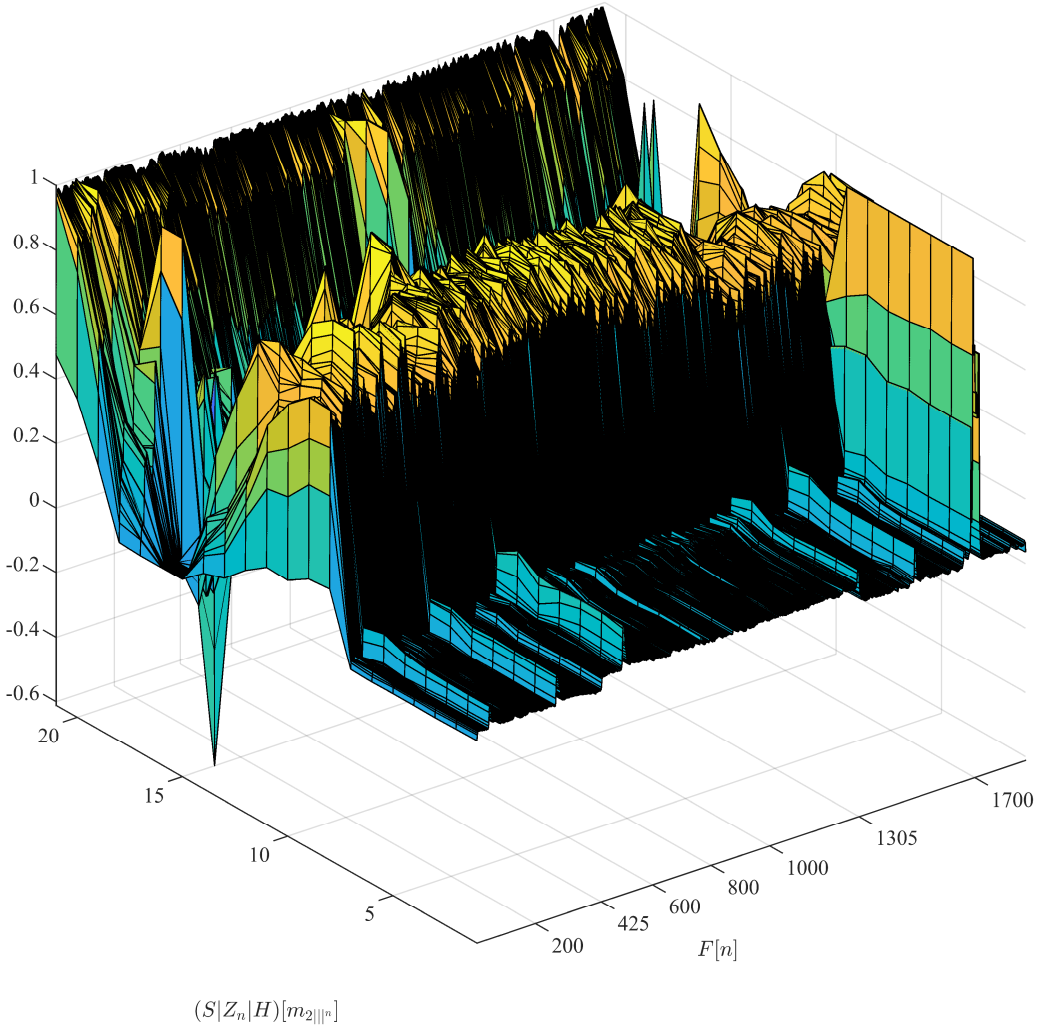


Fig. 6.43. Segment of normalized feature vector corresponding to the secondary operators applied to the variance multiscalor components ($m_{2|||^1}$ to $m_{2|||^7}$).

6.13.2 Ensemble of Features Stemming from Secondary Operators Applied to Skewness Multiscalor

The real valued outcomes of the cumulative sum, ZCR, and Shannon's entropy, acting as secondary operators on the skewness multiscalor components are shown in Fig. 6.44, where in the horizontal rows shown, the first to the seventh describe the cumulative sum outcomes, the

eight to the 14th describe the ZCR outcomes, and the 15th to the 21st describe the Shannon's entropy outcomes. A complementary surface tridimensional plot is displayed in Fig. 6.45 where the real valued amplitude differences among the three sets of outcome operators are registered.

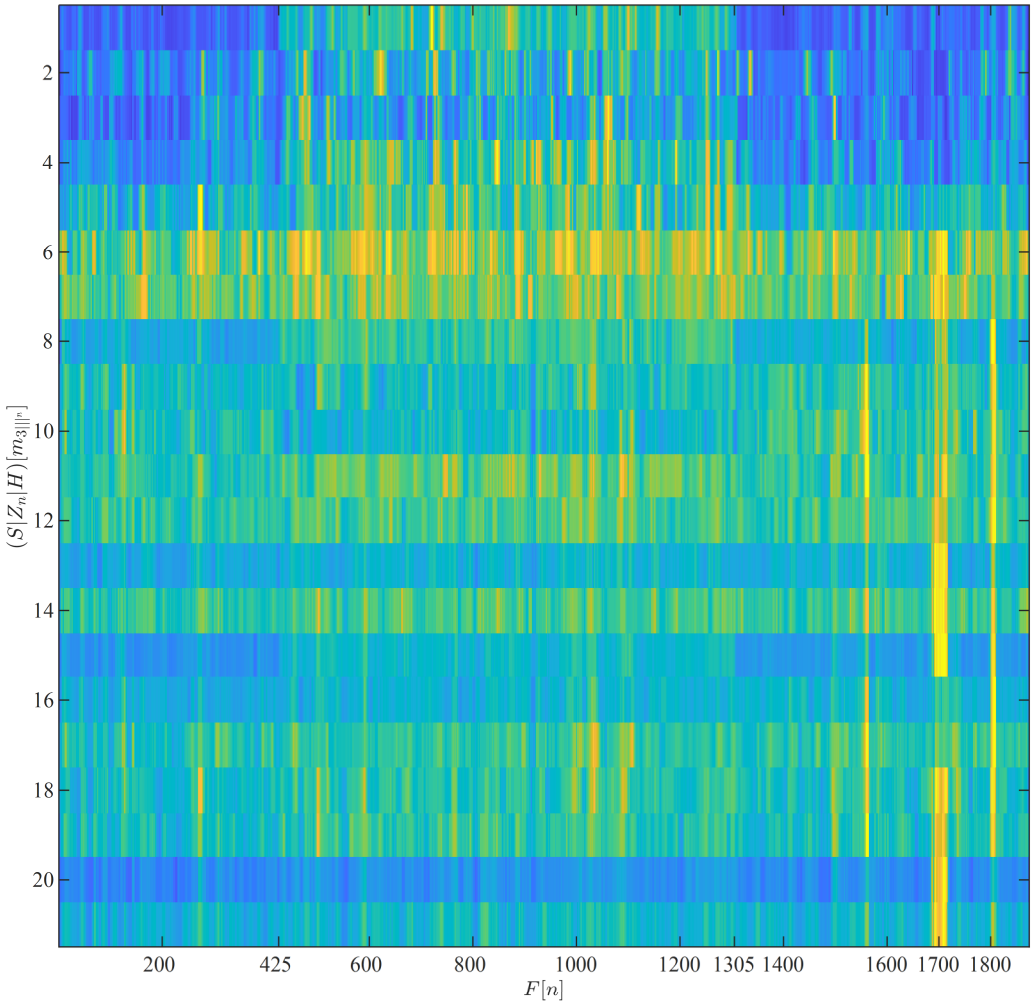


Fig. 6.44. Segment of normalized feature vector corresponding to the secondary operators applied to the skewness multiscalor components ($m_{3|||^1}$ to $m_{3|||^7}$).

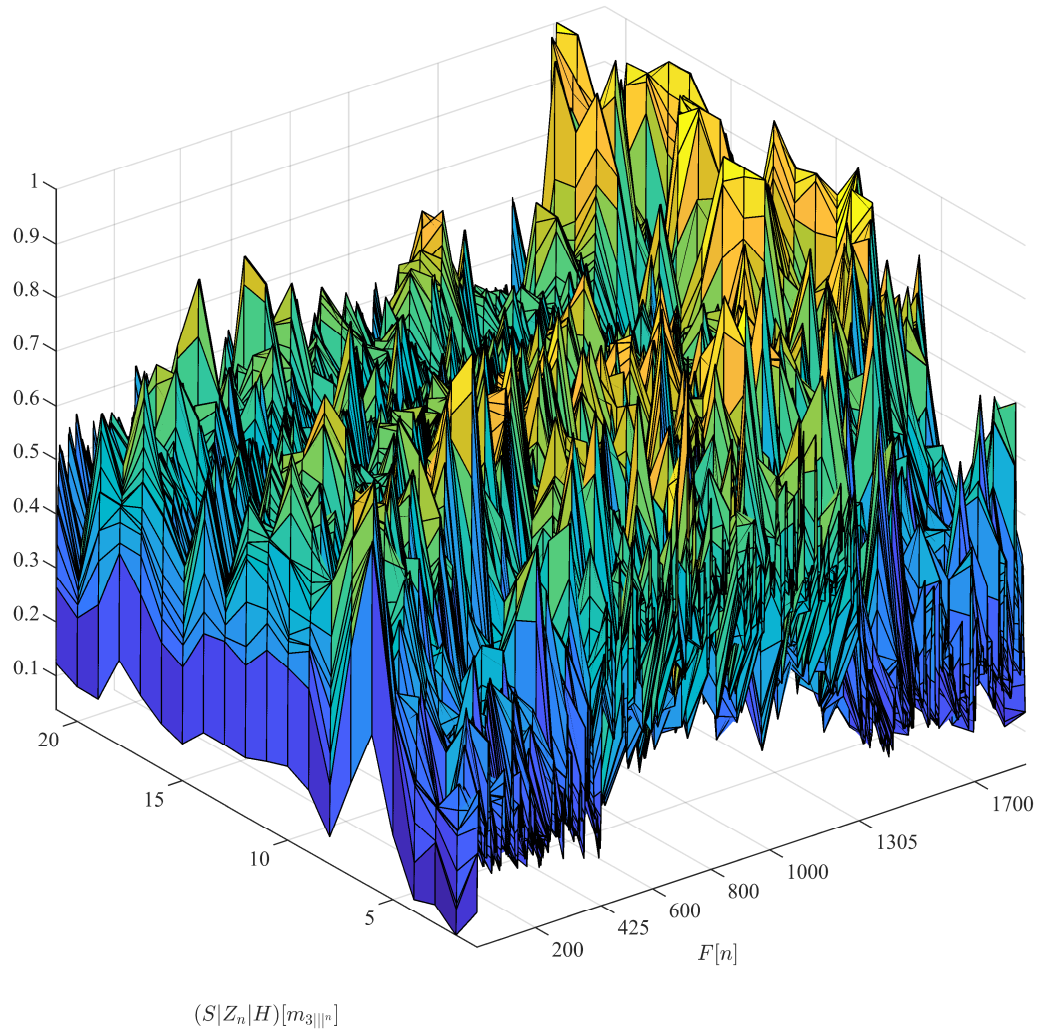


Fig. 6.45. Segment of normalized feature vector corresponding to the secondary operators applied to the skewness multiscalor components ($m_{3|||^4}$ to $m_{3|||^7}$).

6.14 ART1 Classification

6.14.1 ART1 Feature Vector Comprising Secondary Operators Applied to Variance and Skewness Multiscalors

Consequently, pattern recognition is conducted with the premise that conclusive outcomes through the application of ART would become apparent when classifying instances of

the feature vector.

From the feature vector prepared in section 6.12, presented earlier, one shall recall that ART1 only processes data in its binary representation. Hence, the corresponding binary encoding of the quantized values for both variance and skewness multiscalors shown in Figs. 6.32 and 6.40 are shown here in Figs. 6.46 and 6.47 respectively.

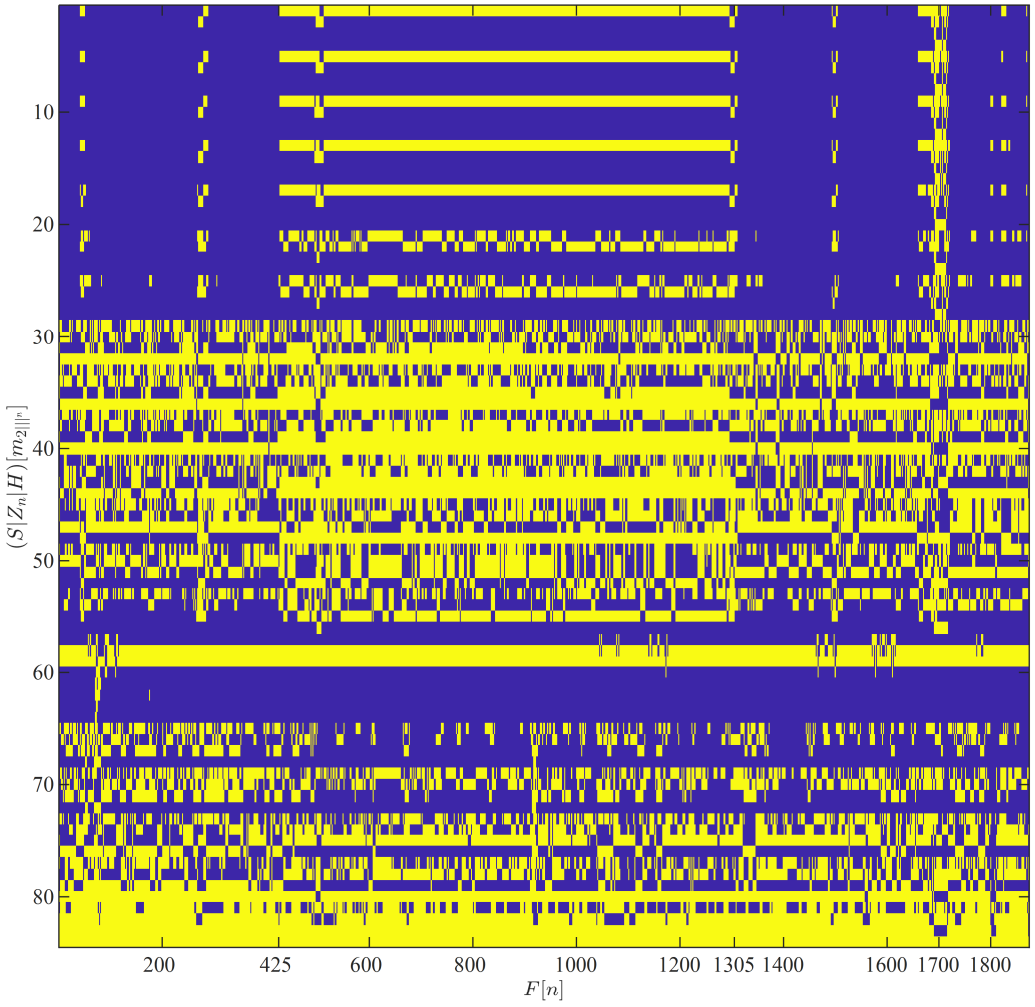


Fig. 6.46. Binary representation, in a four bits word, of the quantized values of the secondary operators applied to the variance multiscalor.

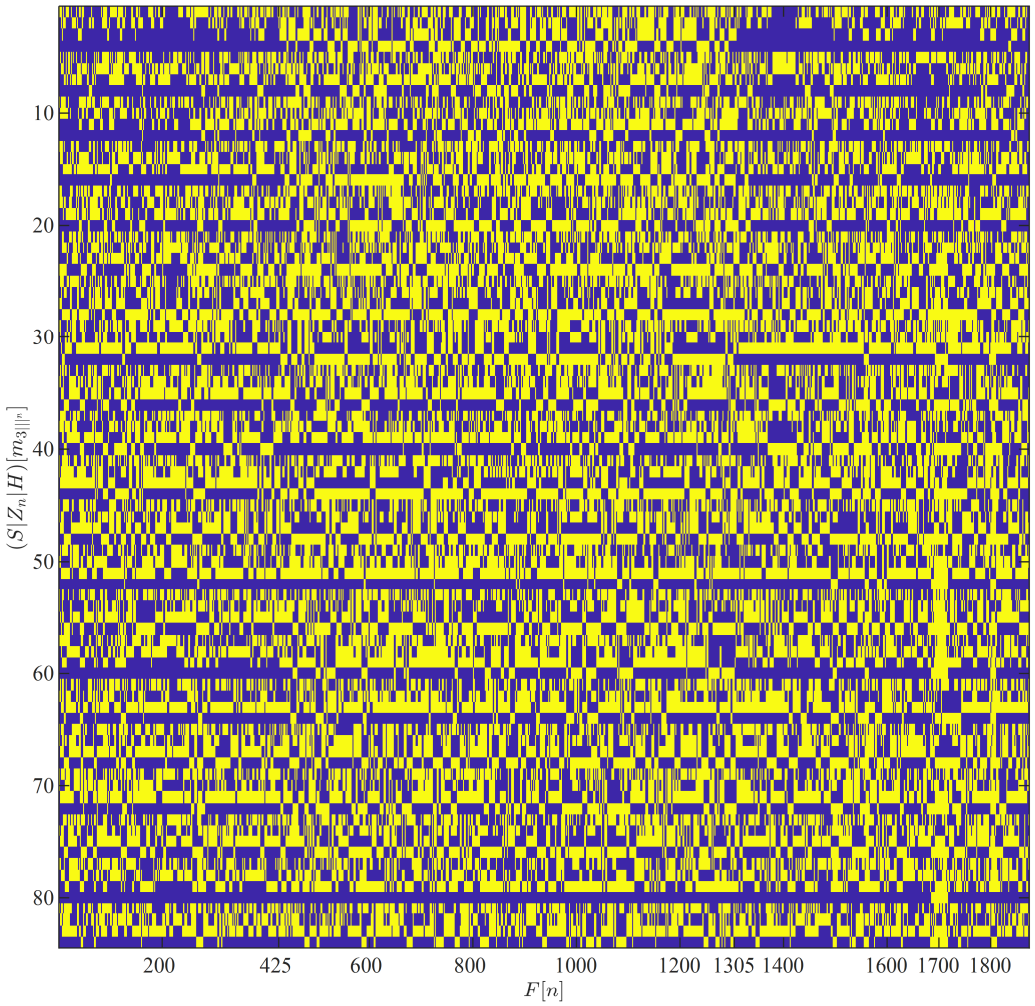


Fig. 6.47. Binary representation, in a four bits word, of the quantized values of the secondary operators applied to the skewness multiscalor.

Upon close inspection of Figs. 6.46 and 6.47, it is seen that the number of rows grows from 21 to 84 due to the binary representation required for ART1 making a feature vector containing 168 binary scalars. Also, clearer patterns for both the DNS and H&R DDoS attacks are observed in Fig. 6.46 (variance multiscalor in binary representation) when comparing it with Fig. 6.47 (skewness multiscalor in binary representation).

6.14.2 Classifications on ART1 Feature Vector

For finding a good vigilance parameter ρ for ART1, a recommended rule of thumb is to set it to 0.9 and then observe if the classes created resemble in any form the real structure of the data under analysis. The outcome of this rule of thumb value for $\rho = 0.9$ is implemented in Fig. 6.48. The results shown in this figure do not resemble in any form neither of the DDoS attacks as indicated for the number of classes created (over 1600) by ART1, almost totalling the number of feature vectors FV_n. The number of neurons $F2_j$ in Fig. 6.48 ranking almost the same as the number of feature vectors indicates a high degree of specialization (overfitting) of the neural network.

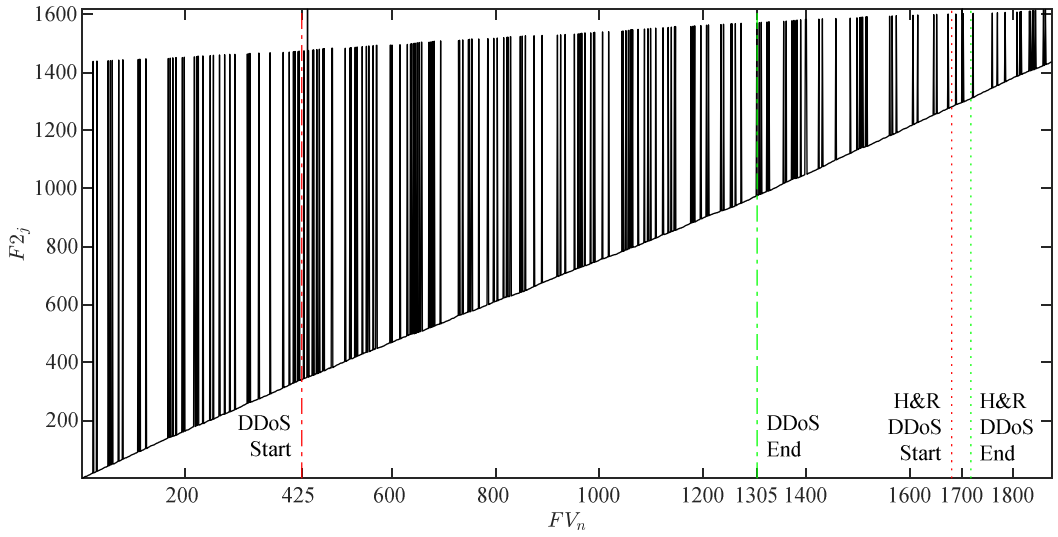


Fig. 6.48. Unsupervised classification of feature vectors FV_n (42 active features for each) with a vigilance parameter value of $\rho = 0.9$.

It is then necessary to depart to another vicinity of values for finding a good vigilance parameter ρ. Since the previous outcome in Fig. 6.48 is highly specialized (overfitted), setting a value close to 0 is appealing. Hence the vigilance parameter is set to $\rho = 0.1$. The outcome for this value is shown in Fig. 6.49, where remarkable achievements become apparent. First of all, the number of classes diminished drastically to just 20; secondly, the DNS DDoS attack is

identified very clearly in its beginning and end almost in its entirety; thirdly, the H&R DDoS attack is also visible in an outstanding form.

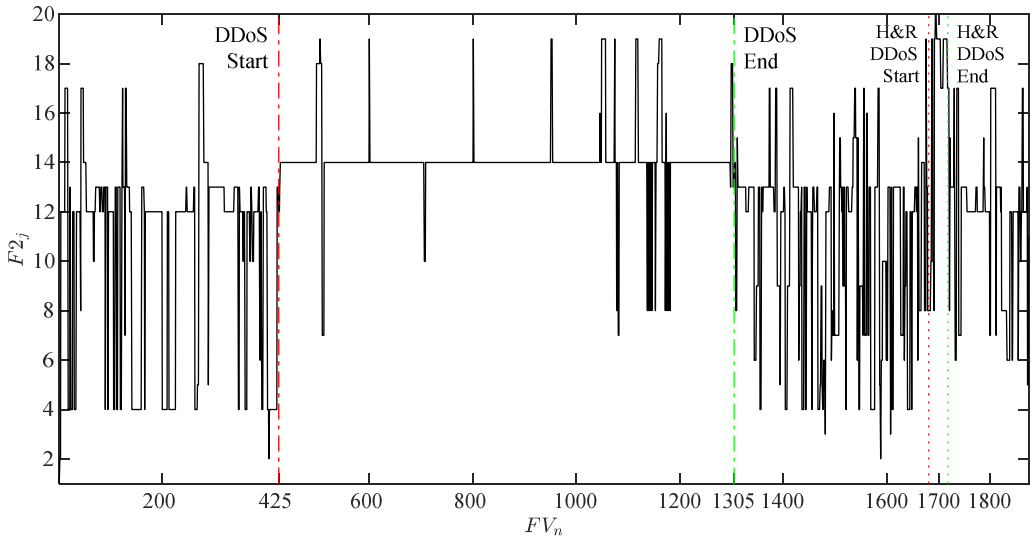


Fig. 6.49. Unsupervised classification of feature vectors FV_n (42 active features for each) with a vigilance parameter value of $\rho = 0.1$.

Figure 6.50 shows the outcome for a value of $\rho = 0.07$ for the vigilance parameter. A smaller number of classes, 14 only, is seen and a clearer shape for the DNS DDoS attack is remarkably outlined.

6.14.3 ART1 Parametogram

Nevertheless, one could get genuinely concerned about setting trial values for the vigilance parameter ρ. There is no reported method in the literature for finding a suitable value candidate for ρ in ART1. Hence, this research proposes a method addressing this ART1 shortcoming for attempting to provide a degree of certainty for choosing a suitable value for the vigilance parameter ρ. This method for drawing suitable values for ρ consists in: (i) Training ART1 for $\rho = 0$ to $\rho = 1$ in increments of 0.001; (ii) making the hypothesis that if a number of elements higher than 660 has been found in a class this would correspond with a high degree of certainty to the DNS DDoS attack since this is represented by 880 feature vectors. Roughly this

step looks for matching at least 75% of the feature vectors to the DNS DDoS attack real occurrence. This step is described mathematically by $\mathrm{Mo}|FV_n|$, where Mo represents the mode and FV_n is the n^{th} feature vector presented to ART1; (iii) finding the onsets for the beginning and end that encapsulate cases of occurrences around 660 events in a class. The outcome of this method is shown in Fig. 6.51.

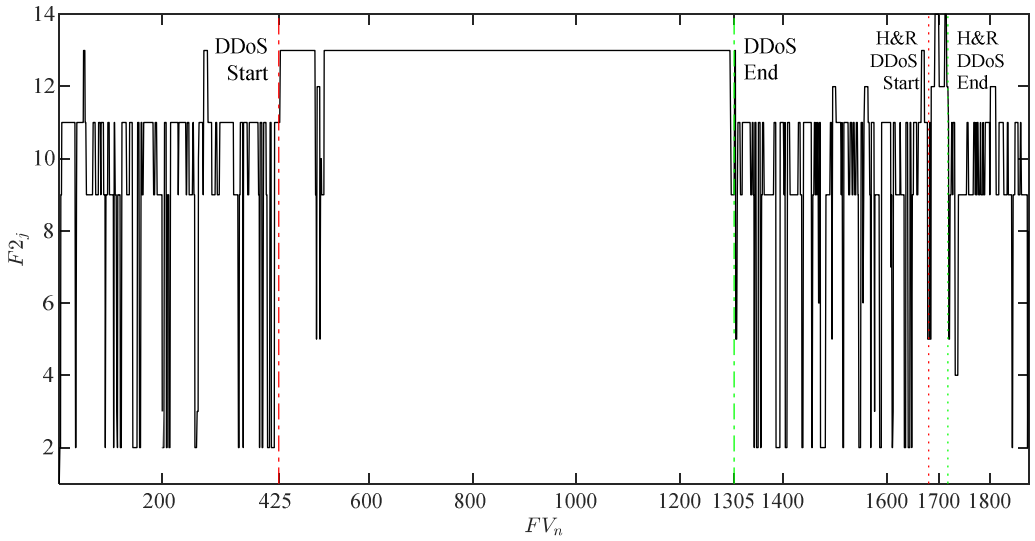


Fig. 6.50. Unsupervised classification of feature vectors FV_n (42 active features for each) with a vigilance parameter value of $\rho = 0.07$.

Two vertical dash-dotted lines divide the waveform presented in Fig. 6.51 into three zones: (i) *Overgeneralization* (OG) zone where the values for ρ fall in the interval [0, 0.07]. This zone would merge classes that are unrelated into a similar one; (ii) *Class-of-Interest* (COI) zone where the values ρ fall in the interval (0.07, 0.216]. Values for ρ that can classify the event of interest into a single class can be found; and (iii) *Overspecialization* (OS) zone where the values for ρ fall in the interval (0.216, 1]. This zone would split events that belong into the same class into two or more separate classes.

The COI zone shows values of the vigilance parameter that starts with the dash-dotted red line (set at a value $\rho = 0.07$) and finishes with the dash-dotted green line (set to a value of

$\rho = 0.216$). The top blue horizontal dash-dotted line represents 880 occurrences corresponding to the DNS DDoS attack. The bottom blue horizontal dash-dotted line is set at 660 occurrences, which arbitrarily represents 75% of the occurrences within the DNS DDoS attack. The OG zone (to the left of the dash-dotted red line) shows a number of occurrences higher to the top blue line, which means that more than 880 cases where placed into a class by ART1 (denoting a high degree of generalization). The OS zone (to the right of the dash-dotted green line) shows a rapid decay for the number of occurrences in the class containing the most elements, which means that less than 880 cases where placed into a class by ART1 (denoting a high degree of specialization as ρ gets close to 1).

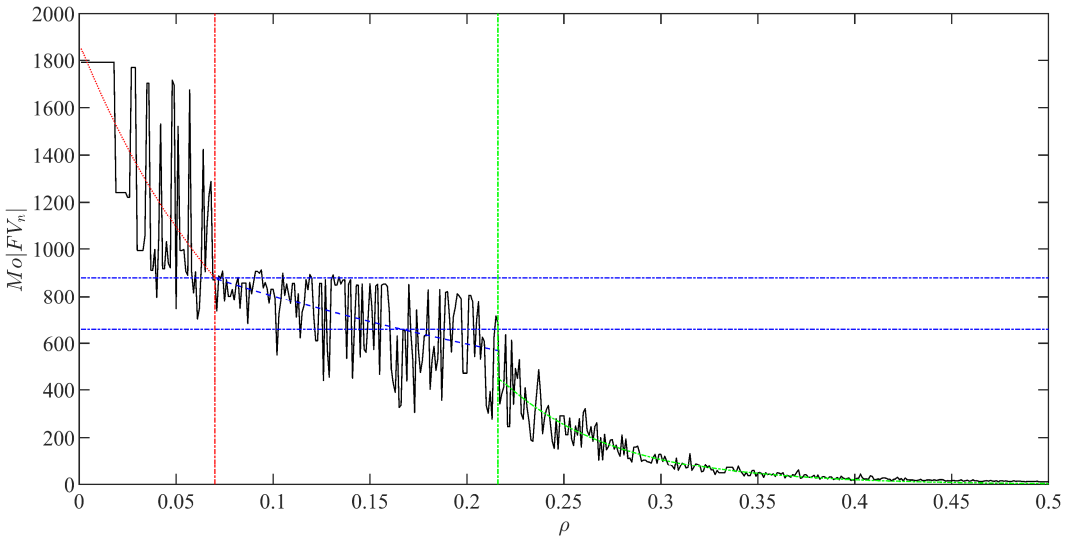


Fig. 6.51. Unsupervised classification of feature vectors FV_n (42 active features for each represented by a four bits binary word) with vigilance parameter values for ρ spanning in the interval $[0, 1]$.

The three distinct zones present in Fig. 6.51 are generalized through piecewise single term exponentials of the form $C_1 e^{\rho C_2}$, where C_1 and C_2 are exponential coefficients and e is the natural logarithm. Exponentials are selected as they provide a better fit for the ART1 vigilance parameter curve. The optimization curve fitting Trust-Region algorithm is used to find the coefficients characterizing each zone. The following equation represents this analysis in a compact form:

$$\mathrm{Mo} \mid FV_n \models \begin{cases} 1.874\mathrm{x}10^3(e^{-10.79\rho}) & \text{for} \quad \rho = [0,\ 0.07] & \text{Overgeneralization} \\ 1.080\mathrm{x}10^3(e^{-2.961\rho}) & \text{for} \quad \rho = (0.07,\ 0.216] & \text{COI} \\ 1.804\mathrm{x}10^4(e^{-17.05\rho}) & \text{for} \quad \rho = (0.216,\ 1] & \text{Overespecialization} \end{cases}$$

6.14.4 Confusion Matrix for Assessing ART1 Classification Performance

To this point, a procession of different data science methodologies and techniques have been discussed and implemented. This procession ranges from data access, data cleaning (removal of corrupt entries), data preprocessing, feature extraction, and machine learning modeling. The performance of the machine learning models is of particular interest. Hence, it is where the confusion matrix comes into the spotlight as it is a performance measurement for machine learning classification. Classification results are often presented in the form of a confusion matrix, a table where the header/sum of every *row* is the *actual/true* COI and the header/sum of every *column* is the *detected/predicted class*. In a confusion matrix, the number of correctly classified samples accumulates in the matrix diagonal. Falsely classified ones will be found outside of the diagonal [Cai011]. *False positives* (a record that is classified as negative but is actually positive) fall above the diagonal, while *false negatives* (a record that is classified as positive but is actually negative) fall below the diagonal. The *overall error rate*, or simply *error rate*, is the sum of the false negatives and false positives, divided by the total number of records $ER = \Sigma(fn + fp) / \Sigma n$. To find the *false negative rate*, divide the number of false negatives by the total number of negative classifications $FNR = \Sigma fn / \Sigma tn$. Similarly, to find the *false positive rate*, divide the number of false positives by the total number of positive classifications $FPR = \Sigma fp / \Sigma tp$ [Laro005].

More formally a confusion matrix is a 2D array of size $J_{cm} \times J_{cm}$ (where J_{cm} is the total number of classes) used to report results of classification experiments. The value in *row i*, *column j* indicates the number of times an object whose *true class* is *i* was labeled as belonging to class *j*. The main diagonal of the confusion matrix indicates the number of cases where the classifier was successful; a perfect classifier would show all off-diagonal elements equal to zero [Marq011].

The measures of performance used in image retrieval borrow from the field of

(document) information retrieval and are based on two primary figures of merit: Precision and recall. *Precision* is the number of relevant documents retrieved by the system divided by the total number of documents retrieved (*i.e.*, true positives plus false positives). *Recall* is the number of relevant documents retrieved by the system divided by the total number of relevant documents in the database (which should, therefore, have been retrieved) [Marq011].

Precision can be interpreted as a measure of exactness, whereas recall provides a measure of completeness. A perfect precision score of 1.0 means that every retrieved document (or image in our case) was relevant, but this situation does not provide any insight as to whether all relevant documents were retrieved. A perfect recall score of 1.0 means that all relevant images were retrieved, but this scenario says nothing about how many irrelevant images might have also been retrieved [Marq011].

Precision, *P*, and *recall*, *R*, measures can also be adapted to and used in classification tasks and expressed in terms of true positives (TP), false positives (FP), and false negatives (FN) as $P = \Sigma tp / \Sigma(tp + fp)$ and $R = \Sigma tp / \Sigma(tp + fn)$ [Marq011]. Additional metrics about machine learning models performance measurement are listed in Appendix D.

In this case, a precision score of 1.0 for a given class means that every item labeled as belonging to that class does indeed belong to the given class, but it says nothing about the number of items from the class that were not labeled correctly. A recall score of 1.0 means that every item from a given class is labeled as belonging to the class, but it says nothing about how many other items are incorrectly labeled as belonging to the class [Marq011].

To keep things consistent, the performance of the machine learning models used in this research (*e.g.*, ART1 for a vigilance parameter $\rho = 0.07$) are analyzed through confusion matrices. The assumptions for examining this confusion matrix analysis are: (i) The feature vectors extracted from the data represented by the packets data rate $S\!.\!l[n]$ (where a time interval of 1.0486 seconds is set) would be capable of classifying positively the DNS DDoS attack for n in the interval [425, 1305], (ii) the data made available for this research provides no labels for the presence of DDoS attacks. As a consequence, through careful study of this data is how the interval just mentioned is set as the best candidate for the DNS DDoS detection, which occupies a 42 percent of the overall traffic data accessed, (iii) the highest occurrence class provided by

ART1 for $\rho = 0.07$ is labelled as "DDoS Attack", and (iv) the rest of the classes are joined and labelled as "Clear Traffic". Since a confusion matrix serves as guiding means to collapse classes if they are found positioned too close in the chosen feature space [Cai011]. This is useful to recognize falsely assumed dissimilarity between those classes and collapse them [Cai011] since in this case they are within the clear traffic space.

A confusion matrix for ART1 set to a vigilance parameter $\rho = 0.07$ is displayed in Fig. 6.52.

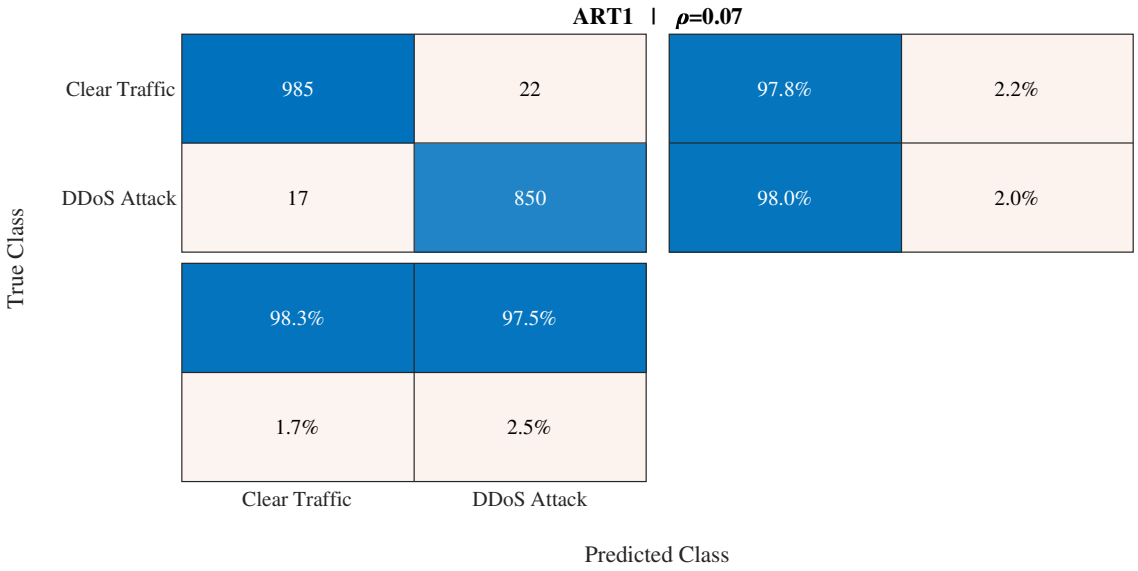


Fig. 6.52. Confusion matrix for ART1 with vigilance parameter $\rho = 0.07$. The matrix displays: (i) 985 cases for clear traffic, (ii) 850 cases for a DDoS attack, (iii) 22 false cases for a DDoS attack, and (iv) 17 false cases for clear traffic. The column normalization (precision): (i) 98.3% for clear traffic, and (ii) 97.5% for a DDoS attack. The row normalization (recall): 97.8% for clear traffic, and (ii) 98% for DDoS attack.

6.14.5 Selected Classifications Based on ART1 Parametogram COI

This subsection explores some cases that fall within the COI zone in Fig. 6.51, which has values of the vigilance parameter starting with the dash-dotted red line (set at a value $\rho = 0.07$) and finishing with the dash-dotted green line (set to a value of $\rho = 0.216$). Hence, it is in this interval [0.07, 0.216] where some specific values for the vigilance parameter ρ are chosen. One experiment that corresponds to a rule of thumb is represented by $\rho = 0.9$. Its extreme counterpart

is set to $\rho = 0.1$. These two values are included for completeness as they have been discussed in the text. Three specified experiments drawn from the ART1 parametogram (Fig. 6.51) are chosen for the values of $\rho = 0.07$, $\rho = 0.088$, and $\rho = 0.09$, which are expected to have a good high precision and recall. The information pertaining to five experiments (including the three specified experiments and the two guesses) is shown in Table III where specifics about the feature vector components and classification metrics are found (experiments of interest are shaded).

TABLE III
ART1 CLASSIFICATION METRICS FOR DETECTION OF A DDOS ATTACK

Machine Learning Model	Vigilance Parameter ρ	Components in Feature Vector							Classification Metrics		
		Variance Multiscalor			Skewness Multiscalor			Total	DDoS True Positives	Precision	Recall
		S	Z	H	S	Z	H				
ART1	0.07	7	7	7	7	7	7	42	850	97.5%	98%
	0.088								848	98.5%	97.8%
	0.09								848	98.5%	97.8%
	0.1								807	97%	93.1%
	0.9								3	100%	0.3%

Confusion matrices for the distinct values of the vigilance parameter ρ in ART1 shown in Table III are included in Appendix J as Figs. J.1 to J.5.

6.15 Findings About ART1 Classification

The trial and error experiments for ART1 with vigilance parameter with values of $\rho = 0.1$, and $\rho = 0.9$, have unbalanced values for precision and recall of 97% and 93.1%, and of 100% and 0.3% respectively for the detection of DDoS attack occurrences.

The vigilance parameter with a value of $\rho = 0.1$ holds the following deductions. Some of the DDoS attack occurrences (60) are misclassified into clear traffic and a lower number of the true occurrences (25) for the clear traffic are also missed. This number of misclassifications is not excessive

For the vigilance parameter with a value of $\rho = 0.9$, the following inferences are collected. A high number of DDoS attack occurrences (864) are misclassified into clear traffic and the entirety of the true occurrences (1007) for the clear traffic are detected. The number of

misclassifications for the DDoS attack is very high.

From the values for the vigilance parameter in ART1, 0.07, 0.088, and 0.09, which were systematically selected from ART1 corresponding parametogram, well balanced values for both precision and recall are found, 97.5% and 98%, 98.5% and 97.8%, and 98.5% and 97.8% respectively corresponding to the occurrences detection of the DDoS attack.

From the vigilance parameter set to values of $\rho = 0.07$, $\rho = 0.088$, and $\rho = 0.09$, the next outcome is notable. A low number of misclassifications is found for both the clear traffic and the DDoS attack classes.

The overall implementation of ART1 as a machine learning approach, with $\rho = 0.07$, $\rho = 0.088$, and $\rho = 0.09$, is found overperforming when compared to FuzzyART set to the best suitable vigilance parameter values found through the FuzzyART parametogram. This excursion is described next. A more detailed description of these findings is included in Appendix J.

6.16 FuzzyART Classification

6.16.1 FuzzyART Feature Vector Comprising Secondary Operators Applied to Variance and Skewness Multiscalors

Moreover, FuzzyART, an analogous approach to ART1 is utilized. FuzzyART is fully described previously in the background chapters. Both ART1 and FuzzyART have the ART methodology at their core and function under distinct value representations for their vectors. The analysis of the feature vector through an additional unsupervised neural network like FuzzyART provides a supplementary testing scenario useful for classifying the DDoS cyberattacks described by multiscalors and secondary operators.

Once the feature vectors in section 6.13 (described previously) for FuzzyART are prepared, one remembers that FuzzyART requires real valued data representation to perform classification. Consequently, the real valued vectors shown in Figs. 6.42 (variance) and 6.44 (skewness) can be used directly in FuzzyART. The referred real valued feature vectors comprised of secondary operators applied to variance and skewness multiscalors, as primary operators, are resketched in Figs. 6.53 and 6.54 respectively in order to improve both the flow and aid the text comprehension.

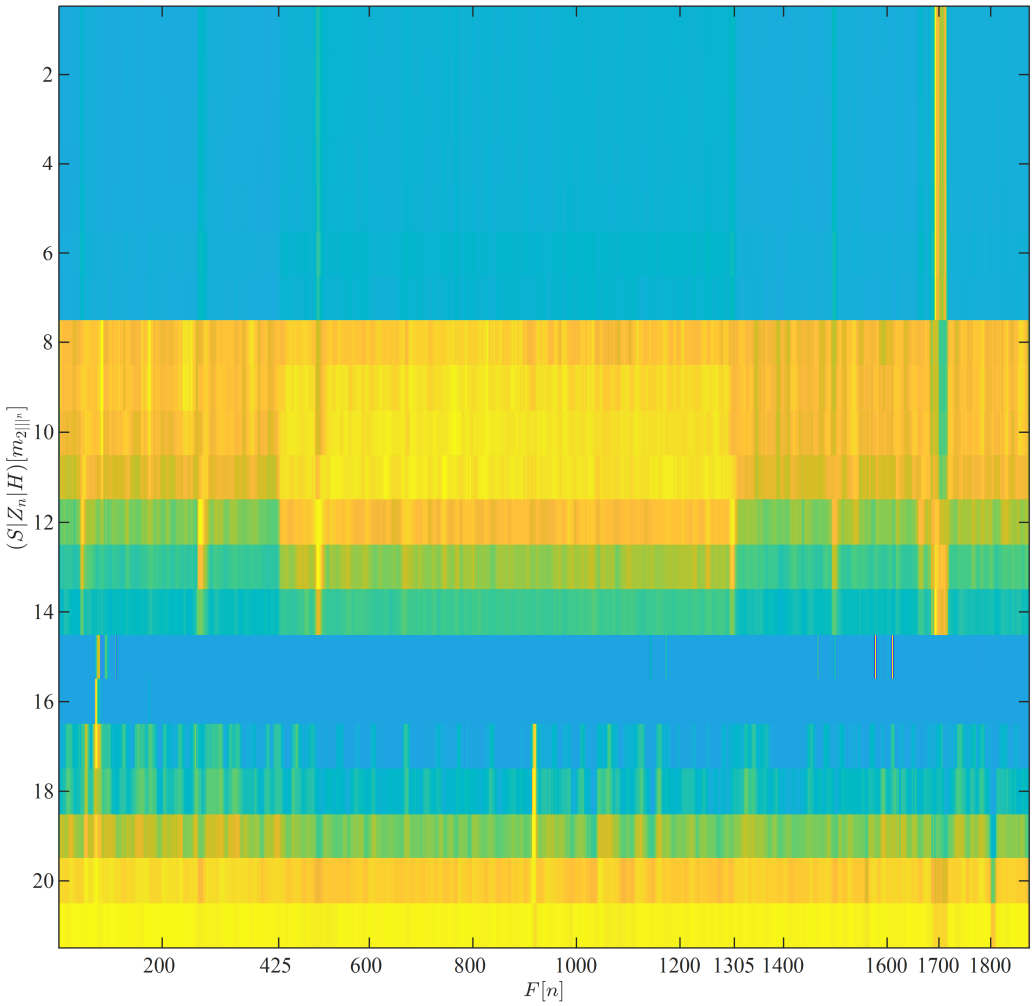


Fig. 6.53. Real valued representation of the non-linear filtered secondary operators applied to the variance multiscalor.

The patterns indicating the presence of both the DNS and H&R DDoS attacks continue to be observed in Fig. 6.53 (variance multiscalor in real valued representation) clearer than in Fig. 6.54 (skewness multiscalor in real valued representation).

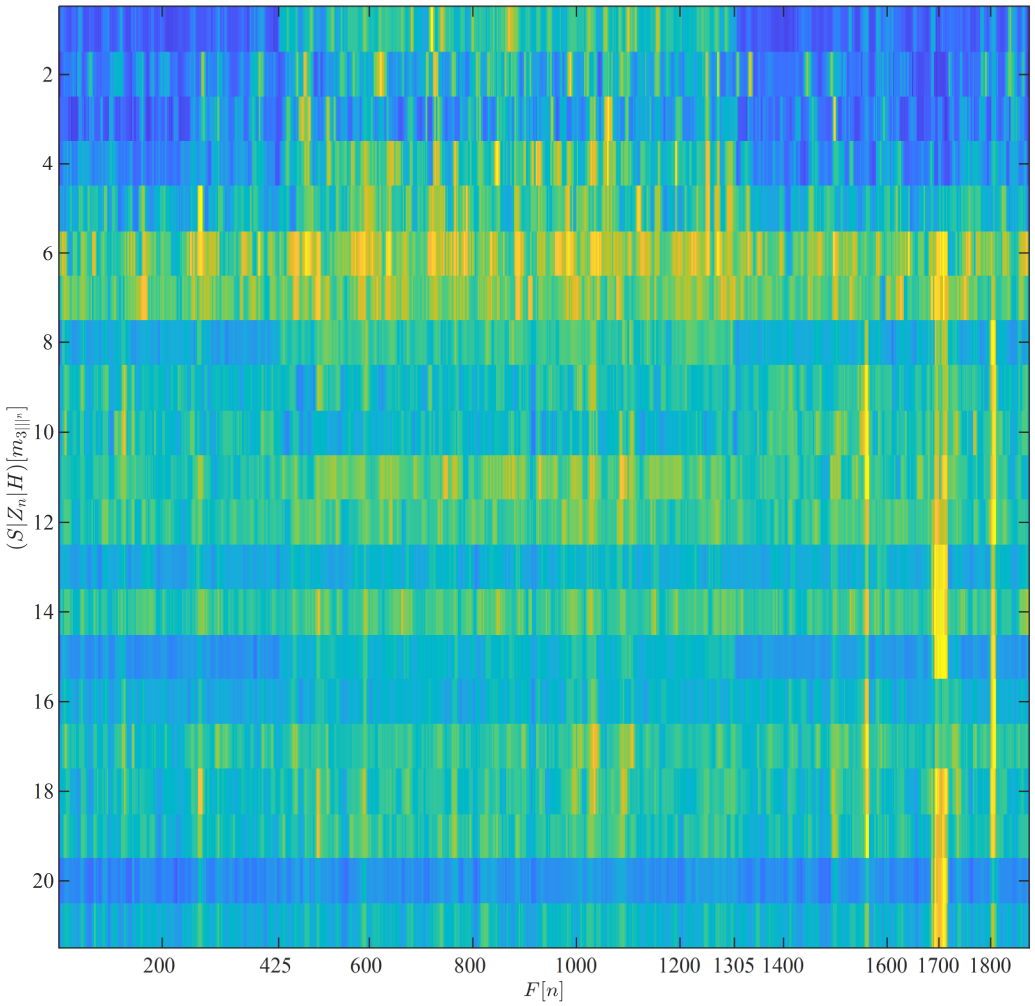


Fig. 6.54. Real valued representation of the non-linear filtered secondary operators applied to the skewness multiscalor.

6.16.2 FuzzyART Parametogram

Analogously, for FuzzyART, there is no reported method in the literature for finding a proper value for the vigilance parameter ρ. Hence, from the experience previously gained for ART1 when defining ρ, the same method to define this parameter is followed for FuzzyART.

This method for defining ρ for FuzzyART, with small modifications fitting the required parameters, consists in: (i) Training FuzzyART for $\rho = 0$ to $\rho = 1$ in increments of 0.001, while

assigning constant values to parameters $\alpha = 0.01$ and $\beta = 1$; (ii) setting a similar hypothesis that if a number of elements higher than 660 has been found in a class this would correspond with a high degree of certainty to the DNS DDoS attack, which is represented by 880 feature vectors instances. This step looks for matching at least 75% of the feature vectors to the DNS DDoS attack real occurrence. This step is represented mathematically by $\mathrm{Mo}|FV_n|$, where Mo represents the mode (as in the experiments previously described with ART1) and FV_n is the n^{th} feature vector presented to FuzzyART; (iii) finding the onsets for the beginning and end that encapsulate cases of occurrences around 660 events in a class. The outcome of the described method is shown in Fig. 6.55.

Analogously, two vertical dash-dotted lines divide the waveform presented in Fig. 6.55 into three zones: (i) The OG zone where the values for ρ fall in the interval [0, 0.578]. This zone would group classes that are otherwise unrelated into a similar one; (ii) the COI zone where the values ρ fall in the interval (0.578, 0.664]. The COI contains the more values for ρ that can classify the event of interest into a single class; and (iii) the OS zone where the values for ρ fall in the interval (0.664, 1]. This zone would split events into two or more separate classes that otherwise belong to the same class.

The COI zone shows values of the vigilance parameter that begins with the dash-dotted red line (set at a value $\rho = 0.578$) and concludes with the dash-dotted green line (set to a value of $\rho = 0.664$). The top blue horizontal dash-dotted line represents 880 occurrences corresponding to the DNS DDoS attack. The bottom blue horizontal dash-dotted line is set at 660 occurrences, which represents 75% of the occurrences within the DNS DDoS attack. The OG zone (to the left of the dash-dotted red line) shows a number of occurrences higher to the top blue line, which means that more than 880 cases where placed into a class by FuzzyART (denoting a high degree of generalization). The OS zone (to the right of the dash-dotted green line) shows a rapid decay for the number of occurrences in the class containing the most elements, which means that less than 880 cases where placed into a class by FuzzyART (denoting a high degree of specialization as ρ gets close to 1).

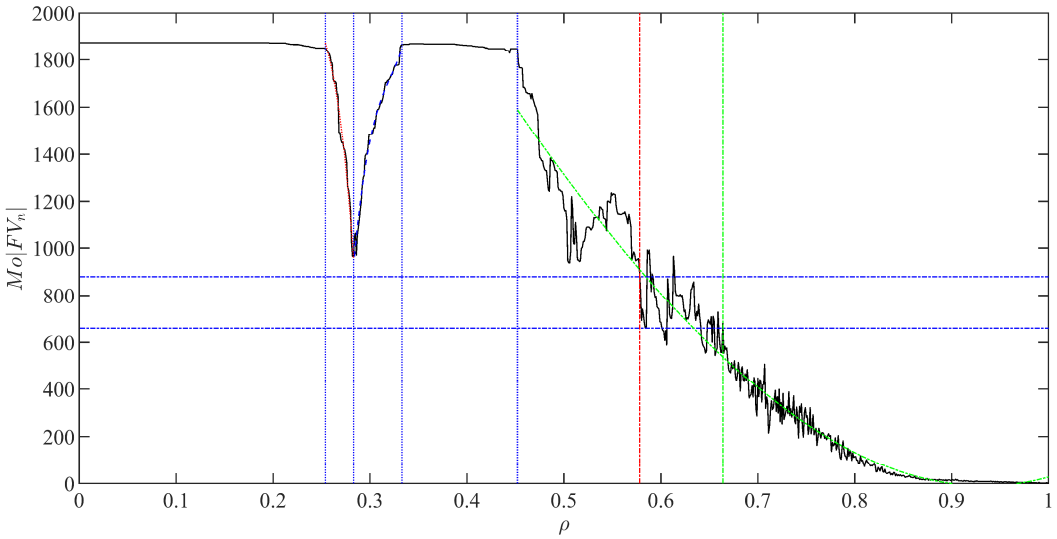


Fig. 6.55. FuzzyART Unsupervised classification of feature vectors FV_n (42 active features represented by real values) with vigilance parameter values for ρ spanning in the interval $[0, 1]$.

The seven distinct zones present in Fig. 6.55 are generalized through piecewise constant intervals and third degree (*aka* cubic) polynomials of the form $a\rho^3 + b\rho^2 + c\rho + d$, where *a*, *b*, *c*, and *d* are the polynomial coefficients. Constant intervals and cubic polynomials are selected because a better fit is achieved for the FuzzyART vigilance parameter curve sections. The optimization curve fitting Trust-Region algorithm is used to find the equations and their coefficients characterizing each zone. The following equations and intervals portray this analysis:

$$\text{Mo} \mid FV_n \models \begin{cases} 1860 \\ \text{for } \rho = [0,\ 0.253] & \text{OG} \\ -1.222\times10^{7}\rho^{3} + 9.101\times10^{6}\rho^{2} - 2.274\times10^{6}\rho - 1.926\times10^{5} \\ \text{for } \rho = (0.253,\ 0.283] & \text{OG} \\ 8.283\times10^{6}\rho^{3} - 7.978\times10^{6}\rho^{2} + 2.571\times10^{6}\rho - 2.753\times10^{5} \\ \text{for } \rho = (0.283,\ 0.332] & \text{OG} \\ 1860 \\ \text{for } \rho = (0.332,\ 0.452] & \text{OG} \\ 2.983\times10^{3}\rho^{3} - 3.405\times10^{1}\rho^{2} - 7.74\times10^{3}\rho + 4.819\times10^{3} \\ \text{for } \rho = (0.452,\ 0.578] & \text{OG} \\ 2.983\times10^{3}\rho^{3} - 3.405\times10^{1}\rho^{2} - 7.74\times10^{3}\rho + 4.819\times10^{3} \\ \text{for } \rho = (0.578,\ 0.664] & \text{COI} \\ 2.983\times10^{3}\rho^{3} - 3.405\times10^{1}\rho^{2} - 7.74\times10^{3}\rho + 4.819\times10^{3} \\ \text{for } \rho = (0.664,\ 1] & \text{OS} \end{cases}$$

6.16.3 Selected Classifications Based on FuzzyART Parametogram COI

Some cases within the COI zone in Fig. 6.55 are explored in this subsection. Herein the vigilance parameter beginning with the dash-dotted red line (set at a value $\rho = 0.578$) and ending with the dash-dotted green line (set to $\rho = 0.664$). Henceforth, the interval [0.578, 0.664] holds specific values of interest for the vigilance parameter ρ. The first experiment is represented by $\rho = 0.1$, while its extreme counterpart is set to $\rho = 0.9$. These two values are included for comparison purposes between FuzzyART and ART1. Three promising experiments are derived from the FuzzyART parametogram (Fig. 6.55) setting values of the vigilance parameter for $\rho = 0.632$, $\rho = 0.633$, and $\rho = 0.634$. The data concerning these five experiments is recapitulated in Table IV where particulars about the feature vector components and

classification metrics are found (relevant experiments are shaded).

TABLE IV

FUZZYART CLASSIFICATION METRICS FOR DETECTION OF A DDOS ATTACK

Machine Learning Model	Vigilance Parameter ρ	Components in Feature Vector							Classification Metrics		
		Variance Multiscalor			Skewness Multiscalor			Total	DDoS True Positives	Precision	Recall
		S	Z	H	S	Z	H				
FuzzyART	0.1	7	7	7	7	7	7	42	867	46.3%	100%
	0.632								735	88.7%	84.8%
	0.633								761	89.7%	87.8%
	0.634								759	88.5%	87.5%
	0.9								14	100%	1.6%

Supplementary confusion matrices corresponding with the distinct values of the vigilance parameter ρ in FuzzyART used for populating information in Table IV are included in Appendix K as Figs. K.1 to K.5.

6.17 Findings About FuzzyART Classification

From the FuzzyART parametogram, it is observed that it shows a high-degree of nonlinear dynamics when comparing it to ART1. It is noticed that the learning section of interest in FuzzyART is smaller than ART1.

The experimental values for FuzzyART vigilance parameter of $\rho = 0.1$, and $\rho = 0.9$, also show unbalanced values for precision and recall of 46.3% and 100%, and of 100% and 1.6% respectively for the detection of DDoS attack occurrences.

A value of $\rho = 0.1$, for the vigilance parameter embraces the subsequent remarks. The entirety of the true occurrences (1,007) for the clear traffic are misclassified as a DDoS attack occurrences.

For the vigilance parameter with a value of $\rho = 0.9$, the next comments are worth noting. A high number of DDoS attack occurrences (853) are misclassified into clear traffic and the entirety of the true occurrences (1007) for the clear traffic are detected. The number of misclassifications for the DDoS attack is very high.

The vigilance parameter in FuzzyART with values, $\rho = 0.632$, $\rho = 0.633$, and $\rho = 0.634$, which also were analytically chosen from the FuzzyART corresponding parametogram, show balanced values for both precision and recall, 88.7% and 84.8%, 89.7% and 87.8%, and 88.5% and 87.5% respectively, which relate to the occurrences detection of the DDoS attack.

From the FuzzyART vigilance parameter with a value of $\rho = 0.632$, $\rho = 0.633$, and $\rho = 0.634$, the next outcome is notable. The number of misclassifications found for both the clear traffic and the DDoS attack classes is higher than ART1, which translates in FuzzyART underperforming when compared with ART1. One shall recall that FuzzyART requires three parameters (α, β, and ρ) for its tunning. When compared to ART1 the FuzzyART shortcomings might be because only the vigilance parameter ρ is subjected to sensitivity analysis in the scope of this research. Nonetheless, finding better operational settings for FuzzyART would require the application of advanced optimization methodologies. A comprehensive account of these findings is contained in Appendix K.

6.18 Summary

This chapter presents the solid results of the experiments outlined for this research to this point. The extent of the research conducted is depicted through the results presented here. The in depth approach to examine the details of the PREDICT dataset have been closely followed and shown graphically. It has been shown that the overall attack traffic is an aggregation of the contributions from the six attack flows. These contributions consider the packets count and data rate sent towards the victim of the DDoS attack. The correctness of the VFD algorithm implementation has also been described.

This chapter also presents the results obtained through a new multiscale analysis methodology, multiscalors, introduced in this thesis. This methodology has been tested with two primary operators, variance and skewness, for a processing frame of 4,096 samples creating seven multiscalor components. Three secondary operators, cumulative sum, ZCR, and Shannon's entropy, are utilized to further analyze and compress, by a factor of a million, the multiscalor components. The work done in the preparation of the feature vectors for both ART1 and

FuzzyART is carefully described. Similarly, explanations about the classification outcomes from the machine learning models stemming from confusion matrices used to analyze their precision are well documented.

This chapter presents results with three perspectives: (i) Very descriptive oriented to document minute details (an extensive collection of plots is present throughout the chapter and in relevant appendices), (ii) visual assessment of the different contributions of each multiscalor component can be easily compared, and (iii) remarking specific findings and observations about the detection quality of all the features derived.

CHAPTER VII

CONCLUSIONS

7.1 Main Findings

The seven variance multiscalor components, $m_{2\text{III}^n}$, appear to have a slightly constant and equal power for detecting both classes of DDoS attacks, DNS amplification and H&R, under the *cumulative sum*, $(S[m_{2\text{III}^n}])$. The detection capacity tends to decrease from the first, $m_{2\text{III}^1}$, to the seventh, $m_{2\text{III}^7}$, variance multiscalor components as the signals become spikier. Figures H.1 to H.7 found in Appendix H, extensively support these conclusions.

When applying ZCR to the variance multiscalor components, $(Z_n[m_{2\text{III}^n}])$, results are not as defined as those coming from the cumulative sum, but the dynamics for both DDoS attacks are maintained and the detection capacity increases from the first, $m_{2\text{III}^1}$, to the seventh, $m_{2\text{III}^7}$, components. However, the dynamics for both DDoS attacks are maintained for all variance multiscalor components (from first, $m_{2\text{III}^1}$, to seventh, $m_{2\text{III}^7}$). The visual quality of the results appears to increase as one traverses from the first multiscalor component, $m_{2\text{III}^1}$, to the seventh, $m_{2\text{III}^7}$. It is worth highlighting that the H&R DDoS attack appears inverted from the first, $m_{2\text{III}^1}$, to forth, $m_{2\text{III}^4}$ and becomes positive for the other components. Nevertheless, the shapes of both DDoS attacks are preserved within all results of the ZCR run on all variance multiscalor components. All components waveforms appear more complex and spikier when compared with the cumulative sum case. A comprehensive support to these inferences is backed up by Figures H.29 to H.35 placed in Appendix H.

The outcomes with less quality, compared with the cumulative sum and ZCR, are

contributed by utilizing *Shannon's entropy* to the variance multiscalor components $(H[m_{2|||^n}])$. For the DNS DDoS attack case there is neither clear beginning nor end found, whereas the H&R attack is found to be represented by an inverse peak in some occurrences. The DC value, in this case Shannon's entropy, of the waveforms increases from the fourth, $m_{2|||^4}$, to seventh, $m_{2|||^7}$, components. These observations are documented from Figs. H.57 to H.60 located in Appendix H.

Regarding the *cumulative sum* utilized with the skewness multiscalor components $(S[m_{3|||^n}])$ results of different visual perception quality are obtained. The DNS amplification DDoS attack appears to have better quality for the first, $m_{3|||^1}$ (Fig. I.1), and fourth, $m_{3|||^4}$ (Fig. I.4), skewness multiscalor components, while a lesser quality for the second, $m_{3|||^2}$ (Fig. I.2), third, $m_{3|||^3}$ (Fig. I.3), fifth, $m_{3|||^5}$ (Fig. N.5), and undiscernible contributions for the rest of the components. For the H&R DDoS attack case, this exhibits better quality in the sixth, $m_{3|||^6}$ (Fig. I.6), and seventh, $m_{3|||^7}$ (Fig. I.7). The shape of both DDoS attacks for the cumulative sum run on all skewness multiscalor components is preserved. These results are not as uniform as the ones obtained with the cumulative sum applied to the variance multiscalor components. Figures I.1 to I.7 found in Appendix I, considerably sustain these observations.

When employing ZCR is applied to the skewness multiscalor components $(Z_n[m_{3|||^n}])$, the DNS amplification DDoS attack appears to have better quality for the first, $m_{3|||^1}$ (Fig. I.29), and a lesser quality for the fourth, $m_{3|||^4}$ (Fig. I.32), and no distinguishable contributions for the rest of the components. Concerning the H&R DDoS attack case, this exhibits better quality from the fifth, $m_{3|||^5}$ (Fig. I.33), to the seventh, $m_{3|||^7}$ (Fig. I.35). The shape of both DDoS attacks for the ZCR applied on all skewness multiscalor components appears in varying quality degrees. The results obtained from ZCR applied to the skewness multiscalor components are not as good as the ones obtained from the variance multiscalor components. Ample support for these deductions is delivered from Figures I.29 to I.35 placed in Appendix I.

From the employment of *Shannon's entropy* on the skewness multiscalor components ($H[m_{3|||^n}]$), the dynamics of the DNS DDoS attack are noticed in the first skewness multiscalor, $m_{3|||^1}$ (Fig. I.57), while the dynamics of the H&R DDoS attack are noticed in the first skewness multiscalor, $m_{3|||^1}$ and from fourth to seventh skewness multiscalors, $m_{3|||^4}$ to $m_{3|||^7}$ (Figs. I.57 and from Figs. I.60 to I.63). These remarks are derived from Figs. I.57 to I.63, which are located in Appendix I.

Through the characterization of ART ANNs, it has been observed that when the feature vectors are more clearly defined, a more robust class is produced. This class robustness translates into having a wider COI as a promising section of learning that can yield high precision classification results.

The method, introduced in this research as parametogram, for finding suitable value candidates for the vigilance parameter ρ needed by both ART1 and FuzzyART is aptly effective to select proper values fitting the mentioned machine learning models. These fitting values can also be grouped into a COI. The more accurate values allow both ART1 and FuzzyART to operate in a regime where they can achieve their best performance. The best performance cases for ART1 have a precision between 97.5% and 98.5%, while the best performance cases for FuzzyART have a precision between 88.5% and 88.7%. Hence, ART1 performs better based on the sensitivity analysis used for the ART based ANN models considered in this research. The high precision and recall achieved by ART1 proves that the multi- and polyscale features used are robust and relevant, and has also managed to avoid the "curse-of-dimensionality" (*i.e.*, the accuracy and generalization reduces as the number of features increase) that other machine learning approaches suffer [DaVS020]. It is important to highlight that ART1 as a machine learning model exceeded the performance expectations crudely set around the 95% vicinity. Another perspective worth noting, since both a high precision and a high recall have been achieved, the dataset used in this research containing 46% of anomalous data with two DDoS attacks (44% for a DNS and 2% for a H&R). This dataset poses a remarkable challenging task for detection as the H&R class of DDoS present has a very small size when compared with the DNS DDoS. Alternate datasets used in the literature contain a single class of DDoS attack as

anomalous data in which the research is based on. With this in view, both the precision and recall achieved with the challenges presented by this dataset are outstanding.

From the FuzzyART parametogram, it is observed that it shows a high-degree of nonlinear dynamics when comparing it to ART1. It is noticed that the learning section of interest in FuzzyART is smaller than ART1.

Based on the work done and presented in this thesis, there are still many important questions related to DDoS. However, two that are very significant are: *Is DDoS a form of cyberattack that is fading away?* and *what makes DDoS to be so prevalent and still proliferating?* Answers to both questions are very challenging. However, in order to tackle these questions from the communications protocols and standards perspective, any usage of DDoS is highly likely related with a form of abuse focused precisely in either protocols or standards. Hence, as long as there are networks that operate based on protocols and standards there is a high potential for a DDoS type to spring up and undermine digital assets based on this critical infrastructure. Based on this argument, known and unseen forms of DDoS would continue to be lurking and hiding in the very underpinnings of the Internet and communications networks: Communications protocols and standards. This line of thinking poses a big question to humanity: *Is there a way to create communications protocols and standards that cannot be abused?*

7.2 Answers to the Research Questions Posed in this Thesis

Upon closure of the research scope proposed in this thesis, answers to the posed research questions are discussed in this subsection. These questions have been extensively covered throughout this thesis. Nevertheless, for clarity, these questions are summarized as follows.

A set of features has been assembled into a feature vector composed of multi- and polyscale based metrics. The raw signal obtained from the Internet traffic has been processed through multiscalors, utilizing the variance and skewness operators. Then the multiscalor components obtained are further pipelined for analysis through secondary operators (cumulative sum, zero crossing rate, and Shannon's entropy) as a form of polyscale analysis. The feature vector obtained has proven to be capable of reflecting the dynamics of the Internet traffic helpful in *detecting DDoS attacks accurately and effectively*.

The feature vectors obtained have been capable of operating as a DL construct, from the

feature extraction perspective, because through the amalgamation of the advanced signal analysis stages a deep representation of the dynamics present in the long-range dependencies is retained. The rich representation of the Internet traffic dynamics contained in the feature vector is then consumed by the *adaptive resonance theory* models, ART1 and FuzzyART. From both machine learning models, ART1 and FuzzyART, it was found that ART1 achieved *a high classification performance*, ranking above the 98% of precision and recall, upon processing of the polyscale feature vector. The performance achieved in this research far exceeds some of latest approaches that consider classification methods (*e.g.*, DDoS detection, based on semantic information about cybersecurity events [SaFF019] and even novel methodologies of ensemble classifiers, majority voting, logistic regression, and naïve Bayes, that have defeated plenty of alternative machine learning approaches [HOHR015]). The approach presented here is faster because it is inherently operating with the Internet traffic dynamics vs secondary levels of information about the cybersecurity event, as is the case of the research presented in the latest approaches (*e.g.*, [SaFF019]). From the machine learning perspective, the early anomalies detection system developed in this thesis recounts the following merits: Effective detection of deviations from normal behaviour; discovery of unknown (due to the lack of identified fingerprints) DDoS attacks, which makes it difficult for attackers to exploit the capabilities of the system. Even if attackers would launch a novel attack, it is highly likely it would not be undetectable for this system; and the precision for the system being over 98% with a similar figure for recall (covering almost the full extent of the DDoS attack) is certainly a success because usually anomaly-based detection systems typically produce a high number of false positives, misclassifications, which undermines both precision and recall.

For the multiscalors, the variance and skewness have been utilized as operators. The variance multiscalor reflects the dynamics of the DNS and H&R DDoS attacks visibly in a more accentuated manner than the skewness multiscalor. The skewness multiscalor has been maintained due to the fact that it can contribute additional representations to the feature factor and therefore a richer pool for *detecting network disruptions induced by DDoS*. By further analysing the multiscalor components through secondary operators, a polyscale aggregation of metrics into a feature vector is achieved. These feature vector has proven to contain robust metrics helpful to characterize DDoS disturbances found in data streams in Internet traffic.

The multi- and polyscale based set of features, integrated throughout the course of this research, has proven the utilization of arbitrary operators for analyzing Internet traffic and collecting dynamics representing long-range dependencies. The feature vector attained reliably reflects the time-multifractality nature of the Internet traffic analyzed. The examination of this point is sustained as evidenced by the high classification performance achieved through ART as the machine learning approach. Nonetheless, alternative machine learning models could also provide a similar classification performance to the one obtained with ART1 because the *detection predictive strength comes from the metrics present in the polyscale feature vector*.

Furthermore and as a restatement, the ten objectives listed in Section 1.3, dataset access, insight development, ITB metrics implementation, isolation of the analysis mechanism from the VFD and VFDT methodologies, primary operators implementation, comparison of the operators merits, feature characterization of Internet traffic, feature vector compilation, ART based ANNs models implementation, and the comparison of the classification precision among machine learning models, has been meticulously followed, documented, and completed.

7.3 Contributions

A new methodology, *multiscalors*, capable of allowing arbitrary operators to be functional in the multiscale domain has been implemented. The use of feature vectors comprised by the characterization of time-multifractality, inherently a property of Internet traffic, has contributed to the high precision and confident classification of the instances in the occurrence of a DDoS attack. The introduction of multiscalors through the course of this research and specific applications for DDoS detection have culminated in intellectual property protection in the form of two patents.

Industry standards in DDoS detection at the time this research was conducted fell in time regimes way above 10 seconds, for inspection of traffic dynamics only and ignoring a priori information of attacking sources, which establishes the *time classification*, one second detection, results achieved through ART in this research is very outstanding and exceptional.

A novel approach, *ART parametogram*, for appropriately characterizing the performance of the vigilance parameter ρ in ART based ANNs, ART1 and FuzzyART, is proposed. The use of the parametogram aids in accurately choosing a vigilance parameter value that can provide a

high precision of performance in DDoS detection through ART by making these ANNs more attentive and focused.

The *anomalies detection system* proposed in this research has been developed at the closure of this research. This detection system has been tested with real Internet traffic containing genuine DDoS attacks. Similarly, the implementation and testing of the empowering backbones in this detection system, the novel employment of multi- and polyscale analysis, for obtaining robust metrics, and implementations of ART ANNs, ART1 and FuzzyART, have been accomplished. This detection system can detect departures, *aka* anomalies, from clear traffic with a high classification precision. The usage of multi- and polyscale analysis has been, with a high degree of confidence, the main contributor to achieving the high level of precision in detection. The study of advanced signal processing is what enabled the main methodology, multiscalors, to be used for obtaining the extraction of relevant features in this research.

7.4 Novelty in the Thesis

Deep multi- and polyscale analysis stages for analysing Internet/network traffic in order to achieve *predictive metrics* that describe unequivocally the dynamics of the traffic. Results from the primary multiscalors operators appeared somewhat noisy. However, upon application of secondary operators for achieving *high compression* and *metrics diversity*, and once these *time-multifractality metrics* are comprised into corresponding feature vectors and fed into the ART implementations, these metrics proved to be highly predictive with a high classification precision.

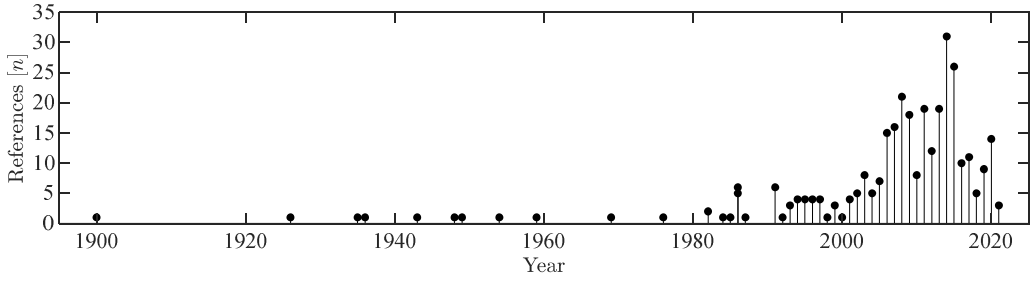
Unsupervised neural networks, based in ART, have been used for the first time in detection of DDoS attacks. Similarly, the fine-tuning of ART based ANNs through a methodology allowing the obtention of a parametogram is original and ground-breaking.

A characterization of the ART1 neural network has been attained and introduced. The more defined or clear the feature vectors are, the more robust the associated class is in the sense that a wider interval of values for the vigilance parameter can be obtained as an operating range helpful in classification.

It has been found that for both ART1 and FuzzyART, sections of learning represented in the parametogram can be described by exponentials, mostly decaying, constant sections, and

cubic polynomials. The coefficient that determines the decay intensity is smaller when the features are less diverse and grows when the features reflect an increment in diversity (appear to be more random).

REFERENCES



[AaAr013] Muhammad Aamir and Muhammad Arif, "Study and performance evaluation on recent DDoS trends of attack & defense," *Int. J. Inf. Technol. Comput. Sci.*, vol. 5, no. 8, pp. 54, 2013, {doi: 10.5815/ijitcs.2013.08.06, ISSN: 2074-9007}.

[AlGo006] Basheer Al-Duwairi and Manimaran Govindarasu, "Novel hybrid schemes employing packet marking and logging for IP traceback," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 5, pp. 403–418, May 2006, {doi: 10.1109/TPDS.2006.63, ISBN: 1045-9219 VO - 17}.

[Alji003] Hassan Aljifri, "IP traceback: A new denial-of-service deterrent?," *IEEE Security & Privacy*, vol. 1, no. 3, pp. 24–31, 2003, {doi: 10.1109/MSECP.2003.1203219, ISBN: 1540-7993 VO - 1}.

[AlSS015] Kazi Masudul Alam, Mukesh Saini, and Abdulmotaleb El Saddik, "Toward social Internet of vehicles: Concept, architecture, and applications," *IEEE Access*, vol. 3, pp. 343–357, February 2015, {doi: 10.1109/ACCESS.2015.2416657, ISBN: 2169-3536 VO - 3}.

[Amin008] Pedram Amini, "Kraken botnet infiltration," *Digital Vaccine Laboratories*, 2008. [Online]. Available: http://dvlabs.tippingpoint.com/blog/2008/04/28/kraken-botnet-infiltration. [Accessed: 21-Nov-2015].

[Amor020] Edward G. Amoroso, *Introduction to Cybersecurity*. New York, NY, USA: New York University, 2020.

[Ande935] Edgar Anderson, "The irises of the Gaspé peninsula," *Bull. Am. Iris Soc.*, vol. 59, pp. 2–5, 1935.

[AnMS982] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of a data-handling system with multiple sources," *Bell Syst. Tech. J.*, vol. 61, no. 8, pp. 1871–1894, October 1982, {doi: 10.1002/j.1538-7305.1982.tb03089.x, ISSN: 0005-8580 VO - 61}.

[ArGu021] John Arquilla and Mark Guzdial, "The SolarWinds Hack, and a Grand Challenge for CS Education," Commun. ACM, vol. 64, no. 4, pp. 6–7, March 2021, {doi: 10.1145/3449047, ISSN: 0001-0782}.

[ASCL006] Paulo E. Ayres, Huizhong Sun, H. Jonathan Chao, and Wing Cheong Lau, "ALPi: A DDoS defense system for high-speed networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 10, pp. 1864–1876, 2006, {doi: 10.1109/JSAC.2006.877136, ISBN: 0733-8716 VO - 24}.

[AsHu010] Sitaram Asur and Bernardo A. Huberman, “Predicting the future with social media,” in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2010 WI-IAT*, (Toronto, ON, Canada; August 2010), 2010, vol. 1, pp. 492–499, {doi: 10.1109/WI-IAT.2010.63}.

[AsLa014] Javed Ashraf and Seemab Latif, “Handling intrusion and DDoS attacks in software defined networks using machine learning techniques,” in *2014 National Software Engineering Conference, 2014 NSEC*, (Rawalpindi, Pakistan; November 2014), 2014, pp. 55–60, {doi: 10.1109/NSEC.2014.6998241}.

[AtIM014] Luigi Atzori, Antonio Iera, and Giacomo Morabito, “From ‘smart objects’ to ‘social objects’: The next evolutionary step of the Internet of Things,” *IEEE Communications Magazine*, vol. 52, no. 1, pp. 97–105, Jan-2014, {doi: 10.1109/MCOM.2014.6710070, ISBN: 0163-6804 VO - 52}.

[Axel000] S. Axelsson, “Instrusion Detection Systems: A Survey and Taxonomy,” Department of Computer Engineering, Chalmers University of Technology, 2000.

[AZZS019] A. M. Aleesa, B. B. Zaidan, A. A. Zaidan, and Nan. M. Sahar, “Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions,” *Neural Comput. Appl.*, pp. 32, October 2019, {doi: 10.1007/s00521-019-04557-3, ISSN: 1433-3058}.

[BaAZ014] Eray Balkanli, Jander Alves, and A. Nur Zincir-Heywood, “Supervised learning to detect DDoS attacks,” in *2014 IEEE Symposium on Computational Intelligence in Cyber Security, 2014 CICS*, (Orlando, FL, USA; December 2014), 2014, pp. 1–8, {doi: 10.1109/CICYBS.2014.7013367}.

[Barr012] John Barret, “The Internet of Things,” *TEDx CIT*, 2012. [Online]. Available: http://tedxtalks.ted.com/video/The-Internet-of-Things-Dr-John;search:internet of things barret. [Accessed: 29-Jul-2015].

[BAUM014] Mehdi Barati, Azizol Abdullah, Nur Izura Udzir, Ramlan Mahmod, and Norwati Mustapha, “Distributed denial of service detection using hybrid machine learning technique,” in *2014 International Symposium on Biometrics and Security Technologies, 2014 ISBAST*, (Kuala Lumpur, Malaysia; August 2014), 2014, pp. 268–273, {doi: 10.1109/ISBAST.2014.7013133}.

[BBMW014] Johannes Braun, Johannes Buchmann, Ciaran Mullan, and Alex Wiesmaier, “Long-term confidentiality: A survey,” *Des. Codes Cryptogr.*, vol. 71, no. 3, pp. 459–478, 2014, {doi: 10.1007/s10623-012-9747-6, ISSN: 0925-1022}.

[BCFP999] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Shenker Shenker, “Web caching and Zipf-like distributions: Evidence and implications,” in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies, 1999 IEEE/INFOCOM*, 1999, vol. 1, pp. 126–134 vol.1, {doi: 10.1109/INFCOM.1999.749260, ISBN: 0743-166X VO - 1}.

[BCJX009] Michael Bailey, Evan Cooke, Farnam Jahanian, Yunjing Xu, and Manish Karir, “A survey of botnet technology and defenses,” in *Cybersecurity Applications & Technology Conference for Homeland Security, 2009 CATCH*, 2009, pp. 299–304, {doi: 10.1109/CATCH.2009.40}.

[BeCu007] Yoshua Bengio and Yann Le Cun, “Scaling Learning Algorithms Towards AI,” in *Large-Scale Kernel Machines*, Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, Eds. Massachusetts, MA, USA: MIT Press, 2007, pp. 408, {ISBN: 978-0262026253}.

[BeDe014] Hakem Beitollahi and Geert Deconinck, “ConnectionScore: A statistical technique to resist application-layer DDoS attacks,” *J. Ambient Intell. Humaniz. Comput.*, vol. 5, no. 3, pp. 425–442, 2014, {doi: 10.1007/s12652-013-0196-5, ISSN: 1868-5137}.

[BeSF994] Yoshua Bengio, Patrice Simard, and Paolo Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157–166, March 1994, {doi: 10.1109/72.279181, ISSN: 1045-9227 VO - 5}.

[BhBK015] Monowar H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection,” *Pattern Recognit. Lett.*, vol. 51, pp. 1–7, January 2015, {doi: http://dx.doi.org/10.1016/j.patrec.2014.07.019, ISSN: 0167-8655}.

[Bick019] Ian Bickis, “TransUnion breach shows rising third-party cyberattack threat in Canada - BNN Bloomberg,” *BNN Bloomberg | The Canadian Press*, 2019. [Online]. Available: https://www.bnnbloomberg.ca/transunion-breach-shows-rising-third-party-cyberattack-threat-in-canada-1.1329808. [Accessed: 16-Aug-2020].

[BKBK014] Monowar H. Bhuyan, H. J. Kashyap, D. K. Bhattacharyya, and J. K. Kalita, “Detecting distributed denial of service attacks: Methods, tools and future directions,” *Comput. J.*, vol. 57, no. 4, pp. 537–556, 2014, {doi: 10.1093/comjnl/bxt031, ISSN: 0010-4620}.

[BKPR002] Paul Barford, Jeffery Kline, David Plonka, and Amos Ron, “A signal analysis of network traffic anomalies,” in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement*, (New York, NY, USA;2002), 2002, pp. 71–82, {doi: 10.1145/637201.637210, ISBN: 1-58113-603-X}.

[Blak012] Oskar Blakstad, “Scientific method,” *Explorable*, 2012. [Online]. Available: https://explorable.com/scientific-method. [Accessed: 23-Dec-2014].

[BLNS020] Jeremy Bryans, Lin Shen Liew, Hoang Nga Nguyen, Giedre Sabaliauskaite, Siraj Shaikh, Fengjun Zhou, Maryline Laurent, and Thanassis Giannetsos, “A Template-Based Method for the Generation of Attack Trees,” in *Information Security Theory and Practice: Proceedings of the 13th IFIP WG 11.2 International Conference, WISTP 2019*, vol. 12024, Paris, France: Springer International Publishing, 2020, pp. 155–165, {doi: 10.1007/978-3-030-41702-4_10, ISBN: 0302-9743}.

[Blow015] Misty Blowers, Editor, *Evolution of Cyber* Technologies *and Operations to 2035*, vol. 63. Rome, NY, USA: Springer International Publishing Switzerland, 2015, 201 pp., {doi: 10.1007/978-3-319-23585-1}.

[BLPL006] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle, “Greedy layer-wise training of deep networks,” in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, (Cambridge, MA, USA; December 2006), 2006, pp. 153–160.

[BMZA012] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli, “Fog computing and its role in the Internet of things,” in *Proceedings of the MCC workshop on Mobile*

Cloud Computing, (Helsinki, Finland; August 2012), 2012, pp. 13–16, {doi: 10.1145/2342509.2342513}.

[BoAy013] Ali Sharifi Boroujerdi and Saeed Ayat, "A robust ensemble of neuro-fuzzy classifiers for DDoS attack detection," in *2013 3rd International Conference on Computer Science and Network Technology, 2013 ICCSNT*, (Dalian, China; October 2013), 2013, pp. 484–487, {doi: 10.1109/ICCSNT.2013.6967159}.

[BrKu019] Steven L. Brunton and J. Nathan Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. New York, NY, USA: Cambridge University Press, 2019, 495 pp., {ISBN: 978-1-108-42209-3}.

[BrLe005] Anat Bremler-Barr and Hanoch Levy, "Spoofing prevention method," in *2005 24th Annual Joint Conference of the IEEE/INFOCOM Computer and Communications Societies*, 2005, vol. 1, pp. 536–547 vol. 1, {doi: 10.1109/INFCOM.2005.1497921, ISBN: 0743-166X VO - 1}.

[BSMT014] Sajal Bhatia, Desmond Schmidt, George Mohay, and Alan Tickle, "A framework for generating realistic traffic for distributed denial-of-service attacks and flash events," *Comput. Secur.*, vol. 40, pp. 95, 2014, {ISSN: 01674048}.

[BuGu016] Anna L. Buczak and Erhan Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016, {doi: 10.1109/COMST.2015.2494502, ISSN: 1553-877X}.

[Bull014] Scott R. Bullock, *Cognitive Systems*, 4th ed. Edison, NJ, USA: Scitech Publishing-Institution of Engineering and Technology, 2014, 369 pp., {ISBN: 9781613532034}.

[Butt015] Stefano Buttiglione, "Internet of [insecure] Things," *TEDx Bergamo*, 2015. [Online]. Available: http://tedxtalks.ted.com/video/Internet-of-insecure-Things-Ste;search:intern of things. [Accessed: 02-Aug-1983].

[BiVS013] Saugata S. Biswas, Ceeman B. Vellaithurai, and Anurag K. Srivastava, "Development and real time implementation of a synchrophasor based fast voltage stability monitoring algorithm with consideration of load models," in *2013 IEEE Industry Applications Society Annual Meeting*, 2013, pp. 1–9, {doi: 10.1109/IAS.2013.6682584, ISSN: 0197-2618}.

[Cai011] Yang Cai, Editor, "Sound Recognition," in *Computing with Instinct: Rediscovering Artificial Intelligence*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 173, {doi: 10.1007/978-3-642-19757-4_2, ISBN: 978-3-642-19757-4}.

[Call019] Spencer Callaghan, "Survey finds 71 per cent of Canadian organizations impacted by a cyber-attack last year," *Canadian Internet Registration Authority (CIRA)*, 2019. [Online]. Available: https://www.cira.ca/newsroom/cybersecurity/survey-finds-71-cent-canadian-organizations-impacted-a-cyber-attack-last. [Accessed: 16-Aug-2020].

[Cart015] Hugh M. Cartwright, Editor, *Artificial Neural Networks*, 2nd ed. New York, NY, USA: Springer New York, 2015, 344 pp.

[Cass016] Christos G. Cassandras, "Smart Cities as Cyber-Physical Social Systems," *Engineering*, vol. 2, no. 2, pp. 156–158, June 2016, {doi: 10.1016/J.ENG.2016.02.012, ISSN: 2095-8099}.

[Caud012] Jo Caudron, “Openness and security on the Internet,” *TEDx UHowest*, 2012. [Online]. Available: http://tedxtalks.ted.com/video/TEDxUHowest-Jo-Caudron-Openness;search:intern of things. [Accessed: 06-Aug-2015].

[CBKD014] Armando W. Colombo, Thomas Bangemann, Stamatis Karnouskos, Jerker Delsing, Petr Stluka, and Robert Harrison, Editors, *Industrial Cloud-Based Cyber-Physical Systems: The IMC-AESOP Approach*. Cham, Switzerland: Springer International Publishing, 2014, 261 pp., {doi: 10.1007/978-3-319-05624-1}.

[CCGP010] Chia Yuan Cho, Juan Caballero, Chris Grier, Vern Paxson, and Dawn Song, “Insights from the inside: A view of botnet management from infiltration,” *Proc. 3rd Conf. Largescale Exploit. Emergent Threat. Botnets, Spyware, Worms, More, USENIX*, 2010.

[CFBG006] J. M. Carrasco, L. G. Franquelo, J. T. Bialasiewicz, E. Galvan, R. C. PortilloGuisado, M. A. M. Prats, J. I. Leon, and N. Moreno-Alfonso, “Power-Electronic Systems for the Grid Integration of Renewable Energy Sources: A Survey,” *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1002–1016, June 2006, {doi: 10.1109/TIE.2006.878356, ISSN: 1557-9948}.

[ChHw006] Yu Chen and Kai Hwang, “Collaborative detection and filtering of shrew DDoS attacks using spectral analysis,” *J. Parallel Distrib. Comput.*, vol. 66, no. 9, pp. 1137–1151, September 2006, {doi: http://dx.doi.org/10.1016/j.jpdc.2006.04.007, ISSN: 0743-7315}.

[ChJi009] Zesheng Chen and Chuanyi Ji, “An information-theoretic view of network-aware malware attacks,” in *IEEE Transactions on Information Forensics and Security*, 2009, vol. 4, no. 3, pp. 530–541, {doi: 10.1109/TIFS.2009.2025847, ISBN: 1556-6013 VO - 4}.

[ChKT002] Chen-Mou Cheng, H. T. Kung, and Koan-Sin Tan, “Use of spectral analysis in defense against DoS attacks,” in *2002 IEEE Global Telecommunications Conference, 2002 GLOBECOM*, 2002, vol. 3, pp. 2143–2148 vol.3, {doi: 10.1109/GLOCOM.2002.1189011}.

[ChMW013] Yonghong Chen, Xinlei Ma, and Xinya Wu, “DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory,” *IEEE Commun. Lett.*, vol. 17, no. 5, pp. 1052–1054, May 2013, {doi: 10.1109/LCOMM.2013.031913.130066, ISBN: 1089-7798 VO - 17}.

[ChPB013] S. S. Chapade, K. U. Pandey, and D. S. Bhade, “Securing Cloud servers against flooding based DDOS attacks,” in *2013 International Conference on Communication Systems and Network Technologies, CSNT 2013*, 2013, pp. 524–528, {doi: 10.1109/CSNT.2013.114}.

[ChPM007] Ruiliang Chen, Jung-Min Park, and Randolph Marchany, “A divide-and-conquer strategy for thwarting distributed denial-of-service attacks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 5, pp. 577–588, May 2007, {doi: 10.1109/TPDS.2007.1014, ISBN: 1045-9219 VO - 18}.

[ChZa005] Sen-ching S. Cheung and Avideh Zakhor, “Fast similarity search and clustering of video sequences on the world-wide-web,” *IEEE Trans. Multimed.*, vol. 7, no. 3, pp. 524–537, 2005, {doi: 10.1109/TMM.2005.846906, ISBN: 1520-9210 VO - 7}.

[ClSc004] Christopher R. Clark and David E. Schimmel, “Scalable pattern matching for high speed networks,” in *2004 12th Annual IEEE Symposium on Field-Programmable Custom*

Computing Machines, 2004 FCCM, 2004, pp. 249–257, {doi: 10.1109/FCCM.2004.50, ISBN: VO -}.

[CKBR006] Glenn Carl, George Kesidis, Richard R. Brooks, and Suresh Rai, "Denial-of-service attack-detection techniques," *IEEE Internet Computing*, vol. 10, no. 1, pp. 82–89, 2006, {doi: 10.1109/MIC.2006.5, ISBN: 1089-7801 VO - 10}.

[Corm014] Joshua Corman, "Swimming with sharks - Security in the Internet of Things," *TEDx Naperville*, 2014. [Online]. Available: http://tedxtalks.ted.com/video/Swimming-with-sharks-security-i;search:intern of things. [Accessed: 05-Aug-2015].

[CoTh005] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons Inc., 2005, 774 pp., {ISBN: 9780471241959}.

[CPFS014] Maximo Cobos, Juan J. Perez-Solano, Santiago Felici-Castell, Jaume Segura, and Juan M. Navarro, "Cumulative-Sum-Based Localization of Sound Events in Low-Cost Wireless Acoustic Sensor Networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 1792–1802, 2014, {doi: 10.1109/TASLP.2014.2351132, ISSN: 2329-9290 VO - 22}.

[Cran019] Casey Crane, "The largest DDoS Attack in history," Hashedout, 2019. [Online]. Available: https://www.thesslstore.com/blog/largest-ddos-attack-in-history/. [Accessed: 20-Apr-2020].

[Cran020] Casey Crane, "The 15 Top DDoS statistics you should know In 2020," Cybercrime Magazine, 2020. [Online]. Available: https://cybersecurityventures.com/the-15-top-ddos-statistics-you-should-know-in-2020/. [Accessed: 20-Apr-2020].

[CrBe997] Mark E. Crovella and Azer Bestavros, "Self-similarity in world wide web traffic: Evidence and possible causes," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 835–846, 1997.

[CRDB011] P. Crolla, A. J. Roscoe, A. Dyśko, and G. M. Burt, "Methodology for testing loss of mains detection algorithms for microgrids and distributed generation using real-time power hardware-in-the-loop based technique," in *8th International Conference on Power Electronics - ECCE Asia*, 2011, pp. 833–838, {doi: 10.1109/ICPE.2011.5944703, ISSN: 2150-6086}.

[CTDB009] Alper Caglayan, Mike Toothaker, Dan Drapeau, Dustin Burke, and Gerry Eaton, "Real-time detection of fast flux service networks," in *Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security*, (Washington, DC, USA;2009), 2009, pp. 285–292, {doi: 10.1109/CATCH.2009.44, ISBN: 978-0-7695-3568-5}.

[Cun987] Yann Le Cun, "Modèles Connexionnistes de l'Apprentissage," Université Pierre et Marie Curie, Paris, France, 1987.

[CuRT014] Hermann Cuntz, Michiel W. H. Remme, and Benjamin Torben-Nielsen, Editors, *The Computing Dendrite From Structure to Function*. Dordrecht, The Netherlands: Springer, 2014, 519 pp.

[Cybe020] Cybersecurity and Infrastructure Security Agency, "Critical infrastructure sectors," 2020. [Online]. Available: http://www.dhs.gov/critical-infrastructure-sectors. [Accessed: 09-Jun-2020].

[DaTh015] Jisa David and Ciza Thomas, "DDoS attack detection using fast entropy approach on flow-based network traffic," *Procedia Comput. Sci.*, vol. 50, pp. 30–36, 2015, {doi: http://dx.doi.org/10.1016/j.procs.2015.04.007, ISSN: 1877-0509}.

[Davi015] Amanda Davis, “A history of hacking,” *IEEE The Institute*, 2015. [Online]. Available: http://theinstitute.ieee.org/technology-focus/technology-history/a-history-of-hacking.

[DaVS020] Saikat Das, Deepak Venugopal, and Sajjan Shiva, “A Holistic Approach for Detecting DDoS Attacks by Using Ensemble Unsupervised Machine Learning,” in *Advances in Information and Communication: Proceedings of the 2020 Future of Information and Communication Conference (FICC), Volume 2*, vol. 1130, Kohei Arai, Supriya Kapoor, and Rahul Bhatia, Eds. Cham, Switzerland: Springer International Publishing, 2020, pp. 721–738, {doi: 10.1007/978-3-030-39442-4_53, ISBN: 2194-5357}.

[DaZL006] David Dagon, Cliff Changchun Zou, and Wenke Lee, “Modeling botnet propagation using time zones,” *Proc. 13 th Netw. Distrib. Syst. Secur. Symp. NDSS*, vol. 6, pp. 2–13, 2006, {doi: 10.1.1.128.8689}.

[DDHT008] R. Dobrescu, M. Dobrescu, D. Hossu, and S. Taralunga, “Using internet traffic self-similarity for detection of network anomalies,” in *2008 11th International Conference on Optimization of Electrical and Electronic Equipment, 2008 OPTIM*, 2008, pp. 81–86, {doi: 10.1109/OPTIM.2008.4602461}.

[DeBe011] Olivier Delalleau and Yoshua Bengio, “Shallow vs. deep sum-product networks,” in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, (Granada, Spain; December 2011), 2011, pp. 666–674, {ISBN: 978-1-61839-599-3}.

[DeDW999] H. Debar, M. Dacier, and A. Wespi, *A revised taxonomy for intrusion-detection systems*. Zurich, Switzerland: IBM Research, 1999.

[DeFS002] Drew Dean, Matt Franklin, and Adam Stubblefield, “An algebraic approach to IP traceback,” *ACM Trans. Inf. Syst. Secur.*, vol. 5, no. 2, pp. 119–137, May 2002, {doi: 10.1145/505586.505588, ISSN: 1094-9224}.

[DeLD009] Pradip De, Yonghe Liu, and Sajal K. Das, “An epidemic theoretic framework for vulnerability analysis of broadcast protocols in wireless sensor networks,” *IEEE Trans. Mob. Comput.*, vol. 8, no. 3, pp. 413–425, March 2009, {doi: 10.1109/TMC.2008.115, ISBN: 1536-1233 VO - 8}.

[Depa013] Department of Homeland Security (DHS) of the United States of America: PREDICT Project, “Scrambled Internet trace measurement dataset - PREDICT ID: USC-LANDER/DoS_DNS_amplification-20130617/rev5037,” *USC/LANDER Project*, 2013. [Online]. Available: http://www.isi.edu/ant/lander. {doi: 10.23721/109/1353940} [Accessed: 02-Nov-2015].

[Desc020] Tara Deschamps, “Chinese military members face charges in Equifax breach impacting Canadians,” *CTV News*, 2020. [Online]. Available: https://www.ctvnews.ca/business/chinese-military-members-face-charges-in-equifax-breach-impacting-canadians-1.4805070. [Accessed: 16-Aug-2020].

[Doan976] David P. Doane, “Aesthetic frequency classifications,” *Am. Stat.*, vol. 30, no. 4, pp. 181–183, November 1976, {doi: 10.2307/2683757, ISSN: 00031305}.

[Dono995] David L. Donoho, “De-noising by soft-thresholding,” *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, 1995, {doi: 10.1109/18.382009, ISSN: 0018-9448 VO - 41}.

[DoSe011] David P. Doane and Lori E. Seward, “Measuring Skewness: A forgotten statistic?,” *J. Stat. Educ.*, vol. 19, no. 2, 2011, {ISSN: 1069-1898}.

[Duff004] Nick Duffield, “Sampling for passive Internet measurement: A review,” *Stat. Sci.*, vol. 19, no. 3, pp. 472–498, 2004, {doi: 10.1214/088342304000000206, ISBN: 08834237, ISSN: 0883-4237}.

[DuHS001] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley Publishing Inc., 2001, 654 pp.

[DuYC008] Zhenhai Duan, Xin Yuan, and Jaideep Chandrashekar, “Controlling IP spoofing through interdomain packet filters,” *IEEE Trans. Dependable Secur. Comput.*, vol. 5, no. 1, pp. 22–36, 2008, {doi: 10.1109/TDSC.2007.70224, ISBN: 1545-5971 VO - 5}.

[EATB009] Adel El-Atawy, Ehab Al-Shaer, Tung Tran, and Raouf Boutaba, “Adaptive early packet filtering for defending firewalls against DoS attacks,” in *2009 IEEE INFOCOM*, 2009, pp. 2437–2445, {doi: 10.1109/INFCOM.2009.5062171, ISBN: 0743-166X VO -}.

[ErRé959] P. Erdös and A. Rényi, “On random graphs, I,” *Publ. Math.*, vol. 6, pp. 290–297, 1959.

[Fard015] Ashkan Fardost, “Internet of Things - Beyond our current imagination,” *TEDx Östersund*, 2015. [Online]. Available: http://tedxtalks.ted.com/video/Internet-of-things-beyond-our-c;search:intern of things. [Accessed: 04-Aug-2015].

[Fars013] Rashida Farsath.K, “Virtual ring network protection against flooding DDOS attack,” *Int. J. Comput. Trends Technol.*, vol. 4, no. 8, pp. 2889, 2013, {ISSN: 2231-2803}.

[FeSR009] Maryam Feily, Alireza Shahrestani, and Sureswaran Ramadass, “A survey of botnet and botnet detection,” in *2009 3rd International Conference on Emerging Security Information, Systems and Technologies, 2009 SECURWARE*, 2009, pp. 268–273, {doi: 10.1109/SECURWARE.2009.48, ISBN: VO -}.

[Fish936] Ronald A. Fisher, “The use of multiple measurements in taxonomic problems,” *Ann. Eugen.*, vol. 7, no. 2, pp. 179–188, September 1936, {doi: 10.1111/j.1469-1809.1936.tb02137.x, ISSN: 2050-1420}.

[FrSc997] Yoav Freund and Robert E Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997, {doi: 10.1006/jcss.1997.1504, ISSN: 0022-0000}.

[FrSk991] James A. Freeman and David M. Skapura, *Neural Networks: Algorithms, Applications, and Programming Techniques*. Reading, MA, USA: Addison-Wesley Publishing Company, 1991, 401 pp., {ISBN: 0-201-51376-5}.

[FSBK003] Laura Feinstein, Dan Schnackenberg, Ravindra Balupari, and Darrell Kindred, “Statistical approaches to DDoS attack detection and response,” in *Proceedings of the 2003 DARPA Information Survivability Conference and Exposition*, 2003, vol. 1, pp. 303–314 vol.1, {doi: 10.1109/DISCEX.2003.1194894}.

[Fu011] Zhang Fu, “Mitigating distributed denial-of-service attacks: Application-defense and network-defense methods,” in *2011 7th European Conference on Computer Network Defense, 2011 EC2ND*, 2011, pp. 59, {doi: 10.1109/EC2ND.2011.18}.

[Gars900] J. G. Garson, “The metric system of identification of criminals, as used in Great Britain and Ireland,” *J. Anthropol. Inst. Gt. Britain Irel.*, vol. 30, pp. 161–198, 1900, {doi: 10.2307/2842627, ISSN: 09595295}.

[GFBH996] Michael Georgiopoulos, Hans Fernlund, George Bebis, and Gregory L. Heileman, “Order of search in Fuzzy ART and Fuzzy ARTMAP: Effect of the choice parameter,”

Neural Networks, vol. 9, no. 9, pp. 1541–1559, December 1996, {doi: 10.1016/S0893-6080(96)00018-4, ISSN: 0893-6080}.

[Glig984] Virgil D. Gligor, "A note on denial-of-service in operating systems," in *IEEE Transactions on Software Engineering*, 1984, vol. SE-10, no. 3, pp. 320–324, {doi: 10.1109/TSE.1984.5010241, ISBN: 0098-5589 VO - SE-10}.

[GoBC016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*. Massachusetts, MA, USA: MIT Press, 2016, 781 pp.

[Good008] Michael T. Goodrich, "Probabilistic packet marking for large-scale IP traceback," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 15–24, February 2008, {doi: 10.1109/TNET.2007.910594, ISBN: 1063-6692 VO - 16}.

[Goss008] William S. Gosset, "The probable error of a mean," *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908, {doi: 10.2307/2331554, ISSN: 00063444}.

[GPYF007] Guofei Gu, Phillip Porras, Vinod Yegneswaran, Martin Fong, and Wenke Lee, "BotHunter: Detecting malware infection through IDS-driven dialog correlation," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, (Berkeley, CA, USA; August 2007), 2007, pp. 12:1–12:16, {ISBN: 111-333-5555-77-9}.

[GPZL008] Guofei Gu, Roberto Perdisci, Junjie Zhang, and Wenke Lee, "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proceedings of the 17th Conference on Security Symposium*, (Berkeley, CA, USA; July 2008), 2008, pp. 139–154.

[Grau013] Daniel Graupe, *Principles of Artificial Neural Networks*, 3rd ed. Singapore, Singapore: World Scientific, 2013, 363 pp., {ISBN: 978-981-4522-73-1}.

[Gree021] Samuel Greengard, "The Worsening State of Ransomware," Commun. ACM, vol. 64, no. 4, pp. 15–17, March 2021, {doi: 10.1145/3449054, ISSN: 0001-0782}.

[GuYS012] Xizheng Guo, Xiaojie You, and Yumei Song, "Real-time digital simulation of high-power electrical traction system," in *2012 15th International Power Electronics and Motion Control Conference (EPE/PEMC)*, (Novi Sad, Serbia; September 2012), 2012, pp. LS4a.1-1-LS4a.1-5, {doi: 10.1109/EPEPEMC.2012.6397445}.

[GuYZ015] Bin Guo, Zhiwen Yu, and Xingshe Zhou, "A data-centric framework for cyber-physical-social systems," *IT Prof.*, vol. 17, no. 6, pp. 4–7, November 2015, {doi: 10.1109/MITP.2015.116, ISSN: 1520-9202 VO - 17}.

[Hagi013] Jeff Hagins, "The Internet of Things," *TEDx SF*, 2013. [Online]. Available: http://tedxtalks.ted.com/video/The-Internet-of-Things-Jeff-Hag;search:intern of things.

[Hebb005] Donald O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. New York, NY, USA: Taylor & Francis, 2005, 378 pp., {doi: 10.4324/9781410612403, ISBN: 9780805843002}.

[Hell018] Michael Heller, "Terabit DDoS attack hits 1.7Tbps and experts expect higher," *TechTarget SearchSecurity*, 2018. [Online]. Available: https://searchsecurity.techtarget.com/news/252436340/Terabit-DDoS-attack-hits-17Tbps-and-experts-expect-higher. [Accessed: 17-Apr-2018].

[HeLu986] Harry Heffes and David Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," *IEEE J. Sel. Areas Commun.*, vol. 4, no. 6, pp. 856–868, September 1986, {doi: 10.1109/JSAC.1986.1146393, ISSN: 0733-8716 VO - 4}.

[Hens016] Michael J. de C Henshaw, “Systems of systems, cyber-physical systems, the Internet-of-things… Whatever next?,” *INSIGHT*, vol. 19, no. 3, pp. 51–54, October 2016, {doi: 10.1002/inst.12109, ISSN: 2156-4868}.

[HeZH011] Ran He, Wei-Shi Zheng, and Bao-Gang Hu, “Maximum correntropy criterion for robust face recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1561–1576, 2011, {doi: 10.1109/TPAMI.2010.220, ISBN: 0162-8828 VO - 33}.

[HGRF008] Thorsten Holz, Christian Gorecki, Konrad Rieck, and Felix C. Freiling, “Measuring and detecting fast-flux service networks,” in *2008 3rd International Conference on Malicious and Unwanted Software, 2008 MALWARE*, 2008, pp. 24 – 31, {doi: 10.1.1.140.188, ISSN: 13669516}.

[HiSe986] Geoffrey E. Hinton and Terrence J. Sejnowski, “Learning and Relearning in Boltzmann Machines,” in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition - Volume 1: Foundations*, David E. Rumelhart and James L. McClelland, Eds. Cambridge, MA, USA: MIT Press, 1986, pp. 282–317, {ISBN: 0-262-68053-X}.

[HiSh991] Geoffrey E. Hinton and Tim Shallice, “Lesioning an attractor network: Investigations of acquired dyslexia,” *Psychol. Rev.*, vol. 98, no. 1, pp. 74–95, January 1991, {ISSN: 0033-295X}.

[HiOT006] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, July 2006, {doi: 10.1162/neco.2006.18.7.1527, ISSN: 0899-7667}.

[HNGH007] Ling Huang, XuanLong Nguyen, Minos Garofalakis, Joseph M. Hellerstein, Michael I. Jordan, Anthony D. Joseph, and Nina Taft, “Communication-efficient online detection of network-wide anomalies,” in *2007 26th IEEE International Conference on Computer Communications, 2007 IEEE/INFOCOM*, 2007, pp. 134–142, {doi: 10.1109/INFCOM.2007.24, ISBN: 0743-166X VO -}.

[Hoch991] Sepp Hochreiter, “Untersuchungen zu Dynamischen Neuronalen Netzen,” Technische Universität München, Munich, Germany, 1991.

[HOHR015] Nutan Farah Haq, Abdur Rahman Onik, Md. Avishek Khan Hridoy, Musharrat Rafni, Faisal Muhammad Shah, and Dewan Md. Farid, “Application of machine learning approaches in intrusion detection system: A survey,” *Int. J. Adv. Res. Artif. Intell.*, vol. 4, no. 3, pp. 9–18, 2015, {doi: 10.14569/IJARAI.2015.040302}.

[Houg015] Benson Hougland, “What is the Internet of Things? And why should you care?,” *TEDx Temecula*, 2015. [Online]. Available: http://tedxtalks.ted.com/video/What-is-the-Internet-of-Things;search:Internet of things. [Accessed: 30-Jul-2015].

[HXCL014] Song Han, Miao Xie, Hsiao-Hwa Chen, and Yun Ling, “Intrusion Detection in Cyber-Physical Systems: Techniques and Challenges,” *IEEE Syst. J.*, vol. 8, no. 4, pp. 1052–1062, December 2014, {doi: 10.1109/JSYST.2013.2257594, ISSN: 1937-9234}.

[IaHa007] Nicolas Ianelli and Aaron Hackworth, “Botnets as a vehicle for online crime,” *Int. J. Forensic Comput. Sci.*, pp. 19–39, 2007, {doi: 10.5769/J200701002, ISSN: 18099807}.

[IdTJ013] Joseph Idziorek, Mark F. Tannian, and Doug Jacobson, “The insecurity of Cloud utility models,” *IT Professional, IT Pro*, vol. 15, no. 2, pp. 22–27, 2013, {doi: 10.1109/MITP.2012.43, ISBN: 1520-9202 VO - 15}.

[JaBM013] Madiha Jalil, Faran Awais Butt, and Ahmed Malik, “Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals,” in *2013 International Conference on Technological Advances in Electrical, Electronics and Computer Engineering, 2013 TAEECE*, (Konya, Turkey; May 2013), 2013, pp. 208–212, {doi: 10.1109/TAEECE.2013.6557272}.

[JaRo986] Raj Jain and Shawn Routhier, “Packet trains - Measurements and a new model for computer network traffic,” *IEEE J. Sel. Areas Commun.*, vol. 4, no. 6, pp. 986–995, September 1986, {doi: 10.1109/JSAC.1986.1146410, ISSN: 0733-8716 VO - 4}.

[Java018] Javapipe, “35 types of DDoS attacks,” *Javapipe*, 2018. [Online]. Available: https://javapipe.com/ddos/blog/ddos-types/. [Accessed: 1-Apr-2018].

[JCJL010] Nan Jiang, Jin Cao, Yu Jin, Li Li, and Zhi-Li Zhang, “Identifying suspicious activities through DNS failure graph analysis,” in *2010 18th IEEE International Conference on Network Protocols, 2010 ICNP*, 2010, pp. 144–153, {doi: 10.1109/ICNP.2010.5762763, ISBN: 1092-1648 VO -}.

[JFKY013] Zhu Jian-Qi, Fu Feng, Yin Ke-xin, and Liu Yan-Heng, “Dynamic entropy based DoS attack detection method,” *Comput. Electr. Eng.*, vol. 39, no. 7, pp. 2243–2251, October 2013, {doi: http://dx.doi.org/10.1016/j.compeleceng.2013.05.003, ISSN: 0045-7906}.

[JiHT017] Yuxuan Jiang, Zhe Huang, and Danny H. K. Tsang, “Challenges and solutions in fog computing orchestration,” *IEEE Netw.*, no. 99, pp. 1–8, November 2017, {doi: 10.1109/MNET.2017.1700271, ISSN: 0890-8044 VO - PP}.

[JuKR002] Jaeyeon Jung, Balachander Krishnamurthy, and Michael Rabinovich, “Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites,” in *Proceedings of the 11th International Conference on World Wide Web*, (New York, NY, USA;2002), 2002, pp. 293–304, {doi: 10.1145/511446.511485, ISBN: 1-58113-449-5}.

[KaCV014] L. Kavisankar, C. Chellappan, and R. Vaishnavi, “Network layer DDoS mitigation model using hidden semi-Markov model,” *Int. J. e-Education*, vol. 4, no. 1, pp. 42, 2014, {doi: 10.7763/IJEEEE.2014.V4.299, ISSN: 2010-3654}.

[KaSc004] Holger Kantz and Thomas Schreiber, *Nonlinear Time Series Analsis*, 2nd ed. New York, NY, USA: Cambridge University Press, 2004, 388 pp., {ISBN: 0521821509}.

[Kasp014] Kaspersky Lab, *Global IT Security Risks Survey 2014 – Distributed Denial of Service (DDoS) Attacks*. Moscow, Russia: Kaspersky Lab, 2014, 14 pp.

[KCCP003] Witold Kinsner, Vincent Cheung, Kevin Cannons, Joseph Pear, and Toby Martin, “Signal classification through multifractal analysis and complex domain neural networks,” in *2003 2nd IEEE International Conference on Cognitive Informatics, 2003 ICCI*, 2003, pp. 41–46, {doi: 10.1109/COGINF.2003.1225951}.

[KCCP006] Witold Kinsner, Vincent Cheung, Kevin Cannons, Joseph Pear, and Toby Martin, “Signal classification through multifractal analysis and complex domain neural networks,” in *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2006, vol. 36, no. 2, pp. 196–203, {doi: 10.1109/TSMCC.2006.871148, ISBN: 1094-6977 VO - 36}.

[KhFK015a] Muhammad Salman Khan, Ken Ferens, and Witold Kinsner, “A cognitive multifractal approach to characterize complexity of non-stationary and malicious DNS data traffic using adaptive sliding window,” in *2015 IEEE 14th International Conference*

*on Cognitive Informatics & Cognitive Computing, 2015 ICCI*CC*, (Beijing, China; July 2015), 2015, pp. 76–83, {doi: 10.1109/ICCI-CC.2015.7259368, ISBN: VO -}.

[KhFK015b] Muhammad Salman Khan, Ken Ferens, and Witold Kinsner, "A polyscale autonomous sliding window for cognitive machine classification of malicious internet traffic," in *The 2015 International Conference on Security and Management, 2015 SAM*, (Las Vegas, NV, USA; July 2015), 2015, pp. 96–103.

[KiGr008] Witold Kinsner and Warren Grieder, "Speech segmentation using multifractal measures and amplification of signal features," in *2008 7th IEEE International Conference on Cognitive Informatics, 2008 ICCI*, (Stanford, CA, USA; August 2008), 2008, pp. 351–357, {doi: 10.1109/COGINF.2008.4639188}.

[KiGr010] Witold Kinsner and Warren Grieder, "Amplification of signal features using variance fractal dimension trajectory," *Int. J. Cogn. Informatics Nat. Intell.*, vol. 4, no. 4, pp. 1–17, 2010.

[Kins994] Witold Kinsner, "Batch and Real-Time Computation of the Variance Fractal Dimension (VFD) and Variance Fractal Dimension Trajectory (VFDT) of a Time Series," in *2016 Fractal and Chaos Engineering: Lecture Notes. Winnipeg, MB, Canada: University of Manitoba; Ch. 2, Appendix 2B*. May-1994.

[Kins002] Witold Kinsner, "Compression and its metrics for multimedia," in *Proceedings of 2002 1st IEEE International Conference on Cognitive Informatics, 2002 ICCI*, (Calgary, AB, Canada; August 2002), 2002, pp. 107–121, {doi: 10.1109/COGINF.2002.1039289}.

[Kins004] Witold Kinsner, "Is entropy suitable to characterize data and signals for cognitive informatics?," in *Proceedings of the 2004 3rd IEEE International Conference on Cognitive Informatics, 2004 ICCI*, 2004, pp. 6–21, {doi: 10.1109/COGINF.2004.1327455}.

[Kins007] Witold Kinsner, "A unified approach to fractal dimensions," *Int. J. Cogn. Informatics Nat. Intell.*, vol. 1, no. 4, pp. 26–46, 2007, {doi: 10.4018/jcini.2007100103}.

[Kins009] Witold Kinsner, "Challenges in the design of adaptive, intelligent and cognitive systems," *Intern. J. Softw. Sci. Comput. Intell.*, vol. 1, no. 3, pp. 16–35, July 2009.

[Kins011] Witold Kinsner, "It's time for multiscale analysis and synthesis in cognitive systems," in *2011 10th IEEE International Conference on Cognitive Informatics & Cognitive Computing, 2011 ICCI*CC, Keynote*, 2011, pp. 7–10, {doi: 10.1109/COGINF.2011.6016116}.

[Kins014] Witold Kinsner, "Towards humanitarian technology education," in *Proc. IEEE 2014 International Humanitarian Technology Conference, IHTC 2014*, (Montreal, QC, Canada; June 2014), 2014.

[Kins015] Witold Kinsner, "Author guide for engineering course project reports and research papers," Winnipeg, MB, Canada, 2015.

[Kins017] Witold Kinsner, *Microcontroller, Microprocessor, and Microcomputer Interfacing for Real-Time Systems*. Winnipeg, MB, Canada: University of Manitoba, 2017, 452 pp.

[Kins020] Witold Kinsner, *Fractal and Chaos Engineering: Monoscale, Multiscale and Polyscale Analyses*. Winnipeg, MB, Canada: OCO Research, 2020, 1096 pp., {ISBN: 978-0-9939347-1-1}.

[KLCC006] Yoohwan Kim, Wing Cheong Lau, Mooi Choo Chuah, and H. Jonathan Chao, "PacketScore: A statistics-based packet filtering scheme against distributed denial-of-

service attacks," *IEEE Trans. Dependable Secur. Comput.*, vol. 3, no. 2, pp. 141–155, April 2006, {doi: 10.1109/TDSC.2006.25, ISBN: 1545-5971 VO - 3}.

[Knut013] Kevin H. Knuth, "Optimal data-based binning for histograms," *ARXIV Physics. Data Anal. Stat. Probab.*, vol. 2, no. 1, pp. 30, September 2013.

[Koen003] Billy Vaughn Koen, *Discussion of the Method: Conducting the Engineer's Approach to Problem Solving*. Oxford, UK: Oxford University Press, 2003, 273 pp., {ISBN: 978-0195-15599-0}.

[KoSV007] Ramana Rao Kompella, Sumeet Singh, and George Varghese, "On scalable attack detection in the network," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 14–25, February 2007, {doi: 10.1109/TNET.2006.890115, ISBN: 1063-6692 VO - 15}.

[KPBH015] Siwar Kriaa, Ludovic Pietre-Cambacedes, Marc Bouissou, and Yoran Halgand, "A survey of approaches combining safety and security for industrial control systems," *Reliab. Eng. Syst. Saf.*, vol. 139, pp. 156–178, July 2015, {doi: http://dx.doi.org/10.1016/j.ress.2015.02.008, ISSN: 0951-8320}.

[KPHD015] William Knowles, Daniel Prince, David Hutchison, Jules Ferdinand Pagna Disso, and Kevin Jones, "A survey of cyber security management in industrial control systems," *Int. J. Crit. Infrastruct. Prot.*, vol. 9, pp. 52–80, June 2015, {doi: http://dx.doi.org/10.1016/j.ijcip.2015.02.002, ISSN: 1874-5482}.

[KrKr014] Abram Krislock and Nathan Krislock, "Resolving histogram binning dilemmas with binless and binfull algorithms," *ARXIV Physics. Data Anal. Stat. Probab.*, pp. 1–19, May 2014.

[KSSS014] Ahmad Karim, Rosli Salleh, Muhammad Shiraz, Syed Shah, Irfan Awan, and Nor Anuar, "Botnet detection techniques: Review, future trends, and issues," *Comput. Electron.*, vol. 15, no. 11, pp. 943–983, 2014, {doi: 10.1631/jzus.C1300242, ISSN: 1869-1951}.

[KuBL994] P. Kundur, Neal J. Balu, and Mark G. Lauby, *Power system stability and control*. New York, NY, USA: McGraw-Hill, 1994, 1176 pp., {ISBN: 007035958X 9780070359581 0070635153 9780070635159}.

[Kuma016] Gulshan Kumar, "Denial of service attacks – an updated perspective," *Syst. Sci. Control Eng.*, vol. 4, no. 1, pp. 285–294, October 2016, {doi: 10.1080/21642583.2016.1241193}.

[KuKn003] Aleksandar Kuzmanovic and Edward W. Knightly, "Low-rate TCP-targeted denial of service attacks: The shrew vs. the mice and elephants," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, (New York, NY, USA; August 2003), 2003, pp. 75–86, {doi: 10.1145/863955.863966, ISBN: 1-58113-735-4}.

[LaGo018] Sharad R. Laxpati and Vladimir Goncharoff, *Practical Signal Processing and Its Applications: With Solved Homework Problems (Advanced Series in Electrical and Computer Engineering Vol. 17)*. Hackensack, NJ, USA: World Scientific Publishing Company, 2018, 660 pp., {ISBN: 978-9813224025}.

[LaJL008] Ulf E. Larson, Erland Jonsson, and Stefan Lindskog, "A revised taxonomy of data collection mechanisms with a focus on intrusion detection," in *2008 3rd International Conference on Availability, Reliability and Security, 2008 ARES*, (Barcelona, Spain; March 2008), 2008, pp. 624–629, {doi: 10.1109/ARES.2008.38}.

[LaJo006a] U. E. Larson and E. Jonsson, “An intrusion detection-centric taxonomy and survey of mechanisms for computer system logging,” in *Proceedings of the 11th Nordic Workshop on Secure IT-systems, 2006 NordSec*, (Linköping, Sweden; October 2006), 2006.

[LaJo006b] U. E. Larson and E. Jonsson, *An Intrusion Detection-Centric Taxonomy and Survey of Data Log Mechanisms*. Gothenburg, Sweden: Chalmers University of Technology, Department of Computer Science and Engineering, 2006.

[Laro005] Daniel T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*, vol. 222. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005, {doi: 10.1002/0471687545, ISBN: 9780471666578}.

[Laru993] J. R. Larus, “Efficient program tracing,” *Computer (Long. Beach. Calif).*, vol. 26, no. 5, pp. 52–61, 1993, {doi: 10.1109/2.211900, ISBN: 0018-9162 VO - 26}.

[LaWD992] Aurel A. Lazar, Weiguo Wang, and Robert H. Deng, “Models and algorithms for network fault detection and identification: A review,” in *1992 Singapore ICCS/ISITA “Communications on the Move,”* 1992, pp. 999–1003 vol.3, {doi: 10.1109/ICCS.1992.255112}.

[LBMC993] Carl E. Landwehr, Alan R. Bull, John P. McDermott, and William S. Choi, “A taxonomy of computer program security flaws, with examples,” *ACM Comput. Surv.*, vol. 26, no. 3, pp. 211–254, September 1993.

[Lesl014] Sneha Leslie, “Tracing back the botmaster,” *Int. J. Eng. Res. Appl.*, vol. 4, no. 12, pp. 4, 2014, {ISSN: 2248-9622}.

[LeWi991] Will E. Leland and Daniel V. Wilson, “High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection,” in *Proc. of 1991 IEEE INFCOM Conference on Computer Communications. 10th Annual Joint Conference of the IEEE Computer and Communications Societies*, (Bal Harbour, FL,USA; April 1991), 1991, vol. 3, pp. 1360–1366, {doi: 10.1109/INFCOM.1991.147663, ISBN: VO -}.

[LeXi001] Wenke Lee and Dong Xiang, “Information-theoretic measures for anomaly detection,” in *Proceedings of 2001 IEEE Symposium on Security and Privacy, 2001 S&P*, 2001, pp. 130–143, {doi: 10.1109/SECPRI.2001.924294, ISBN: 1081-6011 VO -}.

[LGCP011] Zhichun Li, Anup Goyal, Yan Chen, and Vern Paxson, “Towards situational awareness of large-scale botnet probing events,” *IEEE Trans. Inf. Forensics Secur.*, vol. 6, no. 1, pp. 175–188, March 2011, {doi: 10.1109/TIFS.2010.2086445, ISBN: 1556-6013 VO - 6}.

[LiJo997] Ulf Lindqvist and Erland Jonsson, “How to systematically classify computer security intrusions,” in *Proceedings of 1997 IEEE Symposium on Security and Privacy*, (Oakland, CA, USA; May 1997), 1997, pp. 154–163, {doi: 10.1109/SECPRI.1997.601330, ISBN: 1081-6011 VO -}.

[LiJZ009] Chao Li, Wei Jiang, and Xin Zou, “Botnet: Survey and case study,” *2009 4th Int. Conf. Innov. Comput. Inf. Control. 2009 ICICIC*, pp. 1184–1187, 2009, {doi: 10.1109/ICICIC.2009.127, ISBN: 9780769538730}.

[LiLe003] Lan Li and Gyungho Lee, “DDoS attack detection and wavelets,” in *Proceedings of the 2003 12th International Conference on Computer Communications and Networks, 2003 ICCCN*, 2003, pp. 421–427, {doi: 10.1109/ICCCN.2003.1284203, ISBN: 1095-2055}.

[LJYZ017] Jianhua Li, Jiong Jin, Dong Yuan, and Hongke Zhang, “Virtual fog: A virtualization enabled fog computing framework for Internet of things,” *IEEE Internet Things J.*, no. 99, pp. 1–10, November 2017, {doi: 10.1109/JIOT.2017.2774286}.

[Lloy982] Stuart P. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, March 1982, {doi: 10.1109/TIT.1982.1056489, ISSN: 0018-9448 VO - 28}.

[LTWW994] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson, “On the self-similar nature of ethernet traffic (extended version),” *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 1–15, February 1994, {doi: 10.1109/90.282603, ISSN: 1063-6692 VO - 2}.

[Losh016] Peter Loshin, “Dyn hit by massive DNS DDoS, Eastern U.S. bears brunt of attacks,” *TechTarget SearchSecurity*, 2016. [Online]. Available: https://searchsecurity.techtarget.com/news/450401541/Dyn-hit-by-massive-DNS-DDoS-Eastern-US-bears-brunt-of-attacks. [Accessed: 19-Apr-2018].

[Lunt993] Teresa F. Lunt, “A survey of intrusion detection techniques,” *Comput. Secur.*, vol. 12, no. 4, pp. 405–418, June 1993, {doi: 10.1016/0167-4048(93)90029-5, ISSN: 0167-4048}.

[LWFT007] Kejie Lu, Dapeng Wu, Jieyan Fan, Sinisa Todorovic, and Antonio Nucci, “Robust and efficient detection of DDoS attacks for large-scale Internet,” *Comput. Netw.*, vol. 51, no. 18, pp. 5036–5056, December 2007, {doi: 10.1016/j.comnet.2007.08.008, ISSN: 1389-1286}.

[LXKX020] Zhiyuan Li, Weijia Xing, Samer Khamaiseh, and Dianxiang Xu, “Detecting saturation attacks based on self-similarity of OpenFlow traffic,” *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 1, pp. 607–621, March 2020, {doi: 10.1109/TNSM.2019.2959268, ISSN: 1932-4537}.

[LYWZ011] Zhong Liu, Dong-sheng Yang, Ding Wen, Wei-ming Zhang, and Wenji Mao, “Cyber-physical-social systems for command and control,” *IEEE Intell. Syst.*, vol. 26, no. 4, pp. 92–96, July 2011, {doi: 10.1109/MIS.2011.69, ISSN: 1541-1672 VO - 26}.

[MaCa021] Wojciech Mazurczyk and Luca Caviglione, “Cyber Reconnaissance Techniques,” Commun. ACM, vol. 64, no. 3, pp. 86–95, February 2021, {doi: 10.1145/3418293, ISSN: 0001-0782}.

[MaCh014] Xinlei Ma and Yonghong Chen, “DDoS detection method based on chaos analysis of network traffic entropy,” *IEEE Commun. Lett.*, vol. 18, no. 1, pp. 114–117, 2014, {doi: 10.1109/LCOMM.2013.112613.132275, ISBN: 1089-7798 VO - 18}.

[Mand969] Benoit Mandelbrot, “Long-run linearity, locally Gaussian process, H-spectra and infinite variances,” *Int. Econ. Rev.*, vol. 10, no. 1, pp. 82–111, February 1969, {doi: 10.2307/2525574, ISSN: 00206598, 14682354}.

[Marq011] Oge Marques, *Practical image and video processing using MATLAB*. Hoboken, New Jersey: Wiley-IEEE Press, 2011, {doi: 10.1002/9781118093467, ISBN: 1-118-09347-X}.

[Mayf013] John E. Mayfield, *Complex Systems: Evolution as Computation*. New York, NY, USA: Columbia University Press, 2013, 400 pp., {doi: 10.7312/columbia/9780231163040.003.0009}.

[McGu008] D. Kevin McGrath and Minaxi Gupta, “Behind phishing: An examination of phisher modi operandi,” in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, (Berkeley, CA, USA; April 2008), 2008, pp. 4:1–4:8.

[McPi943] Warren S. McCulloch and Walter H. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.

[McRH995] James L. McClelland, David E. Rumelhart, and Geoffrey E Hinton, “The Appeal of Parallel Distributed Processing,” in *Computation and Intelligence: Collected Readings*, George F. Luger, Ed. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1995, pp. 305–341, {ISBN: 0-262-62101-0}.

[MeTW013] Lei Meng, Ah Hwee Tan, and Donald C. Wunsch II, “Vigilance adaptation in adaptive resonance theory,” in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, (Dallas, TX, USA; August 2013), 2013, pp. 1–7, {doi: 10.1109/IJCNN.2013.6706857, ISBN: 9781467361293, ISSN: 2161-4393}.

[Mieg011] Piet Van Mieghem, *Graph Spectra for Complex Networks*. Cambridge, MA, USA: Cambridge University Press, 2011, 346 pp., {ISBN: 978-1107411470}.

[MiOK009] Piet Van Mieghem, Jasmina Omic, and Robert Kooij, “Virus spread in networks,” in *IEEE/ACM Transactions on Networking*, 2009, vol. 17, no. 1, pp. 1–14, {doi: 10.1109/TNET.2008.925623, ISBN: 1063-6692 VO - 17}.

[Mont014] Guido F. Montúfar, “Universal approximation depth and errors of narrow belief networks with discrete units,” *Neural Comput.*, vol. 26, no. 7, pp. 1386–1407, July 2014, {doi: 10.1162/NECO_a_00601, ISSN: 0899-7667}.

[MoSt017] Amir Modarresi and James P. G. Sterbenz, “Toward resilient networks with fog computing,” in *2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM 2017)*, (Alghero, Italy; September 2017), 2017, pp. 1–7, {doi: 10.1109/RNDM.2017.8093032, ISBN: VO -}.

[MPCB014] Guido F. Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio, “On the Number of Linear Regions of Deep Neural Networks,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, (Montreal, Canada; December 2014), 2014, pp. 2924–2932.

[MSBV006] David Moore, Colleen Shannon, Douglas J Brown, Geoffrey M Voelker, and Stefan Savage, “Inferring Internet denial-of-service activity,” *ACM Trans. Comput. Syst.*, vol. 24, no. 2, pp. 115–139, May 2006, {doi: 10.1145/1132026.1132027, ISSN: 0734-2071}.

[MSSV009a] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker, “Beyond blacklists: Learning to detect malicious web sites from suspicious URLs,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (New York, NY, USA; May 2009), 2009, pp. 1245–1254, {doi: 10.1145/1557019.1557153, ISBN: 978-1-60558-495-9}.

[MSSV009b] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker, “Identifying suspicious URLs: An application of large-scale online learning,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, (New York, NY, USA; May 2009), 2009, pp. 681–688, {doi: 10.1145/1553374.1553462, ISBN: 978-1-60558-516-1}.

[NaSP013] A. S. Syed Navaz, V. Sangeetha, and C. Prabhadevi, “Entropy based anomaly detection system to prevent DDoS attacks in Cloud,” *ARXIV*, 2013.

[Nati018] National Initiative for Cybersecurity Careers and Studies, “A glossary of common cybersecurity terminology,” *November 28*, 2018. [Online]. Available: https://niccs.us-cert.gov/about-niccs/glossary. [Accessed: 20-Jun-2020].

[NGWL013] Tongguang Ni, Xiaoqing Gu, Hongyuan Wang, and Yu Li, “Real-time detection of application-layer DDoS attack using time series analysis,” *J. Control Sci. Eng.*, vol. 2013, pp. 1–6, 2013, {doi: 10.1155/2013/821315, ISSN: 1687-5249}.

[NHAB017] Ivan Nunes da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, and Silas Franco dos Reis Alves, *Artificial Neural Networks: A Practical Course*. Bern, Switzerland: Springer International Publishing Switzerland, 2017, 309 pp., {doi: 10.1145/242224.242229, ISBN: 0070428077, ISSN: 87567016}.

[NiBe016] Oliver Niggemann and Jürgen Beyerer, Editors, *Machine Learning for Cyber Physical Systems: Selected Papers from the International Conference ML4CPS 2015*. Springer-Verlag Berlin Heidelberg, 2016, 124 pp., {doi: 10.1007/978-3-662-48838-6}.

[Nowa020] Peter Nowak, “Canada a target for cyberattacks on COVID-19 research,” *The Globe and Mail*, 2020. [Online]. Available: https://www.theglobeandmail.com/featured-reports/article-canada-a-target-for-cyberattacks-on-covid-19-research/. [Accessed: 16-Aug-2020].

[NWCB011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng, “Reading digits in natural images with unsupervised feature learning,” in *NIPS 2011 - Neural Information Processing Systems - Workshop on Deep Learning and Unsupervised Feature Learning*, (Granada, Spain; December 2011), 2011, pp. 1–9.

[Open017] OpenFog Consortium Architecture Working Group, *OpenFog Reference Architecture for Fog Computing*. OpenFog Consortium, 2017, 949 pp.

[Osbo020] Charlie Osborne, “16 DDoS attacks take place every 60 seconds, rates reach 622 Gbps,” *Zero Day Net*, 2020. [Online]. Available: https://www.zdnet.com/article/16-ddos-attacks-take-place-every-60-seconds-rates-reach-622-gbps/. [Accessed: 20-Apr-2020].

[ÖzBr015] İlker Özçelik and Richard R. Brooks, “Deceiving entropy based DoS detection,” *Comput. Secur.*, vol. 48, pp. 234–245, February 2015, {doi: http://dx.doi.org/10.1016/j.cose.2014.10.013, ISSN: 0167-4048}.

[PaFl995] Vern Paxson and Sally Floyd, “Wide area traffic: The failure of Poisson modeling,” *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 226–244, 1995, {doi: 10.1109/90.392383, ISBN: 1063-6692 VO - 3}.

[Page954] E. S. Page, “Continuous inspection schemes,” *Biometrika Trust*, vol. 41, no. 1–2, pp. 100–115, June 1954, {ISSN: 0006-3444}.

[PaLi020] Raisa Patel and Philip Ling, “CRA shuts down online services after thousands of accounts breached in cyberattacks,” *CBC News*, 2020. [Online]. Available: https://www.cbc.ca/news/politics/canada-revenue-agency-cra-cyberattack-1.5688163?fbclid=IwAR3tAr05ZmJEXlDH_s4i9YmvlLGvBf_YTsxLKdo7kN5qHQbyVqh-9cc3D40. [Accessed: 16-Aug-2020].

[Pate015] Prachi Patel, “Chief scientist of CERT leads IEEE’s efforts in advancing cybersecurity,” *IEEE The Institute*, 2015. [Online]. Available: http://theinstitute.ieee.org/people/profiles/chief-scientist-of-cert-leads-ieees-efforts-in-advancing-cybersecurity.

[Path015] Al-Sakib Khan Pathan, Editor, *Securing Cyber-Physical Systems*. Boca Raton, FL, USA: CRC Press/Taylor & Francis Group, 2015, 408 pp.

[PCDL009] Roberto Perdisci, Igino Corona, David Dagon, and Wenke Lee, “Detecting malicious flux service networks through passive analysis of recursive DNS traces,” in

2009 Annual Computer Security Applications Conference, 2009 ACSAC, 2009, pp. 311–320, {doi: 10.1109/ACSAC.2009.36, ISBN: 1063-9527 VO -}.

[PeJS004] Heinz-Otto Peitgen, H Jürgens, and Dietmar Saupe, *Chaos and Fractals: New Frontiers of Science*. New York: Springer, 2004, 730 pp., {ISBN: 9780387202297}.

[PeLR007] Tao Peng, Christopher Leckie, and Kotagiri Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Comput. Surv. CSUR*, vol. 39, no. 1, April 2007, {doi: 10.1145/1216370.1216373, ISSN: 0360-0300}.

[Perk010] Jeffrey Perkel, "Cybersecurity: How safe are your data?," *Nature*, vol. 464, no. 7293, pp. 1260, 2010, {doi: 10.1038/4641260a}.

[PFND006] Lok-Fu Pak, M. Omar Faruque, Xin Nie, and Venkata Dinavahi, "A versatile cluster-based real-time digital simulator for power engineering research," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 455–465, May 2006, {doi: 10.1109/TPWRS.2006.873414, ISSN: 1558-0679}.

[PGCB014] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio, "How to construct deep recurrent neural networks," in *2014 ICLR, International Conference on Learning Representations*, (Banff, Canada; April 2014), 2014, pp. 1–13.

[PKKG010] Pawan Prakash, Manish Kumar, Ramana Rao Kompella, and Minaxi Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1–5, {doi: 10.1109/INFCOM.2010.5462216, ISBN: 0743-166X VO -}.

[Plan015] PLANETLAB, "Planetlab: An open platform for developing, deploying, and accesing planetary-scale services," 2015. [Online]. Available: http://www.planet-lab.org. [Accessed: 18-Nov-2015].

[PoSY009] Phillip Porras, Hassen Saïdi, and Vinod Yegneswaran, "A foray into Conficker's logic and rendezvous points," in *Proceedings of the 2Nd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More, 2009 LEET*, (Berkeley, CA, USA; April 2009), 2009, pp. 7.

[Prat001] William K. Pratt, *Digital Image Processing*, 3rd ed. Los Altos, CA, USA: John Wiley & Sons Inc., 2001, 738 pp., {ISBN: 0-471-22132-5}.

[PrMa996] John G. Proakis and Dimitris G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1996, 1033 pp.

[ReRL014] V. Sushma Reddy, K. Damodar Rao, and P. Sowmya Lakshmi, "Efficient detection of DDoS attacks by entropy variation," *IOSR J. Comput. Eng. IOSRJCE*, 2014, {doi: 10.6084/M9.FIGSHARE.1145931}.

[Reze014] Chris Rezendes, "Rethink money and meaning with the Internet of Things," *TEDx San Diego*, 2014. [Online]. Available: http://tedxtalks.ted.com/video/reThink-Money-and-Meaning-with;search:intern of things. [Accessed: 04-Aug-2015].

[Ride008] Jamie Riden, "How fast-flux service networks work," *The Honeynet Project,* 2008. [Online]. Available: https://www.honeynet.org/node/132. [Accessed: 21-Nov-2015].

[Roze015a] Monica Rozenfeld, "For hire: Ethical hackers," *IEEE The Institute*, 2015. [Online]. Available: http://theinstitute.ieee.org/ieee-roundup/opinions/ieee-roundup/for-hire-ethical-hackers.

[Roze015b] Monica Rozenfeld, “Six ways to improve mobile app security and p rivacy,” *IEEE The Institute*, 2015. [Online]. Available: http://theinstitute.ieee.org/ieee-roundup/opinions/ieee-roundup/six-ways-to-improve-mobile-app-security-and-privacy.

[RPCL006] Marc’Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann LeCun, “Efficient learning of sparse representations with an energy-based model,” in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, (Cambridge, MA, USA; December 2006), 2006, pp. 1137–1144.

[RuHW986] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, {doi: 10.1038/323533a0, ISSN: 1476-4687}.

[RuMG986] David E. Rumelhart, James L. McClelland, and University of California-San Diego PDP Research Group., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA, USA: MIT Press, 1986, 576 pp., {ISBN: 978-0262680530}.

[RZMT007] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis, “My botnet is bigger than yours (Maybe, better than yours): Why size estimates remain challenging,” in *Proceedings of the 1st Conference on 1st Workshop on Hot Topics in Understanding Botnets*, (Berkeley, CA, USA; April 2007), 2007, pp. 5.

[SaFF019] Taneeya Satyapanich, Tim Finin, and Francis Ferraro, “Extracting Rich Semantic Information about Cybersecurity Events,” in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 5034–5042, {doi: 10.1109/BigData47090.2019.9006444}.

[Savo015] Michael Savoie, “Modularity and the Internet of Things,” *TEDx Utah Valley University*, 2015. [Online]. Available: http://tedxtalks.ted.com/video/Modularity-and-the-Internet-of;search:intern of things.

[SBRS007] M. Steurer, F. Bogdan, W. Ren, M. Sloderbeck, and S. Woodruff, “Controller and Power Hardware-In-Loop Methods for Accelerating Renewable Energy Integration,” in *2007 IEEE Power Engineering Society General Meeting*, (Tampa, FL, USA; June 2007), 2007, pp. 1–4, {doi: 10.1109/PES.2007.386022, ISSN: 1932-5517}.

[SCGK011] Brett Stone-Gross, Marco Cova, Bob Gilbert, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna, “Analysis of a botnet takeover,” *IEEE Security & Privacy Magazine*, vol. 9, no. 1, pp. 64–72, 2011, {doi: 10.1109/MSP.2010.144, ISBN: 1540-7993 VO - 9}.

[Schr991] Manfred R. Schroeder, *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*. New York, NY, USA: W. H. Freeman and Company, 1991, 429 pp.

[Schr995] Beth A. Schroeder, “On-line monitoring: A tutorial,” *Computer (Long. Beach. Calif).*, vol. 28, no. 6, pp. 72–78, 1995, {doi: 10.1109/2.386988, ISBN: 0018-9162 VO - 28}.

[ScKW996] Analucia Schiaffino Morales De Franceschi, Luiz Fernando Kormann, and Carlos Becker Westphall, “Performance evaluation for proactive network management,” in *1996 IEEE International Conference on Communications, 1996 ICC*, 1996, vol. 1, pp. 22–26 vol.1, {doi: 10.1109/ICC.1996.540238, ISBN: VO - 1}.

[SeLA012] Teresa Serrano-Gotarredona, Bernabé Linares-Barranco, and Andreas G. Andreou, *Adaptive Resonance Theory Microchips: Circuit Design Techniques*. New York, NY, USA: Springer Science+Business Media New York, 2012, 234 pp., {doi: 10.1007/978-1-4419-8710-5, ISBN: 978-1-4419-8710-5}.

[Shan948] Claude Elwood Shannon, “A mathematical theory of communication,” *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, July 1948, {doi: 10.1002/j.1538-7305.1948.tb01338.x, ISBN: 0005-8580 VO - 27}.

[Shan949] Claude Elwood Shannon, “Communication theory of secrecy systems,” *Bell Syst. Tech. Journal,* vol. 28, no. 4, pp. 656–715, October 1949, {doi: 10.1002/j.1538-7305.1949.tb00928.x, ISBN: 0005-8580}.

[ShKG012] Craig A. Shue, Andrew J. Kalafut, and Minaxi Gupta, “Abnormally malicious autonomous systems and their Internet connectivity,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 220–230, February 2012, {doi: 10.1109/TNET.2011.2157699, ISBN: 1063-6692 VO - 20}.

[ShSa016] Subana Shanmuganathan and Sandhya Samarasinghe, Editors, *Artificial Neural Network Modelling*. Cham: Springer International Publishing, 2016, 468 pp., {doi: 10.1007/978-3-319-28495-8}.

[ShSe015] Ravi R. Shenoy and Chandra Sekhar Seelamantula, “Spectral zero-crossings: Localization properties and applications,” *IEEE Trans. Signal Process.*, vol. 63, no. 12, pp. 3177–3190, June 2015, {doi: 10.1109/TSP.2015.2420538, ISBN: 1053-587X VO - 63}.

[ShSe012] Ravi R. Shenoy and Chandra Sekhar Seelamantula, “Spectral zero-crossings: Localization properties and application to epoch extraction in speech signals,” in *2012 International Conference on Signal Processing and Communications, 2012 SPCOM*, (Bangalore, India; July 2012), 2012, pp. 1–5, {doi: 10.1109/SPCOM.2012.6290218}.

[ShSh007] Hideaki Shimazaki and Shigeru Shinomoto, “A method for selecting the bin size of a time histogram,” *Neural Comput.*, vol. 19, no. 6, pp. 1503–1527, 2007, {doi: 10.1162/neco.2007.19.6.1503}.

[Simo019] Vitaly Simonovich, “Imperva Blocks Our Largest DDoS L7: Brute Force Attack Ever (Peaking at 292,000 RPS),” *Imperva: Research Labs*, 2020. [Online]. Available: https://www.imperva.com/blog/imperva-blocks-our-largest-ddos-l7-brute-force-attack-ever-peaking-at-292000-rps/. [Accessed: 20-Apr-2020].

[SiPK017] Dick Sigmund, Gyeong-Moon Park, and Jong-Hwan Kim, “Context preference-based deep adaptive resonance theory: Integrating user preferences into episodic memory encoding and retrieval,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, (Anchorage, AK, USA; May 2017), 2017, pp. 1879–1886, {doi: 10.1109/IJCNN.2017.7966079}.

[SiRP020] Kyle A. Simpson, Simon Rogers, and Dimitrios P. Pezaros, “Per-Host DDoS Mitigation by Direct-Control Reinforcement Learning,” *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 1, pp. 103–117, March 2020, {doi: 10.1109/TNSM.2019.2960202, ISSN: 1932-4537}.

[Skeg015] Richard Skeggs, “Identifying the needle in the Internet of Things haystack,” *TEDx University of Essex*, 2015. [Online]. Available: http://tedxtalks.ted.com/video/Identifying-the-needle-in-the-I;search:intern of things. [Accessed: 31-Jul-2015].

[SLOB007] Antoine Scherrer, Nicolas Larrieu, Philippe Owezarski, Pierre Borgnat, and Patrice Abry, “Non-Gaussian and long memory statistical characterizations for Internet traffic with anomalies,” *IEEE Trans. Dependable Secur. Comput.*, vol. 4, no. 1, pp. 56–70, 2007, {doi: 10.1109/TDSC.2007.12, ISBN: 1545-5971 VO - 4}.

[SLSS014] Alexander Smirnov, Tatiana Levashova, Nikolay Shilov, and Kurt Sandkuhl, "Ontology for cyber-physical-social systems self-organisation," in *Proceedings of 16th Conference of Open Innovations Association FRUCT*, (Oulu, Finland; October 2014), 2014, pp. 101–107, {doi: 10.1109/FRUCT.2014.7000933, ISBN: 2305-7254 VO -}.

[SBWH13] Michael Swearingen, Steven Brunasso, Joe Weiss, Dennis Huber, "What you need to know (and don't) about the AURORA vulnerability," *Power*, 2013. [Online]. Available: http://www.powermag.com/what-you-need-to-know-and-dont-about-the-aurora-vulnerability/. [Accessed: 27-Oct-2020].

[SmKP015] Alexander Smirnov, Alexey Kashevnik, and Andrew Ponomarev, "Multi-level self-organization in cyber-physical-social systems: Smart home cleaning scenario," *7th Ind. Prod. Syst. Conf. - PSS, Ind. Transform. Sustain. Bus.*, vol. 30, pp. 329–334, 2015, {doi: 10.1016/j.procir.2015.02.089, ISSN: 2212-8271}.

[Solo020] Howard Solomon, "Canadian accounting association website gets hacked," *Financial Post*, 2020. [Online]. Available: https://financialpost.com/technology/tech-news/canadian-accounting-association-website-gets-hacked. [Accessed: 16-Aug-2020].

[SoMA009] Fabio Soldo, Athina Markopoulou, and Katerina Argyraki, "Optimal filtering of source address prefixes: Models and algorithms," in *2009 IEEE INFOCOM*, 2009, pp. 2446–2454, {doi: 10.1109/INFCOM.2009.5062172, ISBN: 0743-166X VO -}.

[Spro003] Julien Clinton Sprott, *Chaos and Time-Series Analysis*. Oxford, UK: Oxford University Press, 2003, 507 pp.

[SqAS011] Mohammed H. Sqalli, Fahd Al-Haidari, and Khaled Salah, "EDoS-shield - A two-steps mitigation technique against EDoS attacks in Cloud computing," in *2011 4th IEEE International Conference on Utility and Cloud Computing, 2011 UCC*, 2011, pp. 49–56, {doi: 10.1109/UCC.2011.17}.

[Stat014] Statistical Machine Translation, "Translation task - ACL 2014 Ninth Workshop on Statistical Machine Translation," 2014. [Online]. Available: https://www.statmt.org/wmt14/translation-task.html. [Accessed: 19-Jul-2019].

[Stin006] Douglas R. Stinson, *Cryptography: Theory and Practice*, 3rd ed. Boca Raton, FL, USA: CRC Press Inc., 2006, 593 pp., {ISBN: 978-1584885085}.

[Stur926] Herbert A. Sturges, "The choice of a class interval," *J. Am. Stat. Assoc.*, vol. 21, no. 153, pp. 65–66, 1926, {ISSN: 01621459}.

[SuRH016] Siddharth Suryanarayanan, Robin Roche, and Timothy M. Hansen, Editors, *Cyber-Physical-Social Systems and Constructs in Electric Power Engineering*. Stevenage, UK: Institution of Engineering and Technology, 2016, 485 pp., {ISBN: 978-1-84919-936-0}.

[SWWJ008] Hemant Sengar, Haining Wang, Duminda Wijesekera, and Sushil Jajodia, "Detecting VoIP floods using the Hellinger distance," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 6, pp. 794–805, 2008, {doi: 10.1109/TPDS.2007.70786, ISBN: 1045-9219 VO - 19}.

[SXLL008] Minho Sung, Jun Xu, Jun Li, and Li Li, "Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Information-Theoretic Foundation," *Networking, IEEE/ACM Trans.*, vol. 16, no. 6, pp. 1253–1266, 2008, {doi: 10.1109/TNET.2007.911427, ISBN: 1063-6692 VO - 16}.

[TaCh011] Jin Tang and Yu Cheng, "Quick detection of stealthy SIP flooding attacks in VoIP networks," in *2011 IEEE International Conference on Communications, 2011 ICC*, 2011, pp. 1–5, {doi: 10.1109/icc.2011.5963248, ISBN: 1550-3607 VO -}.

[TaLe986] Murad S. Taqqu and Joshua B. Levy, "Using renewal processes to generate long range dependencies and high variability," in *Dependence in Probability and Statistics*, (Boston, MA, USA; May 1986), 1986, vol. 11, pp. 73–89.

[Tech017a] Techopedia, "What is a packet analyzer?," 2017. [Online]. Available: https://www.techopedia.com/definition/25323/packet-analyzer. [Accessed: 02-Aug-2017].

[Tech017b] Techopedia, "What is packet capture?," 2017. [Online]. Available: https://www.techopedia.com/definition/25333/packet-capture. [Accessed: 02-Aug-2017].

[TeKi011] Jesus David Terrazas Gonzalez and Witold Kinsner, "A modular dynamical cryptosystem based on continuous cellular automata," in *2011 10th IEEE Intern. Conf. on Cognitive Informatics & Cognitive Computing, 2011 ICCI*CC*, (Banff, AB, Canada; August 2011), 2011, pp. 203–215, {doi: 10.1109/COGINF.2011.6016142, ISBN: 978-1-4577-1695-9}.

[TeKi012a] Jesus David Terrazas Gonzalez and Witold Kinsner, "Evaluating the security level of a cryptosystem based on chaos," *Intern. J. Softw. Sci. Comput. Intell. IJSSCI*, vol. 4, no. 3, pp. 80–120, January 2012, {doi: 10.4018/jssci.2012070105, ISSN: 1942-9045}.

[TeKi012b] Jesus David Terrazas Gonzalez and Witold Kinsner, "Security testing of a modular cryptosystem based on continuous cellular automata," in *2012 11th IEEE Intern. Conf. on Cognitive Informatics & Cognitive Computing, 2012 ICCI*CC*, (Kyoto, Japan; August 2012), 2012, pp. 78–85, {doi: 10.1109/ICCI-CC.2012.6311130, ISBN: 978-1-4673-2794-7}.

[TeKi012c] Jesus David Terrazas Gonzalez and Witold Kinsner, "A modular dynamical cryptosystem based on continuous-interval cellular automata," in *Cognitive Informatics for Revealing Human Cognition: Knowledge Manipulations in Natural Intelligence*, Yingxu Wang, Ed. Calgary, AB, Canada: IGI Global, 2012, pp. 261–286, {doi: 10.4018/978-1-4666-2476-4, ISBN: 9781466624764}.

[TeKi013a] Jesus David Terrazas Gonzalez and Witold Kinsner, "Comparison of cryptosystems using a single-scale statistical measure," in *2013 26th Annual IEEE Canadian Conference on Electrical and Computer Engineering, 2013 CCECE*, (Regina, SK, Canada; May 2013), 2013, pp. 1–5, {doi: 10.1109/CCECE.2013.6567841, ISBN: 0840-7789 VO -}.

[TeKi013b] Jesus David Terrazas Gonzalez and Witold Kinsner, "Comparison of selected cryptosystems using single-scale and poly-scaly measures," in *2013 12th IEEE Intern. Conf. on Cognitive Informatics & Cognitive Computing, 2013 ICCI*CC*, (New York, NY, USA; July 2013), 2013, {doi: 10.1109/ICCI-CC.2013.6622230, ISBN: 978-1-4799-0781-6}.

[TeKi016a] Jesus David Terrazas Gonzalez and Witold Kinsner, "Zero-crossing analysis of Lévy walks for real-time feature extraction," in *2016 IEEE International Conference on Electro Information Technology, 2016 EIT*, (Grand Forks, ND, USA; April 2016), 2016, {doi: 10.1109/EIT.2016.7535276}.

[TeKi016b] Jesus David Terrazas Gonzalez and Witold Kinsner, "Zero-crossing analysis of Lévy walks for real-time feature extraction: Composite analysis for strengthening the IoT against DDoS attacks," in *2016 15th IEEE Intern. Conf. on Cognitive Informatics &*

*Cognitive Computing, 2016 ICCI*CC*, (Standford, CA, USA; August 2016), 2016, {doi: 10.1109/ICCI-CC.2016.7862027}.

[TeKi016c] Jesus David Terrazas Gonzalez and Witold Kinsner, "Zero-crossing analysis of Lévy walks and a DDoS dataset for real-time feature extraction: Composite and applied signal analysis for strengthening the Internet-of-Things against DDoS attacks," *Int. J. Softw. Sci. Comput. Intell. IJSSCI*, vol. 8, no. 4, pp. 1–28, 2016, {doi: 10.4018/IJSSCI.2016100101}.

[TeKi016d] Jesus David Terrazas Gonzalez and Witold Kinsner, "Zero-crossing analysis and information divergence of Lévy walks for real-time feature extraction," *Int. J. Handheld Comput. Res. IJHCR*, vol. 7, no. 4, pp. 41–59, 2016, {doi: 10.4018/IJHCR.2016100104}.

[TeKi018] Jesus David Terrazas Gonzalez and Witold Kinsner, "Multi-scale analysis of skewness for feature extraction in real-time," in *2018 17th IEEE Intern. Conf. on Cognitive Informatics & Cognitive Computing, 2018 ICCI*CC*, (Berkeley, CA, USA; July 2018), 2018.

[TeKi019a] Jesus David Terrazas Gonzalez and Witold Kinsner, "A method of digital signal feature extraction comprising multiscale analysis," Patent No.: WO/2019/227227, 2019.

[TeKi019b] Jesus David Terrazas Gonzalez and Witold Kinsner, "System and method for analyzing Internet traffic to detect distributed denial of service (DDoS) attack," Patent No.: WO/2019/051595, 2019.

[Terr012] Jesus David Terrazas Gonzalez, "A Multi Modular Dynamical Cryptosystem Based on Continuous-Interval Celluar Automata," University of Manitoba: Electrical and Computer Engineering, Winnipeg, MB, Canada, 2012.

[Terr020] Jesus David Terrazas Gonzalez, "Cognitive Detection of Anomalies in Internet Traffic: Traversing from Internet to Interplanetary Internet," in *2020 7th IEEE International Workshop on Space-Terrestrial INTernetworking, (STINT 2020)*, (Vicenza, Italy; October 2020), 2020.

[ThJi003] Marina Thottan and Chuanyi Ji, "Anomaly detection in IP networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2191–2204, August 2003, {doi: 10.1109/TSP.2003.814797, ISBN: 1053-587X VO - 51}.

[ThSD007] Vrizlynn L. L. Thing, Morris Sloman, and Naranker Dulay, "A survey of bots used for distributed denial of service attacks," *Inf. Secur.*, vol. 232/2007, no. May, pp. 229–240, 2007, {doi: 10.1007/978-0-387-72367-9, ISBN: 9780387723662}.

[ToMi985] David S. Touretzky and Geoffrey E. Minton, "Symbols among the neurons: Details of a connectionist inference architecture," in *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, (San Francisco, CA, USA; August 1985), 1985, vol. 1, pp. 238–243, {ISBN: 0-934613-02-8}.

[VCJL017] Nandor Verba, Kuo-Ming Chao, Anne James, Jacek Lewandowski, Xiang Fei, and Chen-Fang Tsai, "Graph analysis of fog computing systems for industry 4.0," in *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE 2017)*, (Shanghai, China, China; November 2017), 2017, pp. 46–53, {doi: 10.1109/ICEBE.2017.17}.

[VeAb999] Darryl Veitch and Patrice Abry, "A wavelet-based joint estimator of the parameters of long-range dependence," *IEEE Trans. Inf. Theory*, vol. 45, no. 3, pp. 878–897, 1999, {doi: 10.1109/18.761330, ISBN: 0018-9448 VO - 45}.

[Voll015] Michael T. Vollmann, "The 414s: The Original Teenage Hackers," *CNN*, 2015. [Online]. Available: http://www.cnn.com/videos/tech/2015/03/10/digital-shorts-original-teenage-hackers-orig.cnn.

[WaBo06] Zhou Wang and Alan C. Bovik, *Modern Image Quality Assessment*. Williston, VT, USA: Morgan & Claypool Publishers, 2006, 157 pp., {doi: 10.2200/S00010ED1V01Y2005081VM003, ISBN: 1598290223}.

[WaJS007] Haining Wang, Cheng Jin, and Kang G. Shin, "Defense against spoofed IP traffic using hop-count filtering," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 40–53, February 2007, {doi: 10.1109/TNET.2006.890133, ISSN: 1063-6692}.

[Wand997] M. P. Wand, "Data-based choice of histogram bin width," *Am. Stat.*, vol. 51, no. 1, pp. 59–64, 1997, {doi: 10.2307/2684697, ISSN: 00031305}.

[WaSZ010] Ping Wang, Sherri Sparks, and Cliff C. Zou, "An advanced hybrid peer-to-peer botnet," *IEEE Trans. Dependable Secur. Comput.*, vol. 7, no. 2, pp. 113–127, April 2010, {doi: 10.1109/TDSC.2008.35, ISBN: 1545-5971 VO - 7}.

[WeKu991] Sholom M. Weiss and Casimir A. Kulikowski, *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. San Mateo, CA, USA: Morgan Kaufmann Publishers, 1991, 223 pp.

[Werb007] Paul J. Werbos, "Using ADP to understand and replicate brain intelligence: The next level design," in *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, (Honolulu, HI, USA; April 2007), 2007, pp. 209–216, {doi: 10.1109/ADPRL.2007.368158, ISBN: 1424407060}.

[Werb009] Paul J. Werbos, "Intelligence in the brain: A theory of how it works and how to build it," *Neural Networks*, vol. 22, no. 3, pp. 200–212, 2009, {doi: 10.1016/j.neunet.2009.03.012, ISSN: 0893-6080}.

[WiKP998] A. Witt, J. Kurths, and A. Pikovsky, "Testing stationarity in time series," *Phys. Rev. E*, vol. 58, no. 2, pp. 1800–1810, August 1998.

[Witt014] Peter Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining*. Amsterdam, Netherlands: Elsevier Academic Press, 2014, 163 pp.

[Wolf008] Julia Wolf, "Technical details of Srizbi's domain generation algorithm," 2008. [Online]. Available: https://www.fireeye.com/blog/threat-research/2008/11/technical-details-of-srizbis-domain-generation-algorithm.html.

[Womb015] WOMBAT, "Worldwide observatory of malicious behaviors and attack threats (WOMBAT)," 2015. [Online]. Available: http://www.wombat-project.eu. [Accessed: 18-Nov-2015].

[Worn996] Gregory Wornell, *Signal Processing with Fractals: A Wavelet-Based Approach*. Upper Saddle River, NJ, USA: Prentice Hall, 1996, 177 pp.

[Wulf008] William A. Wulf, "Responsible citizenship in a technological democracy," *IEEE Central Virginia Section Course*. 21-Feb-2008.

[WuOL011] Zheng Wu, Yang Ou, and Yujun Liu, "A taxonomy of network and computer attacks based on responses," in *2011 International Conference on Information Technology, Computer Engineering and Management Sciences, 2011 ICM*, (Nanjing, Jiangsu, China; September 2011), 2011, vol. 1, pp. 26–29, {doi: 10.1109/ICM.2011.363}.

[WWLL017] Meng Wang, Jun Wu, Gaolei Li, Jianhua Li, and Qiang Li, "Fog computing based content-aware taxonomy for caching optimization in information-centric networks," in

2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), (Atlanta, GA, USA; November 2017), 2017, pp. 474–475, {doi: 10.1109/INFCOMW.2017.8116422}.

[XiLS001] Yong Xiong, Steve Liu, and Peter Sun, "On the defense of the distributed denial of service attacks: An on-off feedback control approach," *IEEE Trans. Syst. Man Cybern. Part A Syst. Humans*, vol. 31, no. 4, pp. 282–293, 2001, {doi: 10.1109/3468.935045, ISBN: 1083-4427 VO - 31}.

[XiRa015] Feng Xia and Azizur Rahim, *MAC Protocols for Cyber-Physical Systems*. Berlin, Germany: Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, 98 pp.

[XiYu009a] Yi Xie and Shun-zheng Yu, "A large-scale hidden semi-Markov model for anomaly detection on user browsing behaviors," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 54–65, 2009, {doi: 10.1109/TNET.2008.923716, ISBN: 1063-6692 VO - 17}.

[XiYu009b] Yi Xie and Shun-zheng Yu, "Monitoring the application-layer DDoS attacks for popular websites," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 15–25, 2009, {doi: 10.1109/TNET.2008.925628, ISBN: 1063-6692 VO - 17}.

[XiZG009] Yang Xiang, Wanlei Zhou, and Minyi Guo, "Flexible deterministic packet marking: An IP traceback system to find the real source of attacks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 4, pp. 567–580, 2009, {doi: 10.1109/TPDS.2008.132, ISBN: 1045-9219 VO - 20}.

[XQQL015] Peng Xiao, Wenyu Qu, Heng Qi, and Zhiyang Li, "Detecting DDoS attacks against data center with correlation analysis," *Comput. Commun.*, vol. 67, pp. 66–74, August 2015, {doi: http://dx.doi.org/10.1016/j.comcom.2015.06.012, ISSN: 0140-3664}.

[XYAP008] Yinglian Xie, Fang Yu, Kannan Achan, Rina Panigrahy, Geoff Hulten, and Ivan Osipkov, "Spamming botnets: Signatures and characteristics," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 171–182, August 2008, {doi: 10.1145/1402946.1402979, ISSN: 0146-4833}.

[Yang008] Chritopher C. Yang, "Information sharing and privacy protection of terrorist or criminal social networks," in *2008 IEEE International Conference on Intelligence and Security Informatics, 2008 ISI*, 2008, pp. 40–45, {doi: 10.1109/ISI.2008.4565027}.

[YaPS005] Abraham Yaar, Adrian Perrig, and Dawn Song, "FIT: Fast Internet traceback," in *2005 24th Annual Joint Conference of the IEEE/INFOCOM Computer and Communications Societies*, 2005, vol. 2, pp. 1395–1406 vol. 2, {doi: 10.1109/INFCOM.2005.1498364, ISBN: 0743-166X VO - 2}.

[YaPS006] Aabraham Yaar, Adrian Perrig, and Dawn Song, "StackPi: New packet marking and filtering mechanisms for DDoS and IP spoofing defense," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 10, pp. 1853–1863, October 2006, {doi: 10.1109/JSAC.2006.877138, ISBN: 0733-8716 VO - 24}.

[YASR008] Jie Yu, Jaume Amores, Nicu Sebe, Petia Radeva, and Qi Tian, "Distance learning for similarity estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 451–462, 2008, {doi: 10.1109/TPAMI.2007.70714, ISBN: 0162-8828 VO - 30}.

[YKGF008] Haifeng Yu, Michael Kaminsky, Philip B. Gibbons, and Abraham D. Flaxman, "SybilGuard: Defending against Sybil attacks via social networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 576–589, 2008, {doi: 10.1109/TNET.2008.923723, ISBN: 1063-6692 VO - 16}.

[YKPB013] Jaehak Yu, Hyunjoong Kang, Daeheon Park, Hyo-Chan Bang, and Do Kang, “An in-depth analysis on traffic flooding attacks detection and system using data mining techniques,” *J. Syst. Archit.*, vol. 59, no. 10, pp. 1005, 2013, {ISSN: 13837621}.

[YRRR012] Sandeep Yadav, Ashwath Kumar Krishna Reddy, A. L. Narasimha Reddy, and Supranamaya Ranjan, “Detecting algorithmically generated domain-flux attacks with DNS traffic analysis,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1663–1677, 2012, {doi: 10.1109/TNET.2012.2184552, ISBN: 1063-6692 VO - 20}.

[YSKG009] Haifeng Yu, Chenwei Shi, Michael Kaminsky, Philip B. Gibbons, and Feng Xiao, “DSybil: Optimal Sybil-resistance for recommendation systems,” in *2009 30th IEEE Symposium on Security and Privacy*, 2009, pp. 283–298, {doi: 10.1109/SP.2009.26, ISBN: 1081-6011 VO -}.

[YTLW009] Shui Yu, Theerasak Thapngam, Jianwen Liu, Su Wei, and Wanlei Zhou, “Discriminating DDoS flows from flash crowds using information distance,” in *2009 3rd International Conference on Network and System Security, 2009 NSS*, 2009, pp. 351–356, {doi: 10.1109/NSS.2009.29}.

[Yu014] Shui Yu, *Distributed Denial of Service Attack and Defense*. New York, NY, USA: Springer, 2014, 104 pp., {doi: 10.1007/978-1-4614-9491-1, ISBN: 978-1-4614-9491-1}.

[YuLi008] Wei Yu and K. J. Ray Liu, “Secure cooperation in autonomous mobile ad-hoc networks under noise and imperfect monitoring: A game-theoretic approach,” *IEEE Trans. Inf. Forensics Secur.*, vol. 3, no. 2, pp. 317–330, June 2008, {doi: 10.1109/TIFS.2008.922453, ISBN: 1556-6013 VO - 3}.

[YuUS019] Ahmad Riza’ain Yusof, Nur Izura Udzir, and Ali Selamat, “Systematic literature review and taxonomy for DDoS attack detection and prediction,” *Int. J. Digit. Enterp. Technol.*, vol. 1, no. 3, February 2019, {doi: 10.1504/ijdet.2019.10019068}.

[YuZW010] Sheng Yu, Shijie Zhou, and Sha Wang, “Fast-flux attack network identification based on agent lifespan,” in *2010 IEEE International Conference on Wireless Communications, Networking and Information Security, 2010 WCNIS*, 2010, pp. 658–662, {doi: 10.1109/WCINS.2010.5541861}.

[YWFX009] Wei Yu, Xun Wang, Xinwen Fu, Dong Xuan, and Wei Zhao, “An invisible localization attack to Internet threat monitors,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 11, pp. 1611–1625, 2009, {doi: 10.1109/TPDS.2008.255, ISBN: 1045-9219 VO - 20}.

[YWWG011] Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y Zhao, and Yafei Dai, “Uncovering ocial network Sybils in the wild,” in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, (New York, NY, USA; November 2011), 2011, pp. 259–268, {doi: 10.1145/2068816.2068841, ISBN: 978-1-4503-1013-0}.

[YYZH008] Fasheng Yi, Shui Yu, Wanlei Zhou, Jing Hai, and Alessio Bonti, “Source-based filtering scheme against DDOS attacks,” *Int. J. Database Theory Appl.*, vol. 1, no. 1, pp. 9–20, 2008.

[YZDJ011] Shui Yu, Wanlei Zhou, Robin Doss, and Weijia Jia, “Traceback of DDoS Attacks Using Entropy Variations,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 3, pp. 412–425, 2011, {doi: 10.1109/TPDS.2010.97, ISBN: 1045-9219 VO - 22}.

[ZaJT013] Saman Taghavi Zargar, James Joshi, and David Tipper, “A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks,” *IEEE Commun. Surv. Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013, {doi: 10.1109/SURV.2013.031413.00127}.

[ZAJW017] Samman Zahra, Masoom Alam, Qaisar Javaid, Abdul Wahid, Nadeem Javaid, S. U. R. Malik, and Muhammad Khurram Khan, “Fog computing over IoT: A secure deployment and formal verification,” *IEEE Access*, no. 99, pp. 1–13, November 2017, {doi: 10.1109/ACCESS.2017.2766180}.

[ZGTG005] Cliff C. Zou, Weibo Gong, Don Towsley, and Lixin Gao, “The monitoring and early detection of Internet worms,” *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 961–974, October 2005, {doi: 10.1109/TNET.2005.857113, ISBN: 1063-6692 VO - 13}.

[ZhLK009] Chenfeng Vincent Zhou, Christopher Leckie, and Shanika Karunasekera, “Collaborative detection of fast flux phishing domains,” *J. Networks*, vol. 4, no. 1, pp. 75–84, February 2009, {doi: 10.4304/jnw.4.1.75-84, ISSN: 17962056}.

[ZhMa012] Cha Zhang and Yunqian Ma, Editors, *Ensemble Machine Learning*. Dordrecht, The Netherlands: Springer, 2012, 332 pp., {ISBN: 1-4899-8817-3}.

[ZhWC012] Yuli Zhang, Huaiyu Wu, and Lei Cheng, “Some new deformation formulas about variance and covariance,” in *2012 Proceedings of International Conference on Modelling, Identification and Control*, (Wuhan, Hubei, China; June 2012), 2012, pp. 987–992, {ISBN: 978-0-9567157-1-5}.

[ZJWX014] Wei Zhou, Weijia Jia, Sheng Wen, Yang Xiang, and Wanlei Zhou, “Detection and defense of application-layer DDoS attacks in backbone web traffic,” *Futur. Gener. Comput. Syst.*, vol. 38, pp. 36–46, September 2014, {doi: http://dx.doi.org/10.1016/j.future.2013.08.002, ISSN: 0167-739X}.

[ZKHC019] Fan Zhang, Hansaka Angel Dias Edirisinghe Kodituwakku, J. Wesley Hines, and Jamie Coble, “Multilayer Data-Driven Cyber-Attack Detection System for Industrial Control Systems Based on Network, System, and Process Data,” *IEEE Trans. Ind. Informatics*, vol. 15, no. 7, pp. 4362–4369, July 2019, {doi: 10.1109/TII.2019.2891261, ISSN: 1941-0050}.

[ZYLN016] Jing Zeng, Laurence T. Yang, Man Lin, Huansheng Ning, and Jianhua Ma, “A survey: Cyber-physical-social systems and their system-level design methodology,” *Futur. Gener. Comput. Syst.*, pp. 1–15, June 2016, {doi: 10.1016/j.future.2016.06.034, ISSN: 0167-739X}.

[ZYLS017] Jing Zeng, Laurence T. Yang, Man Lin, Zili Shao, and Dakai Zhu, “System-level design optimization for security-critical cyber-physical-social systems,” *ACM Trans. Embed. Comput. Syst. - Spec. Issue LCETES 2015, Spec. Issue ACSD 2015 Spec. Issue Embed. Devise Forensics Secur. TECS*, vol. 16, no. 2, April 2017.

APPENDIX A

DDOS ATTACKS: DETECTION AND SOFTWARE DEFINED NETWORKING

To defend against DDoS attacks, countermeasures consist of three components: *Detection* ([AlGo006], [ASCL006], [ChHw006], [KLCC006], [KoSV007], and [WaJS007]), *defense* ([AlGo006], [BrLe005], [ChPM007], [XiLS001] and [YaPS006]), and *IP trace back* ([Alji003], [SXLL008] and [YaPS005]). Detection of DDoS attacks is the most important step to further launch a combat against them. Methods documented in the literature for DDoS attacks detection include activity profiling ([FSBK003] and [MSBV006]), packet filtering ([EATB009] and [SoMA009]), sequential change-point detection ([ChHw006] and [WaJS007]), wavelet analysis ([BKPR002] and [TaCh011]), among others. The methods mentioned beforehand are based on specific features or fingerprints of DDoS attacks. Wanting to avoid detection, hackers manipulate network traffic in different forms: (i) Spoofing the source IP addresses of attack packets making source address distribution based detection algorithms ineffective ([DuYC008] and [YYZH008]), (ii) modifying the TTL value of attack packets disabling hop-count detection methods ([WaJS007] and [YYZH008]), or (iii) mimicking the behaviour of flash crowds ([ChHw006] and [CKBR006]) for suddenly increasing legitimate traffic to disguise attacks [Yu014].

The DDoS detection methods so far mentioned are based on specific features ([ASCL006], [KLCC006], [KoSV007], and [WaJS007]). Hence, these detection methods are passive and incapable of detecting new attacks [Yu014].

The entropy of attack flows is a method independent from specific attack features ([BhBK015] and [MaCh014)]. Relative entropy has been used to measure similarities between a known attack and suspected datasets [LeXi001]. Nevertheless, the relative entropy is not a perfect metric because of its asymmetrical property. Stochastic methods in the frequency domain and data mining techniques for DDoS detection have also been tested ([ChKT002], [LWFT007], and [YKPB013]). Traffic with DDoS attacks is mapped from the time domain to the frequency domain, and further transformed to the power spectral density for identification [ChKT002]. Data mining technology extracts the DDoS attack information [LWFT007].

A.1 Feature Based Detection Methods

A.1.1 Profile Based Detection

A strategy to disguise attack sources is IP spoofing. A **hop-count filter** is a method to fight against source IP spoofing. Hackers cannot falsify the number of hops an IP packet takes to reach its destination although any field in the IP header can be forged [WaJS007]. Receivers infer the hop-count based on the TTL field of the IP header. Internet servers can establish mapping tables of IP addresses with their related hop-counts, known as IP-to-hop-count (IP2HC), from legitimate clients. Defenders can consequently discriminate spoofed IPs from legitimate IPs. The detection rate in three cases has been considered: Single source, multiple sources, and multiple sources with an awareness of the detection method [Yu014].

In the *single source* case, hackers spoof the IP source address using an IP address of a legitimate client, C_i, of the victim. The client C_i usually submits n_i legitimate packets to the victim for a given time interval. The attacker injects N_a attack packets to the victim for the same time interval. It is expected that $N_a >> n_i$. The detection rate is

$$Z_s = 1 - \frac{n_i}{N_a} = 1 - \alpha_i \tag{A.1}$$

where α_i represents legitimate traffic.

For the *multiple source* case, hackers use n legitimate clients IP addresses, C_1, C_2, ..., C_n, for spoofing. Each client injects N_1, N_2, ..., N_n, packets to the victim from and legitimate traffic is defined as α_1, α_2, ..., α_n. The detection rate is

$$Z_m = \frac{\sum_{i=1}^{n}(1-\alpha_i)N_a}{\sum_{i=1}^{n} N_a} \tag{A.2}$$

For uniformly distributed spoofed packets among the n IP addresses, the previous equation becomes

$$Z_m = 1 - \frac{1}{n}\sum_{i=1}^{n}\alpha_i \tag{A.3}$$

When hackers are *aware of detection mechanisms*, but have no further information, an initial TTL with a range of $[h_m, h_n]$ is generated using a given distribution (*e.g.*, Gaussian). The probability of hop-count h_k is p_k for the chosen distribution. The legitimate traffic is defined as α_k in this case and the detection rate is

$$Z_{anti} = 1 - \sum_{k=m}^{n}\alpha_k p_k \tag{A.4}$$

From the defenders viewpoint, if hackers understand the victim better, they can obtain a lower detection rate [Yu014].

An alternative DDoS detection method is **packet score** [KLCC006], implemented at the victim end. By knowing the statistical distribution of legitimate packets, Bayes inference can be used to obtain the probability of incoming packets legitimacy. For this probability inference the following can be used: Packets attributes (A, B, ...), corresponding attributes value spaces

$\{a_1,\ a_2,\ ...\}$, $\{b_1,\ b_2,\ ...\}$, so on and so forth. The count for attack packets, normal packets, and measured packets are represented by N_a, N_n and, N_m, where the subscripts identify a given count correspondingly. For a given time interval, the following equation holds [Yu014]

$$N_m = N_a + N_n \tag{J.5}$$

The function $\zeta(\circ)$ denotes an accumulator and it is used to define N_a, N_n and, N_m as

$$\begin{aligned} N_a &= \sum_{i=1}^{\infty} \zeta_a\left(A = a_i\right) = \sum_{i=1}^{\infty} \zeta_a\left(B = b_i\right) = ... \\ N_n &= \sum_{i=1}^{\infty} \zeta_n\left(A = a_i\right) = \sum_{i=1}^{\infty} \zeta_n\left(B = b_i\right) = ... \\ N_m &= \sum_{i=1}^{\infty} \zeta_m\left(A = a_i\right) = \sum_{i=1}^{\infty} \zeta_m\left(B = b_i\right) = ... \end{aligned} \tag{A.6}$$

From this, the probability distributions for packets attributes in the attack, normal and measured cases can be derived as

$$\begin{aligned} &\Pr_a(A = a_i) = \frac{\zeta_a\left(A = a_i\right)}{N_a},\ \text{for } i = 1,\ 2,\ ... \\ &\Pr_a(B = b_i) = \frac{\zeta_a\left(B = b_i\right)}{N_a},\ \text{for } i = 1,\ 2,\ ... \end{aligned} \tag{A.7}$$

...,

$$\begin{aligned} &\Pr_n(A = a_i) = \frac{\zeta_n\left(A = a_i\right)}{N_n},\ \text{for } i = 1,\ 2,\ ... \\ &\Pr_n(B = b_i) = \frac{\zeta_n\left(B = b_i\right)}{N_n},\ \text{for } i = 1,\ 2,\ ... \end{aligned} \tag{A.8}$$

...,

$$\begin{aligned} &\Pr_m(A = a_i) = \frac{\zeta_m\left(A = a_i\right)}{N_m},\ \text{for } i = 1,\ 2,\ ... \\ &\Pr_m(B = b_i) = \frac{\zeta_m\left(B = b_i\right)}{N_m},\ \text{for } i = 1,\ 2,\ ... \end{aligned} \tag{A.9}$$

...,

The joint probability distribution among attributes for attack, normal and measured is

$$\begin{aligned} \Pr_a\left(A = a_i,\ B = b_j,\ ...\right) &= \frac{\zeta_a\left(A = a_i,\ B = b_j,\ ...\right)}{N_a} \\ \Pr_n\left(A = a_i,\ B = b_j,\ ...\right) &= \frac{\zeta_n\left(A = a_i,\ B = b_j,\ ...\right)}{N_n} \\ \Pr_m\left(A = a_i,\ B = b_j,\ ...\right) &= \frac{\zeta_m\left(A = a_i,\ B = b_j,\ ...\right)}{N_m} \end{aligned} \tag{A.10}$$

The *conditional legitimate probability* (CLP) for a packet *p* is defined by

$$CLP(p) = \Pr\left\{p = legitimate \mid p.A = a_p,\ p.B = b_p,\ ...\right\} \tag{A.11}$$

where $p.X$ denotes the attribute X of packet p.

Finally, the Bayes inference for calculating the *packet score* or CLP of packet p is

$$CLP(p) = \frac{\Pr\{p = legitimate\} \cap \Pr\{A = a_p,\ B = b_p,\ ...\}}{\Pr\{A = a_p,\ B = b_p,\ ...\}} \quad \text{(A.12)}$$

Packet discarding requires a threshold in the $CLP(p)$ value, which is not trivial. Adjusting this threshold dynamically based on the score distribution of recent incoming packets and the current level of system overload has been proposed previously in the literature [Yu014].

A.1.2 Low Rate DDoS Attack Detection

A DDoS attack featuring a low attack rate is also known as *shrew DDoS attack* and it is harder to detect it [KuKn003]. These attacks inherent a specific characteristic: Submitting attack packets periodically [Yu014].

Spectral analysis methods have been proposed to detect this kind of low rate attacks [ChHw006]. The theoretical foundation for this is herein introduced. For simplicity, a sequence of network traffic is denoted as $X = x[n]$ for $n = 1,\ 2,\ ...$. The *autocorrelation* of X is defined as

$$R_{XX}(m) = \frac{1}{N-m} \sum_{n=0}^{N-m+1} x[n]x[n+m] \quad \text{(A.13)}$$

The autocorrelation function can identify *periodicity* in the original signal. However, since it is a time domain function, and is not easy to identify the periodicity. Hence, the need to map it to the frequency domain through the *discrete Fourier transform* (DFT), where it is straightforward to identify periodicities since the content of a given signal is decomposed into its constituent frequencies. The autocorrelation DFT is defined by

$$DFT\left[R_{XX}(m), f\right] = \frac{1}{N} \sum_{n=0}^{N-1} R_{XX}(m) e^{\frac{-j2\pi fn}{N}} \quad \text{for } f = 0,\ 1,\ ...,\ N-1 \quad \text{(A.14)}$$

When having the output of the DFT, it is important to ensure that the *power spectrum density* (PSD) is available since it is a descriptor of the distribution of the power of a signal or time series in the frequency domain. Usually, the PSD shows that the energy of shrew attacks is concentrated in low frequency bands, while the energy of normal traffic could occupy the whole spectrum (in a *flatter* shape) under analysis [Yu014].

A.2 Network Traffic Based Detection

Network traffic is a powerful feature for DDoS detection at the network layer [KaCV014]. Distinct methods based on network traffic have been proposed in the literature: (i) *Mean quadratic distances* measuring traffic anomalies to distinguish DDoS traffic from flash crowds [SLOB007], (ii) signal processing and data mining technologies for extracting DDoS attack information [LWFT007], and (iii) wavelets methods and techniques ([LiLe003] and [VeAb999]), among others [Yu014].

Network traffic based DDoS detections are usually performed at LANs due to the anarchy nature of the Internet. System administrators manage and configure routers locally. Cooperation among routers can be achieved to detect possible attacks. Topology of LANs can be modelled as a graphs. Nevertheless, the aggregation feature of DDoS attacks makes the attack

paths to form a tree rooted at the victim computer system. The routers located at the edge of the LAN are known as *edge routers* [Yu014].

Flow is defined as all the passing packets, at a given router in a LAN, that share the same destination address. Multiple flows may coexist at a LAN router. For an on-going DDoS attack, there exists one flow addressed to the victim known as *attack flow*. Simultaneously, there are many other flows that are addressed to different destinations. Not as the source IP addresses or TTL values of attack packets, hackers cannot spoof the attack flow as the address of the victim is given. Henceforth, *flow based detection is independent of any specific attack features, and it can deal with new types of flooding attacks*. One of the fundamental metrics to measure flows for anomaly detection is *entropy* ([BhBK015], [CoTh005], [NaSP013], [ÖzBr015], [ReRL014], and [ZJWX014]). The Shannon's entropy of a discrete random variable X is defined as

$$H(X) = -\sum_{x \in \chi} \Pr\{X = x\} \log\left(\Pr\{X = x\}\right) \tag{A.15}$$

where χ is the sample space of X [CoTh005]. Entropy of a random variable is a measurement of its uncertainty ([CoTh005] and [Yu014]).

The entropy of flows at a given router is called *flow entropy*, which represents the randomness of the flows at the router. The flow entropy in a router is stable in non-DDoS attacks [JFKY013]. For on-going DDoS attacks, the attack flow dominates the traffic on LAN routers. Subsequently, the *flow entropy drops dramatically in a short time period* (*e.g.*, a few seconds) [Yu014].

The assumptions in network traffic based detection are: (i) The attack packets for a given attack come from one botnet (generated by the same attack tools), (ii) attack packets enter the LAN via a minimum of two edge routers and the attack flows merge at the junction routers, and (iii) the network system is linear and stable when the DDoS attack is on-going. Hereafter, the detection algorithm is running on all LAN routers. The edge routers monitor the network traffic using flow entropy, which in an attack free case remains in stable. Whilst an attack is present, *the flow entropy drops dramatically because there is one or more flows dominating the routers*. DDoS detection requires finding a suitable threshold Δ for the flow entropy decrease. Variation of flow entropy equal or greater than Δ is an indicator of a DDoS attack [Yu014].

Attacker uses a random variable X to control the generation speed of attack packets known as *attack rate* or *packet rate* of attack flow. For a constant speed of packets generation

$$\Pr\{X = C\} = 1 \text{ for } C \text{ constant} \tag{A.16}$$

If the number of attack packets is increased in time by shortening the *attacking time t*, the packets generation speed can be defined as

$$X = at + b \tag{A.17}$$

where a and b are constants [Yu014].

Different probability distributions have been used for modelling network traffic patterns (*e.g.*, the Poisson distribution)

$$\Pr\{X = k\} = \frac{\lambda^k e^{-\lambda}}{k!} \text{ for } k = 0,\ 1,\ ...,\ \text{and } \lambda \text{ constant} \tag{A.18}$$

A random variable X represents the packet rates of flows on a router for a given time interval. The vector $X = \{x_1, x_2, ..., x_n\}$ denotes the number of packets for n flows. The

probability distribution of the flows is

$$\Pr(x_i) = \frac{x_i}{\sum_{j=1}^{n} x_j} \text{ for } i = 1,\ 2,\ ...,\ \text{and } j = 1,\ 2,\ ... \quad \text{(A.19)}$$

From this, the flow entropy of flows is represented by $H_f(X)$ as

$$H_f(X) = -\sum_{i=1}^{n} p(x_i)\log(p(x_i)) \quad \text{(A.20)}$$

As previously discussed, $H_f(X)$ is stable in with minor fluctuations in normal network operations. Whereas, *in a DDoS attack the packet rate of the flow targeting the victim is significantly larger than packet rates of other legitimate flows at the same router*. Hence, $H_f(X)$ decreases dramatically. For a threshold Δ with a value $\Delta > 0$, the following inequality to identify DDoS flooding attacks can be used

$$H_f(X)\Big|_{t=t_0} - H_f(X)\Big|_{t=t_0+\Delta t} \geq \Delta \quad \text{(A.21)}$$

where *t* and Δt represents time and a short time interval respectively [Yu014].

For $H_f(X)$ differentiable at t_0 , the previous inequality becomes

$$H_f^{'}(X) = \lim_{t_0 \to 0} \frac{H_f(X)\Big|_{t=t_0+\Delta t} - H_f(X)\Big|_{t=t_0}}{\Delta_t} \quad \text{(A.22)}$$

Combining the previous two equations it is shown that $H_f^{'}(X) \leq -\Delta$ [Yu014].

A.3 Detection Against Mimicking Attacks

Distinction of DDoS attacks mimicking flash crowds from genuine flash crowds is important ([BSMT014], [ChHw006], [CKBR006], and [JuKR002]). Failure in achieving this allows attackers mimicking flash crowds traffic features for avoiding detection. Moreover, detectors may treat legitimate flash crowds as DDoS attacks. Distinguishing flash crowds from DoS attacks has been addressed by using features like traffic patterns, client characteristics, and file reference characteristics [JuKR002]. However, hackers disable the detector easily by mimicking the flash crowds network traffic. Entropy detectors can raise alarms of crowd access but have difficulty distinguishing DDoS attacks from legitimate accesses of flash crowds [FSBK003]. Distinction of flash crowds from DDoS flooding has been tried through the *change-point detection method* [ChHw006]. User browsing dynamics (*e.g.*, number of requests for a given time interval) for differentiating flash crowds from DDoS attacks has also been implemented ([XiYu009a] and [XiYu009b]). The web pages of a given web site follow the Zipf distribution, but not so the DDoS attack requests. In current botnets the following features prevail: (i) Attack tools are prebuilt programs, which are usually the same for one botnet, (ii) a botmaster issues a command to all bots in a botnet to start one attack session, (iii) attack flows observed at the victim end are an aggregation multiple attack flows, (iv) aggregated attack and original attack flows have similar standard deviation, (v) the flow standard deviation is usually smaller than that of genuine flash crowd flows. The reason for this phenomenon is that the number of live bots of a current botnet is far less (usually hundreds or a few thousands) than the number of concurrent legitimate users of a flash crowd (usually hundreds of thousands). Hence, launching a flash crowd attack requires the botmaster forcing live bots to generate many more

attack packets (*e.g.*, web page requests) than that of a legitimate user [Yu014].

A.3.1 Metrics Similarity

Metrics for similarity measurement includes first order (*e.g.*, mean, Kullback-Leibler divergence, *Jeffrey distance, Sibson distance, Hellinger distance*) and second order metrics (*e.g.*, correlation, standard deviation, and correntropy) [Yu014].

For the case of *first order metrics*, two flows or sequences P and Q have probability distribution $p(x)$ and $q(x)$, respectively. Metrics measure distance or similarity between them. The Kullback-Leibler (KL) distance, not a metric in a rigorous sense, is defined as

$$D(p,q)=\sum_{x\in\chi} p(x)\cdot\log\left(\frac{p(x)}{q(x)}\right) \tag{A.23}$$

where χ is the sample space of x. It shall be noticed that $D(p,q)\neq D(q,p)$ for $p\neq q$. This asymmetry is corrected by the *Jeffrey distance* combining KL distances in

$$D_J(p,q)=\frac{1}{2}\left[D(p,q)+D(q,p)\right] \tag{A.24}$$

The *Sibson distance* defined as

$$D_S(p,q)=\frac{1}{2}\left[D\left(p,\frac{1}{2}(p+q)\right)+D\left(q,\frac{1}{2}(p+q)\right)\right] \tag{A.25}$$

The *Hellinger distance* is defined as

$$D_H(p,q)=\sqrt{\sum_{x\in\chi}\left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2} \tag{A.26}$$

Among the information theoretic based first order metrics, the Sibson distance is the best for DDoS attack detection [YTLW009]. Abstract distances do not include time information and are sensitive to fluctuation of flows [Yu014].

For *second order metrics*, the *correlation* between two flows X_i and X_j for $i\neq j$ with the same length N is defined as

$$r_{X_i,X_j}=\frac{1}{N}\sum_{n=1}^{N} x_i[n]x_j[n] \tag{A.27}$$

Correlation describes similarity between flows [XQQL015]. In some cases, its value may be zero for two completely correlated flows having a phase difference. The previous definition is then modified to

$$r_{X_i,X_j}[k]=\frac{1}{N}\sum_{n=1}^{N} x_i[n]x_j[n+k] \tag{A.28}$$

where $k=0,\ 1,\ ...,\ N-1$ is the shift of flow X_j. Magnitude differences might be possible for different sets of similar flows. Thus, unification is required through the correlation coefficient

$$\rho_{X_i,X_j}[k]=\frac{r_{X_i,X_j}[k]}{\frac{1}{N}\sqrt{\sum_{n=1}^{N} x_i^2[n]\sum_{n=1}^{N} x_j^2[n]}} \tag{A.29}$$

Correlation coefficient is both used as similarity metric in network flow applications ([ChZa005], [SWWJ008], and [YASR008]) and better than abstract distances in terms of stability.

Correntropy is a recently invented, under a clear theoretical foundation, local tool for second-order similarity measurement in statistics. Correntropy is used in various disciplines (*e.g.*, face recognition [HeZH011]). Correntropy is symmetric, positive, and bounded. For two finite data sequences A and B, the similarity of the sample sequences $\{(A_j, B_j)\}_{j=1}^{m}$ where $m \in \frown$ is estimated as

$$\hat{V}_{m,\sigma}(A,B) = \frac{1}{m}\sum_{j=1}^{m} k_\sigma\left(A_j - B_j\right) \tag{A.30}$$

where $k_\sigma(\circ)$ is the Gaussian kernel defined as

$$g(x) \triangleq \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{A.31}$$

A.3.2 Flow Correlation Based Discrimination

Most attacks detections are conducted at the victim end, also known as community network. The number of packets for a network flow is sampled within a time interval. A network flow is represented by a data sequence $X_i[n]$, where $i \geq 1$ represents the network flows, and n denotes the n^{th} element in the data sequence. If the length of a given network flow X_i is N, then the network flow can be presented as

$$X_i[n] = \{x_i[1],\ x_i[2],\ ...,\ x_i[N]\} \tag{A.32}$$

where $x_i[n](1 \leq n \leq N)$ represents the packets count in the n^{th} time interval for the network flow.

The expectation of a data sequence X_i can be defined as

$$E[X_i] = \frac{1}{N}\sum_{n=1}^{N} x_i[n] \tag{A.33}$$

which represents the *flow strength*. The flow strength is the average packet rate of a network flow. If X_i is a DDoS attack flow, then $E[X_i]$ is known as *attack strength*.

The *fingerprint of flow* X_i denoted by X'_i is the flow unified representation and is defined by

$$\begin{aligned} X'_i &= \{x_i'[1],\ x_i'[2],\ ...,\ x_i'[N]\} \\ X'_i &= \left\{\frac{x_i[1]}{N\cdot E[X_i]},\ \frac{x_i[2]}{N\cdot E[X_i]},\ ...,\ \frac{x_i[N]}{N\cdot E[X_i]}\right\} \end{aligned} \tag{A.34}$$

From this it can be shown that $\sum_{k=1}^{N} x_i'[k] = 1$, which means that *fingerprint of flow* is an instance of its probability density distribution. The relationship between a network flow and its fingerprint is $X_i = N \cdot E[X_i] \cdot X_i'$ [Yu014].

Current botnets (*e.g.*, SDbot, Rbot and Spybot) employ the same program to generate attack packets creating as many as they can, usually with a very short delay (1 or 5 ms) between two attack packets. This confirms that flow fingerprint exists in attack flows for a given botnet

[Yu014].

When a DDoS attack alarm goes off, the routers in the community network sample the suspected flows by counting the number of packets for a given time interval (*e.g.*, 100 ms). When the length of a flow N is sufficient, the correlation coefficient between suspected flows is obtained. In DDoS attacks, the suspicious network flows have a strong correlation (although they are a mixture of a number of original attack flows with different delays). The correlation coefficient value of two flows of flash crowds is smaller compared to two attack flows [Yu014].

Sampling M network flows, $X_1, X_2, ..., X_M$, allows calculating the correlation coefficient between two of such flows, $X_i (1 \le i \le M)$ and $X_j (1 \le i \le M,\ i \ne j)$. A function indicating whether a DDoS attack is present (denoted by a value of 1) or not (denoted by a value of 0) is defined by I_{X_i,X_j}, where the flows X_i and X_j are compared through the correlation coefficient. The indication function I_{X_i,X_j} has discrimination threshold defined by δ. Therefore, the indication function can be defined as [Yu014]

$$I_{X_i,X_j} = \begin{cases} 1, \ \rho_{X_i,X_j} \ge \delta \\ 0, \ \text{otherwise} \end{cases} \tag{A.35}$$

A.3.3 System Analysis on the Discrimination Method

Theoretical analysis of the previous method effectiveness and proof of the existence of the threshold δ have been carried. Also, the relationship between the correlation coefficient of flows and their length has been explored. The attack flow convergence is addressed considering the following assumptions: (i) Only one server in a community network is under attack or experiencing flash crowds at any given time, (ii) attack packets enter the community network via a minimum of two different edge routers, (iii) attack packets are generated by only one botnet, hence the attack flows fingerprints are the same, and (iv) network delays are discrete and countable [Yu014].

A flash crowd with known statistics (*e.g.*, mean defined as $n\mu$) to everyone, including attackers, helps a botmaster to use n bots (usually at hundreds or thousands level [RZMT007]) to execute a flash crowd attack. However, a big number of users (*e.g.*, hundreds of thousands of browsers) for generating flash crowds is required. Hence, a botmaster must exhaust bots for generating attack traffic with μ mean per bot. This causes a very small timer interval of the attack packets for injecting sufficient attacking packets and results a very small standard variation of packet arrivals (*e.g.*, $\sigma = 0.01\mu$ or towards the best $\sigma = 0$ from hackers view). Once the attack traffic is aggregated the mean of the flash crowd $n\mu$ is obtained, but not the standard deviation. The reasons for this are that the flash crowd traffic is: (i) Created by many more computer systems than a botnet and (ii) it has genuine distribution with a standard variation larger than the attack traffic. DDoS attack traffic and flash crowd traffic can be distinguished through the correlation coefficient. The fingerprint of flash crowds flow is the Pareto distribution ([CrBe997] and [PaFl995]), which is defined by the probability distribution function

$$\Pr\{X = x\} = \alpha_P \cdot x_m^{\alpha_P} \cdot x^{-(\alpha_P + 1)} \tag{A.36}$$

where $x_m \le x$, and α_P is the *Pareto index*. Additionally, for any two independent flash crowd

flows with length N, the flow correlation coefficient tends to 0 when N tends to infinity.

A DDoS attack flow X_i obtained at an edge router is usually a mixture of attack flows that came from K different bots. The fingerprint of the attack flow X_i can be represented by $X_0^{'}$, which is usually the same in one attack session with delays in different attack flows. Delayed fingerprints by j time units can then be represented by $X'_0[j]$. Hence, the observed attack flow can be denoted as

$$X_i = \sum_{j=0}^{K} N \cdot E\left[X_i\right] \cdot X_0^{'}[j]$$

$$X_i = \sum_{j=0}^{k^{'}} a_j \cdot X_0^{'}[j] \quad \text{(A.37)}$$

where $a_j (1 \le j \le k^{'} \le K)$ represents attack flows magnitude with the delay j at the edge router. Considering no network delay and no background noise in two mixed attack flows X_i and $X_j (i \ne j)$ observed at two edge routers, their correlation coefficient $\rho_{X_i,X_j}[k] = 1$. Hence, two DDoS attack flows from one botnet are totally correlated. It is important to consider that *noise* and *delays* among the attack flows from different bots exist. These attack flows features depend on the *legitimate packets* addressed to the victim and *normal Internet delays* respectively. Delays are limited when compared with fast Internet facilities. From the hackers' perspective, all legitimate traffic sent from users to the victim is considered noise at the time of an on-going DDoS attack. Usually, the strength of the noise is much smaller compared with the strength of DDoS flooding attack flows [Yu014].

Defining two noise flows as Y_i and Y_j for two DDoS attack flows X_i and X_j in an attack session, $\forall \delta (\delta < 1), \exists \Delta_H, \rho_{X_i,X_j}[k] > \delta$ (where δ is the discrimination threshold and Δ_H is the decrease of flow entropy threshold) holds when $E[X_i]/E[Y_i] > \Delta_H$ and $E[X_j]/E[Y_j] > \Delta_H$. The correlation attack flows $\rho_{X_i,X_j}[k] = 1$ when they are in a noiseless environment defined by $E[Y_i] = E[Y_j] = 0$ [Yu014].

If the noise strength is much superior than the signal, $E[Y_i]$ $E[X_i]$ and $E[Y_j]$ $E[X_j]$, then the noise flows can be represented by

$$\begin{cases} Y_i \approx Y_i + X_i \\ Y_j \approx Y_j + X_j \end{cases} \quad \text{(A.38)}$$

and consequently $\rho_{X_i+Y_i,X_j+Y_j}[k] \approx \rho_{Y_i,Y_j}[k]$. It is also known that $\rho_{Y_i,Y_j}[k] \to 0$ and when the length N increases [Yu014].

The correlation coefficient of DDoS attack flows approaches 1 if the SNR $E[X_i]/E[Y_i]$, is large. For an on-going DDoS attack, $E[X_i]$ $E[Y_i]$ and $E[X_j]$ $E[Y_j]$ and the correlation coefficient of attack flows is close to 1 [Yu014].

DDoS attack flow can be discriminated from flash crowds by flow correlation coefficient at edge routers under two conditions: The length of the sampled flow is sufficiently large, and

the DDoS flooding attack strength is sufficiently strong. Defining X_p and $X_q (p \neq q)$ as two random flash crowds and X_i and $X_j (i \neq j)$ two DDoS flooding attack flows, and the discrimination threshold $\delta (\delta < 1)$ is a given small real number. Based on condition N the following equation holds

$$\Pr\left\{ \rho_{X_p, X_q}[k] < \delta \middle| N \right\} = 1 \tag{A.39}$$

which means that the correlation of the two flash crowds is under the threshold that triggers an DDoS alarm. Based on condition *N* and SNR the following equation holds

$$\Pr\left\{ \rho_{X_i, X_j}[k] \geq \delta \middle| N, SNR \right\} = 1 \tag{A.40}$$

which means that the correlation of two DDoS attack flows is above the threshold that flags a DDoS attack [Yu014].

It is known that $\rho_{X_p, X_q}[k]$ is a decrease function on N the length of flow; $\rho_{X_p, X_q}[k] = 1$ for no noise and no delay, and it decreases when the strength of noise increases, therefore, there must exist a value δ where the previous probability equations hold [Yu014]. The case of interest for identifying DDoS attacks is the flow correlation coefficient being greater than δ for a given pair of two flows.

In a DDoS attack or flash crowds, a number of suspected flows M is available. Calculating the flow correlation coefficient I_{X_i, X_j} for any two different flows X_i and $X_j (1 \leq i, j \leq M, i \neq j)$ leads to an integrated DDoS attack positive probability defined by

$$\Pr\{I_A = 1\} = \frac{\sum_{1 \leq i, j \leq M, i \neq j} I_{X_i, X_j}}{\binom{M}{2}} \tag{A.41}$$

where I_A is the indicator for DDoS attacks, and $I_A = 1$ represents positive for DDoS attacks. The final decision with global information is made as follows

$$I_A = \begin{cases} 1, & \Pr\{I_A = 1\} \geq 0.5 \\ 0, & \Pr\{I_A = 1\} \leq 0.5 \end{cases} \tag{A.42}$$

which identifies a DDoS attack if at least half of the comparisons are positive [Yu014].

A.4 DDoS Attacks in Software Defined Networking

Avoiding security threats is the ultimate purpose of DDoS attack detection in both conventional and SDN approaches. The latter separates the network's control logic (control plane) from the underlying routers and switches forwarding the traffic (data plane). One of the most notable architectures for SDN is OpenFlow. Major commercial players (*e.g.*, Google, Yahoo, Rackspace, and Microsoft) are conducting research and experimentation in SDN towards strengthening network security. Two of the most significant attack vectors to SDN are intrusion and DDoS attacks. DDoS attacks usually occur as either forged or faked traffic flows in the data plane. When the source addresses of spoofed packets, the switch cannot match this spoofed packet and forwards it to the controller. Both legitimate and DDoS spoofed packets force the

resources of the controller to continuously process these packets resulting in exhaustion of resources [AsLa014].

APPENDIX B

DIVERSITY OF COMPUTING SYSTEMS IN THE CYBERSECURITY ECOSYSTEM

The European Union (EU) sponsored project, CyPHERS, listed the technology fields of embedded systems, mechatronics, IoT, big data, and systems of systems (SoS) as forming the landscape for cyber-physical systems (CPS) [Hens016].

The IoT, IIoT, CPS, CPSS and SoS describe super-systems, but while the communities of interest in these super-systems intersect, they do not completely overlap. The selected definitions below have some level of consensus within the respective communities [Hens016]. An important definition to consider as a super-systems are cyber-physical-social systems (CPSS).

B.1 Complex Systems

Computers, thunderstorms, the human brain, and corporations, are examples of complex systems made up of parts interacting with one another in specific ways. Outcomes of these interactions are hard to predict. The opportunity for a system to exhibit complex behaviour must be greater when there are more parts. Complex behaviour arousal does not require complex parts. When systems have lots of interacting parts even simple parts and simple interaction rules are sometimes sufficient to produce complex structure and behaviour. One of the principles promoted by Kauffman is summarized by the phrase “edge of chaos”. Large networks tend to be characterized by large domains of parameter space that exhibit chaotic behaviour and other large domains where frozen unchanging behaviours dominate; but at the interface, between frozen and chaotic domains, behaviour can become complex. Nature is full of situations where systems of simple parts and simple interactions self-create order and exhibit a surprising range of behaviours; but in the realms of life and human activity, complex systems exist (*e.g.,* electrical, electronic, biochemical, genetic networks, protein-protein interfaces, and life itself). A living organism provides a prime example of a *complex adaptive system* (CAS). Other examples are a national economy, ant colonies, ecosystems, and the human brain. All are characterized by a degree of stability when confronted with changing external events. Concepts that are important in the study of CAS are emergence, self-organization, adaptation, homeostasis, communication, and cooperation. All CAS can be viewed and studied as networks, but not all networks are adaptive. What sets CAS apart from other complex systems are nodes with memory. Memory allows responses that take past events into account. CAS are characterized by nodes (agents) that

make decisions based on past actions and on new input. All known CAS are characterized by nodes and edges that are prespecified, and most, if not all, can be described as a network with regulatory edges. Without careful prespecification, any particular combination of nodes and edges that behave like CAS is simply too improbable to occur spontaneously. proteins and their interactions in the life network, neurons in the brain, organisms in the biosphere, and traders in the stock market provide a few examples of prespecified nodes with memory [Mayf013]. The reader can get a sense that protecting complex systems, in the CPSS context, is a very serious and challenging task. Hence, the need for designing very fine cybersecurity systems.

B.2 Systems of Systems

As early as 1971, Russell Ackoff had used the term **systems of systems** to describe combinations of individual systems that were bound together in some way, however, this was in the context of management science and related to enterprises, rather than the SoS more appreciated by systems engineers. Recently, Brook attempted to provide a general description applicable to all types of SoS, as follows: "A SoS is a system which results from the coupling of a number of constituent systems at some point in their life cycles." The implications of this are that the constituent systems may, or may not have been designed to work together, but that through some means they are coupled so that they interoperate [Hens016]. Formally, tightly coupled systems are known as multi systems, while loosely coupled systems are known as multiple systems [Kins017].

Future industrial infrastructures are expected to be complex SoS that empowering a new generation of applications and services. An example of this is the factory of the future (FoF) relying on an ecosystem of SoS where collaboration at large scale would take place. The application of the service-oriented architecture (SOA) paradigm to bring the shop-floor to cyberspace exposing capabilities and functionalities as services located on physical resources undertaking society's needs [CBKD014].

Large process industry systems are a complex (potentially very large) set of frequently multidisciplinary, connected, heterogeneous systems that function as a complex distributed system whose overall properties are greater than the sum of its parts (*e.g.*, very large-scale integrated devices) and systems whose components are themselves systems (*e.g.*, systems-on-a-chip). Industry systems link many component systems of a wide variety of scales, from individual groups of sensors to whole control, monitoring, supervisory control systems, performing SCADA and distributed control systems (DCS) functions. The resulting combined systems are able to address problems which the individual components alone would be unable to do and to yield control and automation functionality that is only present as a result of the creation of new, 'emergent', information sources, and results of composition, aggregation of existing and emergent feature- and model-based monitoring indexes. These very large-scale distributed process automation systems constitute SoS and are required to meet Maier's criteria: (i) Operational independence of the constituent systems, (ii) managerial independence of the constituent systems, (iii) geographical distribution of the constituent systems, (iv) evolutionary development, and (v) emergent behaviour. Such systems should be based on process control algorithms, architectures and platforms that are scalable and modular (plug and play) and applicable across several sectors, going far beyond what current SCADA and DCS systems and devices can deliver today [CBKD014].

B.3 Cyber-Physical-Social Systems

The emergence of **cyber-physical-social systems** has revolutionized the relationship between computers, the physical environment, and humans (social domain). CPSS considers social characteristics and relations [GuYZ015], human knowledge, mental capabilities, and sociocultural elements. Information from cyberspace interacts with physical and mental spaces in the real world, as well as the artificial space mapping different facets of the real world. A CPSS carries out parallel execution, self-synchronization, and influences in the physical, information, cognitive, and social domains. These four domains are key elements in military C&C systems. By fusing physical, information, cognitive, and social domains, CPSS transform command and control organizations. To a degree, the nimbleness of a command and control organization stems from the fusion of its internal essential components, especially the human factors involved. Thus, CPSS provide an ideal paradigm for designing and constructing command and control organizations. Information technology has changed command and control organizations, flattened their structures, and formed more dynamic and complex interactions. Such organizations have become organic entities, including sensor, enabler, communication, and social networks composed of human beings. The components are closely linked, enabling information collection, situation awareness, planning and decision-making, and action execution within a loop. With the support of cognitive computing, a command and control organization is able to model its own conduct leading to human centricity and synchronicity. The study of network-centric warfare shows that an efficient military organization must be robust, resilient, responsive, flexible, innovative, and adaptive. These attributes form the basis of self-synchronic military operations conducted by a command and control organization. To effectively carry out command and control, the main challenge lies in the organic integration of multidomains— that is, the physical systems in the physical domain, cyber networks in the information domain, mental elements in the cognitive domain, and social networks in the social domain. Transregional and transdomain self-synchronization leads to the complexity and emergence of system operations. Because there are trade-offs between response time and decision quality, chaotic control might be the key to balance them under critical situations. Hence, to achieve self-synchronization of a command and control organization, establish a chaotic control mechanism for the CPSS is required. The inputs are missions, events, or tasks. The CPSS automatically integrates an organization's essential elements in the four domains, assigns physical resources, sets up sensor networks and enabler networks, constructs the command and control relationships in the social network, and organizes and shares relevant information as needed. The chaotic system's output is the complete operational system; the entire command and control organization as a CPSS. In the operational process, a distributed command and control organization is both self-organizing ([SLSS014] and [SmKP015]) and self-adaptive in accordance with changes in the battlefield. Instead of pursuing an absolute central control of the organization behaviour, a chaotic control mechanism focuses on the harmony of the CPSS as a SoS. A framework for self-synchronization of a CPSS is built upon self-organizing and self-adapting processes. In this context, a CPSS is made up of physical network, cyberspace, cognitive network, organizational network, and artificial societies. The physical network includes sensor, enabler, and communication networks, closing the sensor-to-shoot cycle. Cyberspace supplies space for computing, storage, information processing, and sharing services, especially situation awareness and decision-support services directly for command and control. The cognitive network is a

semantic Web based distributed knowledge management system, with the goal of managing, evolving, and utilizing operational rules and organization intelligence. The organizational network is a social network consisting of commanders and the operating personnel that play a key role in the command and control relationship. Artificial societies are introduced as a virtual world mapping of the physical network, cyberspace, cognitive network, and organizational network in the real world, as well as multiple future virtual worlds for exploration of possible scenarios. The physical network, cyberspace, cognitive network, and organizational network are integrated and overlapping [LYWZ011].

In particular, human social interaction exerts fundamental influence on cyberspace (*e.g.*, using smartphones as sensors in CPSS social relationship is used to assist message forwarding). CPSS establishes an effective connection between human intelligence and machine intelligence. Another modality about CPSS is the social network aspect over CPS. Connecting the social space, cyberspace and physical space via social network is a promising approach. Prototypes towards cyber-physical social networks have already been proposed modelling persons and communities with abstract methods [ZYLN016].

CPSS fuses diverse information originating from cyberspace, physical-space and social-space, providing human-centric computation services. The main task of CPSS is to meet people's social interaction demands and to react to the physical world. Hence, the research focuses on developing CPSS technologies (*e.g.*, smart home, smart transport systems, smart medical services, and smart cities) to support cumbersome tasks in cyber world, physical world, and social world to facilitate work efficiency. The inclusion of the human role in CPSS is fundamentally different from traditional CPS. Human's social activities in CPSS relate to both cyberspace and physical space closely [ZYLN016].

CPSS research topics are divided into: (i) Seamless migration technologies of heterogeneous network, (ii) device management and discovery, (iii) context awareness and management, (iv) human–computer interaction, (v) user behaviour based proactive service, (vi) social computing, and (vii) security and privacy. They can also be partitioned by related space. For cyberspace and physical space, CPSS consists of: (i) Seamless migration technologies of heterogeneous network, (ii) device management and discovery, (iii) human–computer interaction, and (iv) security and privacy. Regarding cyberspace and social space, CPSS refers to: (i) Context awareness and management, (ii) social computing, and (iii) user behaviour based proactive service [ZYLN016].

According to Festo Head of Future Technology, Christoph Hanisch, engineers have to find their way into the right use of technology where the outlook of the possibilities for the integration of mechanics, electronics, and software puts them in a unique historic position as the options for matching virtual and real worlds have never been so close before. Today, engineers have the chance to develop mechanical architectures supported by software tools and control power that enabling users to achieve systems that were non-existent before [CBKD014].

An important aspect in CPSS are the protocols allowing connectivity to the Internet. The OSI model standardizes communication functions and protocols of computing systems by abstracting and locating them into layers: Physical (bits), data link (frames), network (packets and routing (*e.g.*, IP)), transport (segments (*e.g.*, TCP)), session (dynamic creation of network connections), presentation (message data syntax processing (*e.g.*, format conversions, encryption/decryption for supporting the application layer), and application (network services, or

protocols (*e.g.*, hyper-text transfer protocol (HTTP)), for end-user applications). [XiRa015].

The complexity of the data link layer causes it to split into: *Medium access control* (MAC) sublayer and logical link control sublayer. MAC is particularly relevant in CPSS because it is supported by the physical layer. MAC is responsible for a number of functions like addressing and channel access controlling mechanism. For multiple nodes in a network to communicate through shared medium, MAC provides channel access control mechanism known as multiple access protocol. For short-range wireless communications in CPSS (*e.g.*, *wireless sensor networks* (WSNs), *wireless sensor and actuator networks* (WSANs), and *wireless body area networks* (WBANs)), MAC protocols often use *time division multiple access* (TDMA), where nodes access a medium in time slots, or *carrier sense multiple access with collision avoidance* (CSMA/CA), where nodes transmit when the channel is sensed as "idle", for fair access of shared medium. Other communication protocols as *frequency division multiple access* (FDMA) and *code division multiple access* (CDMA) could also be used [XiRa015].

The Institute of Electrical and Electronic Engineers, IEEE, has defined a family of standards, IEEE 802, for networking protocols. The IEEE 802 family has included descriptions since February 1980 until now. Important descriptions interfacing CPSS are IEEE 802.3 (Ethernet), IEEE 802.11 (wireless local area networks and Wi-Fi), IEEE 802.15.1 (Bluetooth), IEEE 802.15.4 (ZigBee), IEEE 802.15.6 (body area networks), IEEE 802.15.7 (visible light communications), and IEEE 802.24 (Smart Grid). Many CPSS are supported by the IEEE 802 family like WSANs built upon the IEEE 802.15.4 (*e.g.*, large-scale factory automation, distributed and process control, machinery health monitoring). These standards have greatly encouraged to bridge real-time (with meticulous and constrained implementations) physical world applications/objects to cyber world for diverse range of time-critical applications. The communication and computing capabilities of a cyber core are utilized to control/monitor real world's objects/applications [XiRa015].

Medical-cyber-physical-social systems (MCPSS), a specialized form of CPSS, is an integration of sensing, computation, communications, and medical processes, which can provide reliable and real-time services. Context-aware, life-critical, and networked medical devices are used to provide continuous high-quality healthcare for patients within or outside the hospital. MCPSS can be classified into: Invasive (in-body sensor nodes monitor physiological signs) and non-invasive (on-body sensor nodes are used [XiRa015].

Smart cities define urban environments with a new generation of innovative services for transportation, energy distribution, healthcare, environmental monitoring, business, commerce, emergency response, and social activities. The technological infrastructure of a smart city is based on a network of sensors and actuators embedded throughout the urban terrain, interacting with wireless mobile devices (*e.g.*, smartphones) and having an Internet-based backbone with cloud service. The data collected and flowing through such CPSS involve traffic conditions, occupancy of parking spaces, air/water quality information, structural health of bridges, roads, or buildings, and location and status of city resources (*e.g.*, transportation vehicles, police officers, or healthcare facilities). Enabling smart cities requires a cyber-physical-social infrastructure combined with new software platforms and strict requirements for mobility, security, safety, privacy, and the processing of massive amounts of data. Data reside and drive a variety of novel and continuously evolving applications with real-time response and stringent security expectations. Smart cities infrastructure consists of sensing, communicating, decision-making,

and actuating [Cass016].

Messages of CPSS are exposed to malicious agents by snooping packets in the communication network. Although security is a fundamental issue for CPSS, most devices in CPSS are resource constrained, where energy efficiency has a direct impact on the performance of security services (*e.g.*, the higher the encryption strength, the more energy the algorithm consumes) [ZYLS017].

B.3.1 Cyber-Physical Systems

It seems Helen Gill of the US National Science Foundation coined the term, **cyber-physical systems** in 2006. As the name suggests, CPS have both cyber (software control) and physical (mechanism) elements; more specifically they are physical entities controlled by computer algorithms. In general, whilst they clearly encompass the field of embedded systems, they go beyond this in terms of complexity because of the importance of computer networking to their capabilities. Certainly, a key attribute of CPS seems to be an expectation that they are a disruptive technology through enabling new business models. The features of CPS are the ability to sense, contextualise, decide, and act in such a way that non-human decision-making agents interact with (change and are changed by) the environment. The following definition of CPS is probably one of the most complete available: "CPS are systems with embedded software (as part of devices, buildings, means of transport, transport routes, production systems, medical processes, logistic processes, coordination processes, and management processes), which: (i) Directly record physical data using *sensors* and affect physical processes using *actuators*; (ii) evaluate and save recorded data, and *actively* or *reactively interact* both with the *physical* and *digital world*; (iii) are connected with one another and in global networks via digital communication facilities (wireless and/or wired, local and/or global); (iv) use globally available data and services; and (v) have a series of dedicated, multimodal human-machine interfaces" [Hens016].

Industrialists, researchers and practitioners are associating advances in computation and communication resources with a fourth industrial revolution (referred to as Industry 4.0 in Germany) where physical "things" get connected to the Internet allowing the real touchable world to integrate part of cyber-space. The world market of monitoring and control (M&C), fundamental in industrial development, is expected to grow reaching € 500 billion by 2020. When analysing the major application domains for real-time M&C from the large process industry viewpoint, these index and related expectations outline the tremendous potential and value [CBKD014].

According to the characteristic of mobility of embedded computers, CPS can be partitioned into: (i) Accompanied (neither wearable nor implanted), (ii) portable (two-handed operation), (iii) hand-held, (iv) wearable (hands-free), and (v) implanted or embedded (*e.g.*, medical services to assist humans) [ZYLN016]. In CPS, numerous embedded devices with limited computational, communication, sensing capacities, and power supply are networked, enabling a variety of innovative applications. In industrial applications, a number of sensor nodes are attached to machines to communicate machines' status to a computation core that sends the feedback control/command to actuators attached to machines. Applications of CPS include: Medical healthcare, home automation, environmental control, assisted living, smart city, transportation, traffic control, process control, automotive systems, defense systems, water

supply, smart grid, robotics, smart spaces, energy conservation, smart factory, industrial automation, battlefield surveillance, communication systems, and aerospace systems, to mention a few [XiRa015].

Characteristics of CPS include cyber capability in physical components, networking at multiple and extreme scales, complexity at multiple temporal and spatial scales, dynamic reorganization and reconfiguration, high degrees of automation, closed control loops at multiple scales, unconventional computational and physical substrates, and dependable even certifiable operations. CPS allow connectivity to control mechanical systems using embedded processors. The environmental states are sensed and controlled using sensors and actuators for a diverse range of applications. CPS provide interaction to physical world which must be safe, secure, efficient, and dependable. This intimate coupling of cyber and physical worlds can vary in both scale and size (*e.g.*, smartphones to smart industrial applications). In CPS, the physical world's processes and the cyber computation world are bridged with feedback loops, where the computation processes affect physical processes, and vice versa. A typical CPS consists of physical objects/applications, actuators, sensors, communication, and computing core. This coupling provides solutions for different applications, not only on large scale but also for personal use and at microlevel. Like transformation of human interaction via Internet, CPS enable the physical world to interact with the cyberworld, thus transforming how human beings interact with the physical world [XiRa015].

B.3.2 Cyber-Social Systems

Cyber-social systems (CySS) focuses on using people social behaviours and relationships analysis to provide suitable information services that greatly promote the quality life of people. Social network is an important application of CySS and has gained increased attention (*e.g.*, typical social network applications, such as and not limited to Facebook, Twitter, or YouTube). Multiple social network based applications exist: (i) Recommendation based on social network where research groups employ social networks to enable automatic recommendation systems providing similar preferences between various social individuals, (ii) electronic negotiation where autonomous agents model automatic negotiation in the electronic market (*e.g.*, auctions, stock market, or forex), (iii) public sector, (iv) *e*-government, and (v) interactive entertainment [ZYLN016].

B.4 Internet of Things

The **Internet-of-things** is a concept introduced in 1998 by Kevin Ashton [Path015] and it got its start in 1999 with the founding of the MIT Auto-ID Center having as a goal the development of a broad class of identification technologies for use in industry to support automation, reduce errors, and increase efficiency. The cornerstone of this technology was the radio frequency identification (RFID) tag, which allows to uniquely identify tagged objects discovering details via a centralized service. The core technologies for the IoT are, in addition to RFID, WSN, and near field communications (NFC) [Path015]. This initial work culminated in the launch of the EPC Network in 2003. After the EPC Network demonstration, the Auto-ID Center was split into Auto-ID Labs (developing the hardware, software, and languages that could be integrated into the current internet in order to realize the IoT) and EPCglobal (handling IoT

commercialization). The ability for objects to communicate delivers the power of the IoT, which is found in the form of data. It is estimated that the number of M2M communication sessions would be 30 times that of human-to-human (H2H) communication on the Internet. The IoT potential is so widely recognized that industries have begun creating their own terms embodying the intent of the IoT within their particular markets. Terms like IIoT, Industry 4.0, smart planet, smart city (a superset of smart facility), smart grid, smart facility, and smart home attempt to restrict the focus of IoT technology to a specific vertical industry [Blow015]. Within these new terms, many forms of critical infrastructure are found. Federal laws in USA mandate that any virtual or physical assets whose incapacity or destruction would have a debilitating impact on security, national economics, or national public health or safety must be considered critical infrastructure.

The internet-of-things links closely to CPSS, but in this case, the focus is on sensing and exchanging the sensed data (via the Internet) and is less concerned with the human-machine interface than is the case for CPSS. A contemporary definition of the IoT is: "The IoT is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction." The development emphasis, then, for the IoT is the connectivity of "smart" devices, that is, devices with varying levels of embedded intelligence but, more particularly, devices that are uniquely identified and carry with them data about themselves and their environment. The IoT is, genuinely, the extension of the Internet from a web connecting information systems to a web connecting physical things that carry and use information [Hens016].

According to Intel CTO, Michael W. Condry, the number connected devices of all kinds continues to grow daily. This connectivity includes smart client devices from PCs to smart phones to control systems, cloud services, and even vehicles. These devices have sensors (*e.g.*, accelerometer/motion, location/GPS, cameras) in addition to their computing and connection capabilities. This coupled with powerful cloud servers opens an environment of opportunity for service-based automation with control-services from the factory to the office to the home [CBKD014].

Information and communication technology (ICT) (insecure and vulnerable to security attacks) is the key infrastructure of the IoT. Due to the limited energy and the resource constraints of the IoT components, it is difficult to shift in-hand security schemes into the future IoT. Some of the major attacks for ICT technologies are: (i) Eavesdropping, which is a passive attack where an attacker listens and uncovers useful information (used to exploit many malicious activities) during communications; (ii) spoofing, where the attacker changes the source address hiding its own (*e.g.*, man-in-the-middle); (iii) DoS, which is a form of jamming attack where an attacker sends a bulk of useless packets to the network impeding it to execute legitimate packets for providing service; and (iv) DDoS, where DoS is extended by having multiple attackers sending bulk packets flooding the target [Path015].

As IoT security solutions demand new energy-efficient security protocols, the use of in-hand cryptographic and security techniques in the IoT is still an open issue, and needs further research to ensure that within the limited processor, memory, and power constraints, the traditional cryptographic and other security algorithms can be implemented in the IoT. The most perilous attacks aiming to disrupt and exploit vulnerabilities in IoT protocols are: (i) DoS attacks

on OSI layer 1 and 3, where data is sent in bulk over the radio carrier disrupting wireless communications. IoT routing protocols like routing information protocol (RIP), border gateway protocol (BGP), open shortest path first (OSPF), are prone to security attacks such as impersonating attacks, spoofing attacks, falsification of routing packets, or selective forwarding; (ii) generic attacks, due to an IoT system incapacity to use security architecture, intelligent firewall systems, intrusion detection systems for securing the upper layer protocols; and (iii) redirection attacks, through the ICMP redirect, address resolution protocol (ARP) poisoning, and domain name system (DNS) poisoning disrupting the communication protocols stack with the attacker first controlling the communication packets and then changing the data or injecting false data. Due to the open nature of the core IoT technologies (RFID, NFC, and WSN) a big attack surface for an attacker is presented making IoT core technologies insecure and vulnerable to attacks. IoT devices are particularly vulnerable to physical attacks, software attacks, and side-channel attacks. A definition of privacy considers four domains: Physical, mental, decision, and information. Actually, privacy in the IoT is a complex sociotechnical and legal issue. The following are the most common requirements for the IoT: (i) User authentication, (ii) tamper resistance, (iii) secure execution, (iv) secure data communication, (v) identity management, (vi) confidentiality, (vii) auditing, and (viii) integrity [Path015].

B.5 Fog Computing

Fog computing, complementary to cloud computing, is a new paradigm proposed by Cisco Systems in 2012 [BMZA012] extending the computing infrastructure from the centre to the edge of the network. The widely adopted cloud computing orchestration framework can be customized to fog computing systems. Facing typical limitations in computational resources and power supply (*e.g.*, batteries and renewable energy) at the end devices in IoT, migration and offloading part of the application processing from the IoT resource-poor end devices to powerful backend clouds, which cannot completely accommodate current IoT applications because of: (i) The *transfer impracticality* of all the data from IoT end devices, where data are generated, to the backend cloud, where data are processed, due to bandwidth limitations (*e.g.*, heterogeneous sensory nodes in a driverless car are estimated to generate about 1 GB per second [LJYZ017]) and excessive transmission costs; (ii) some users not wanting their data traversing long distances from end devices to the cloud due to *privacy risks*; (iii) the noticeable *round-trip delay* from end devices to the cloud can easily degrade the performance of delay-sensitive applications, such as augmented reality (AR), online gaming, social media, and video applications that require online analytics; and (iv) *localized applications adapting dynamically* to local network states and user contexts (*e.g.*, dynamic adaptive video streaming to mobile users, it is difficult for remote clouds to rapidly respond to local contextual changes). In fact, offloaded application segments do not necessarily have to be executed in the cloud. Instead, edge facilities, such as *commercial off-the-shelf* (COTS) routers and Wi-Fi gateways, edge data centres, and home servers, can be leveraged to facilitate data processing. To this end, fog computing, in which some offloaded application segments are pushed to edge facilities close to IoT end devices, has been developed. To handle the highly heterogeneous nature of fog nodes that participate in a fog computing ecosystem, a universal orchestration platform on top of the fog nodes is the technical enabler, which brings *interoperability* (allowing heterogeneous fog nodes to operate under the same architecture), *software-programmability* (eases the way for application developers to program based on

virtualized hardware, where low-level hardware details of fog nodes are shielded), and *virtualization* (divides the resources on fog nodes into resource units, such as kernel-based VMs and containers, shared by multiple IoT applications without mutual interference). As a natural extension of cloud computing to the network edge, fog computing inherits features of cloud computing like resource orchestration, elastic provisioning, and multi-tenancy. Interconnected fog nodes form a shared pool of reconfigurable computational resources. Resource-rich fog nodes can lease their computational resources out to execute IoT applications. Fog nodes can be owned by separate parties at different geographic locations, forming a massive-scale computing network. Fog computing should be autonomous to tackle network dynamics (*e.g.*, on-off switching of IoT applications, mobility of fog nodes, unreliable access links of some fog nodes to the network). IoT applications can specify QoS requirements (non-trivial): Delay, throughput (*e.g.*, streaming rates for video applications), and data locality, to be satisfied in the affinity-aware offloading process. Research on fog computing can be viewed from three perspectives: (i) Underlying networking infrastructure, which is layered into *fog-as-a-service* (FaaS) (*e.g.*, *software-as-a-service* (SaaS), *platform-as-a-service* (PaaS), and *infrastructure-as-a-service* (IaaS)); (ii) control, which allocates the resources in the lower physical resource layer to the cloud applications in the upper service layer; and (iii) physical resource; resource orchestration, and applications. When designing the networking architecture, most studies used the prevalent *software-defined networking*, SDN, approach [JiHT017].

Fog computing offers additional appealing features: Low latency, low cost, high-multitenancy, high-scalability, and consolidation of the IoT ecosystem [ZAJW017]. Fog might be specified in terms of functionality as *fog edge nodes* (FENs), *fog server* (FS) and *foglet*, where FENs and FS are hardware nodes, and foglet is the middleware in charge of data exchange. When employing fog as a platform for IoT, a FEN accommodates adjacent smart objects for network access and edge computing, thus sensing, control and interoperation could be immediately accomplished on the FEN. An FS focuses on the interplay between FENs and the cloud. Hence, a FS controls, manages and coordinates FENs at their one-hop proximity, while a foglet offers cross-platform capability for monitoring, liaising and organizing fog resources. Overall, fog distributes computing, control, storage, and networking services along the cloud-to-things continuum, and facilitates collaborations among IoT devices and applications. Emerging development includes fog control network, fog access network, and fog storage network. The combinations of fog with IoT, *content delivery networks* (CDN), connected vehicle, and radio access network have also been investigated. In industry, Intel promotes its *fog reference design* (FRD), while Cisco advocates its IOx. Both utilize FPGA technologies to cast proprietary hardware in a chassis as a fog node, allowing users to configure and program it after manufacturing [LJYZ017].

Fog computing essentially provides solutions for: (i) On-demand computing power for devices at the edge networks with limited resources, (ii) decreasing the network traffic and delay, and (iii) increasing network resilience. Simulations to study the effect of fog computing on network traffic and delay have shown that fog computing prepares a network for better response time in case of interactive requests and makes the edge networks more resilient to challenges in the core network. The IoT has introduced new types of network protocols suitable for different data rates, range, and energy consumption has boosted this growth substantially. This poses a need for the efficient processing of unexpected traffic load, which threaten the performability

and usability of IoT applications. Fog computing is a potential solution to tackle these problems and it has been shown that adding a fog layer to the architecture of the IoT increases network resilience. In fog computing, all *event-based* and *real-time queries* are executed in the fog and with processed and refined data are transferred to the cloud where needed for more processing and decision-making applications. The OpenFog Consortium, a group founded in November 2015 of more than 60 companies and universities (*e.g.*, ARM, Cisco, Dell, Intel, Microsoft, and Princeton University), expands the fog definition after claiming that cloud connectivity is not adequate for IoT. OpenFog considers fog computing as a horizontal architecture providing a continuum of distributed computing, storage, and network services from the cloud to the edge network. The application requirements and the network status split applications either to the cloud or to the fog. Fog may experience both *challenges*, which are events disrupting the normal operation of the network and *threats* which is a challenge exploiting vulnerabilities in the network to disrupt it. Resilience in terms of fog is defined as *the ability of a system to provide and maintain an acceptable level of service in the face of various faults and challenges to the normal operation*. *Challenge tolerance* includes: (i) *Survivability* [divided into *many and targeted failures* (*e.g.*, natural disasters) and *few and random failures* (*e.g.*, fault tolerance]; (ii) *traffic tolerance* [unexpected legitimate traffic and abnormal traffic (*e.g.*, DDoS)]; and (iii) *disruption tolerance* (including delay, mobility, and connectivity, and device specific challenges like energy). *Trustworthiness* includes measurable characteristics: (i) Dependability; (ii) security; and (iii) performability [MoSt017]. Fog computing through CPSS provides a flexible orchestration and management platform that can meet the needs of emerging model Industry 4.0 [VCJL017]. Fog computing based content aware in *information centric networks* (ICN) reduce the total content number in the caching by classing data into user-shareable and non-shareable before transferring to a global network [WWLL017].

The OpenFog consortium has defined the fog pillars as: Security, scalability, open, autonomy, RAS (reliability, availability, serviceability), agility, hierarchy, and programmability [Open017].

B.6 Cyber Operations

There are many developments appearing in cyber technologies and **cyber operations** influencing innovations in social media, cybersecurity, CPSS, ethics, law, media, economics, infrastructure, military operations, and other elements of societal interaction. An increased disruption is taking place in social media, autonomy, stateless finance, quantum information systems, the IoT, the dark web, space satellite operations, and global network connectivity along with the transformation of the legal and ethical considerations of these technologies. Technical innovations vastly increase interconnectivity of physical and social systems and cause a growing need for resiliency in the vast and dynamic cyber infrastructure. Nevertheless, there is a growing need for cyber defenders to position themselves to dynamically respond to attacks through improved situational awareness, effective cyberspace command and control, and active defenses. There is a huge challenge posed by securing IoT's billions of embedded devices and trillions of sensors in the world around us. These sensors perceive the conditions around them and provide information to support an almost endless array of decision-supporting and decision-making capabilities. IoT is revolutionizing our supply chains, manufacturing, infrastructure, transportation, clothing, homes, agriculture, and even our bodies. We are instrumenting the

world around us to a degree increasingly relying on machines to augment and make decisions for us. The biggest vulnerability in the vast universe of interconnected SoS and devices in CPSS is the network itself. This has deep implications in the electronic war battlefield of the next several decades. **Space operations** [space control (offensive space control, defensive space control, and situational awareness); space support (spacelift operations, satellite operations, rendezvous and proximity operations, and reconstitution of space forces); space force enhancement (missile warning, intelligence, surveillance, and reconnaissance, environmental monitoring, satellite communications, space-based positioning, navigation, and timing); and space force application (intercontinental ballistic missiles and missile defense)] have long been critical for national security, providing critical services as *intelligence*, *surveillance*, and *reconnaissance* (ISR) capabilities since the late 1950s. American and international space assets have evolved exponentially since then, by now offering nations with navigation, timing, communications, weather, targeting, strategic warning, and defense abilities, among the more classical ISR roles. Both military and civilian affairs have been so enhanced by access to space that the realm has become indispensable to society. However, the critical role of cyber relative to space operations is still somewhat out-of-focus or hazy during normal space operations. The maxim "*you can't have space without cyber*" well known to space operators does not reflect the current understanding of both how true it is and the space systems' vulnerabilities to cyber-attacks. The **DarkNet** seems accessible only to IT professionals, hackers, and computer savvy criminals. However, more people are turning to this underground network as anonymity becomes an increasing concern. This is where the never-seen-before computer malware lives and thrives. Anything and everything known to man, both legal and illegal, is found in the DarkNet. New forms of **stateless currencies** could surpass institutions and eliminate transparency, providing a perfect agency. The exploding use of **social media** is staggering, and the ability for populations and governments to stay connected is unprecedented. Human psychology and habit patterns observed today provide understanding of how social media could be used in the future. Proper **ethical behaviour** in the cyber domain and making sure all societies and governments agree on cyber ethics is an increasingly difficult challenge. **Cybersociety** is defined as a combination of legislative actions, state and non-state actors, the military, and public-private partnerships. The ethical challenges of state-sponsored **hacktivism** and the advent of "soft" war in which warfare tactics rely on measures other than kinetic force or conventional armed conflict to achieve the political goals and national interests. Today, high school students have grown up fully immersed in this technological world and have a unique perspective on how individuals and society have to continue to adapt to it. This younger generation would add more disruption into the technological ecosystem [Blow015].

Cyberspace attacks change friendly systems through manipulating data, causing hardware failures, or physical destruction of objects controlled from cyberspace. Cyberspace espionage done well leaves defenders with no idea anyone was ever in their systems, as everything would still function. Resilience is not as useful in examining cyberspace espionage as a cyberspace attack. The Department of Homeland Security Risk Steering Committee has defined resiliency as, "The ability to adapt to changing conditions and prepare for, withstand, and rapidly recover from disruption." A perfect perimeter defense is not possible, and even if it were, attackers are often within the walls as insider threats. The United States Joint Staff has defined **cyberspace** as, "a global domain within the information environment consisting of the interdependent network

of information technology infrastructures and resident data, including the Internet, telecommunications networks, computer systems, and embedded processors and controllers." A very important point that comes from the definition is that while the Internet is part of cyberspace, it is not all of cyberspace. Most modern military equipment more complex than an M4 carbine has some form of processor from a humble truck to an aircraft carrier, and is thus part of cyberspace. There are three elements that have application to resiliency in the cyberspace domain; flexibility, a reduced attack surface, and the ability to respond dynamically to attacks [Blow015].

APPENDIX C

COMPUTATIONAL INTELLIGENCE APPROACHES

Data-driven discovery is transforming modelling, prediction, and control. Diverse fields as machine learning, engineering mathematics, and mathematical physics are amalgamated to integrate modelling and control of highly complex systems, as is the case of dynamical systems, with modern methods in data science [BrKu019].

Advances in scientific computing enable data-driven methods to be applied to a diverse range of complex systems [BrKu019] such as turbulence, the brain, climate, epidemiology, finance, robotics, autonomy, and nonetheless Internet traffic analysis, focal research area in this thesis.

Artificial neural networks are part of the area known as Intelligent Systems (connectionist systems), or Computational Intelligence. Besides artificial neural networks, the intelligent system area includes diverse tools, such as fuzzy systems, evolutionary computing, swarm intelligence, artificial immunologic systems, and intelligent agents [NHAB017].

As Cartwright points, ANNs are among the most fundamental techniques within the field of Artificial Intelligence (AI). Their operation loosely emulates the functioning of the human brain, but the value of an ANN extends well beyond its role as a **biological model**. An ANN can both memorize and reason. It provides a way in which a computer can learn from scratch about a previously unseen problem. Remarkably, the exact form of the problem is rarely critical; it might be financial (*e.g.*, predicting the direction of the stock market); it might be sociological (*e.g.*, what factors make a face attractive?); it could be medical (*e.g.*, detecting a broken bone from an X-ray); or, as in this thesis, the problem might be related to cybersecurity (*e.g.*, DDoS detection) [Cart015].

ANNs are computational models inspired by the nervous system of living beings. They have the ability to acquire and maintain knowledge (information based) and can be defined as a set of processing units, represented by artificial neurons, interlinked by a large number of interconnections (*artificial synapses*), implemented by vectors and matrices of synaptic weights [NHAB017].

The most relevant features concerning artificial neural applications are the following: (i) *Adapting from experience*. The internal parameters of the network, usually its synaptic weights, are adjusted with the examination of successive examples (patterns, samples, or measurements) related to the process behaviour, thus enabling the acquisition of knowledge by experience; (ii) *Learning capability*. Through the usage of a learning method, the network can extract the existing relationship between the several variables of the application; (iii) *Generalization capability*. Once the learning process is completed, the network can generalize the acquired knowledge, enabling the estimation of solutions so far unknown; (iv) *Data organization*. Based

on innate information of a particular process, the network can organize this information, therefore enabling the clustering of patterns with common characteristics; (v) *Fault tolerance*. Thanks to the high number of interconnections between artificial neurons, the neural network becomes a fault-tolerant system if part of its internal structure is corrupted to some degree; (vi) *Distributed storage*. The knowledge about the behaviour of a particular process learned by a neural network is stored in each one of the several synapses between the artificial neurons, therefore improving the architecture robustness in case of neurons loss; and (vii) *Facilitated prototyping*. Depending on the application particularities, most neural architectures can be easily prototyped on hardware or software, since its results, after the training process, are usually obtained with some fundamental mathematical operations [NHAB017].

In 1943, using their knowledge on neurophysiology and publishing the very first article related to artificial neurocomputing, McCulloch and Pitts demonstrated that logical operation could be performed by neurons by developing a mathematical model inspired by **biological neurons**, resulting in the first conception of the artificial neuron ([McPi943], [NHAB017]).

The information processing performed by the human brain is carried out by **biological processing components**, operating in parallel, for producing proper functions, such as thinking and learning. The fundamental cell of the central nervous system is the **neuron**, and its role comes down to conduct impulses (electrical stimuli originated from physical-chemical reactions) under certain operation conditions. This biological component can be divided into: (i) *Dendrites*. Composed of several thin extensions forming the dendritic tree. The fundamental purpose of dendrites is to acquire, continuously, stimuli from several other neurons (connectors) or from the external environment, which is the case of some neurons in contact with the environment (also called sensory neurons); (ii) *Cell body* or *soma*. Responsible for processing all the information that comes from the dendrites, to produce an activation potential that indicates if the neuron can trigger an electric impulse along its axon. It is also in the cell body where the main cytoplasmic organelles (nucleus, mitochondria, centriole, lysosome, and so forth) of the neuron can be found; and (iii) *Axon*. A single extension whose mission is to guide the electrical impulses to other connecting neurons, or to neurons directly connected to the muscular tissue (efferent neurons). The axon termination is also composed of branches called synaptic terminals [NHAB017].

Synapses are the connections that enable the transfer of electric axon impulses from a particular neuron to other neurons dendrites. There is no physical contact between the neurons forming the synaptic junction, hence the neurotransmitter elements released on the junction are in charge of weighting the transmission from one neuron to another. In fact, the functionality of a neuron is dependable of its synaptic weighting, which is also dynamic and dependent on the cerebral chemistry, as pointed by Hodkin and Huxley in 1952. The neural membrane action potential has negative values when resting (polarized), which means that there is a larger concentration of negative ions inside the membrane than at its exterior. When the nervous cell is stimulated (depolarized) with an impulse higher than its activation threshold (−55 mV), caused by the variation of internal concentrations of sodium (Na^+) and potassium (K^+) ions, it triggers an electrical impulse which propagates throughout its axon with a maximum amplitude of 35 mV, as indicated by Kandel in 2012. The amplitude of 35 mV, maximum action voltage, is fixed and strictly satisfied for all neurons when they are stimulated, however, the signal duration in time is variable. This fact can be observed independently of the category of the neuron (connector, afferent, or efferent). When the excitation ends, the membrane repolarizes, meaning the action

voltage returns to its resting value (–70 mV) [NHAB017].

ANNs structures are based on known models of biological nervous systems and the human brain itself. The computational components or processing units, called **artificial neurons**, are simplified models of biological neurons. These models were inspired by the analysis of how a cell membrane of a neuron generates and propagates electrical impulses, as found by Hodgkin and Huxley in 1952. The artificial neurons used in ANNs are nonlinear, usually providing continuous outputs, and performing simple functions, such as gathering signals available on their inputs, assembling them according to their operational functions, and producing a response considering their innate activation functions. The simplest neuron model including the main features of a biological neural network, parallelism and high connectivity, is precisely the one proposed by McCulloch and Pitts in 1943, which still is the most used model in different ANNs architectures. The multiple input signals coming from the external environment (application) are represented by the set $\{x_1, x_2, ..., x_n\}$, analogous to the external electrical impulses gathered by the dendrites in the biological neuron. The weighing carried out by the synaptic junctions of the network are implemented on the artificial neuron as a set of synaptic weights $\{w_1, w_2, ..., w_n\}$. Analogously, the relevance of each of the $\{x_i\}$ neuron inputs is calculated by multiplying them by their corresponding synaptic weight $\{w_i\}$, thus weighting all the external information arriving to the neuron. Therefore, it is possible to verify that the output of the artificial cellular body, denoted by u, is the weighted sum of its inputs. Hence, it is possible to see that the artificial neuron is composed of seven basic elements, namely: (i) *Input signals*. $\{x_1, x_2, ..., x_n\}$ are the signals or samples coming from the external environment and representing the values assumed by the variables of a particular application. The input signals are usually normalized in order to enhance the computational efficiency of learning algorithms; (ii) *Synaptic weights*. $\{w_1, w_2, ..., w_n\}$ are the values used to weight the input variables, which enables the quantification of their relevance with respect to the functionality of the neuron; (iii) *Linear aggregator*. (Σ) gathers all input signals weighted by the synaptic weights to produce an activation voltage; (vi) *Activation threshold* or *bias*. (θ) is a variable used to specify the proper threshold that the result produced by the linear aggregator should have to generate a trigger value toward the neuron output; (v) *Activation potential*. (u) is the result produced by the difference between the linear aggregator and the activation threshold. If this value is positive, (*i.e.*, if $u \geq \theta$) then the neuron produces an excitatory potential; otherwise, its potential is inhibitory; (vi) *Activation function*. (g) whose goal is limiting the neuron output within a reasonable range of values, assumed by its own functional image. The activation functions can be categorized into two fundamental groups, partially differentiable functions (*e.g.*, step function, bipolar step function, and symmetric ramp function, for which the first order derivatives are non-existent for all points of their definition domain), and fully differentiable functions (*e.g.*, logistic function, hyperbolic tangent, Gaussian function and linear function, for which the first order derivatives exists for all points of their definition domain), when considering their complete definition domains; and (vii) *Output signal*. (y) consisting on the final value produced by the neuron given a particular set of input signals, and it can also be used as input for other sequentially interconnected neurons [NHAB017].

The two following expressions synthesize the result produced by the artificial neuron proposed by McCulloch and Pitts [NHAB017]:

$$u = \sum_{i=1}^{n} w_i \circ x_i - \theta$$
$$y = g(u) \tag{C.1}$$

The artificial neuron operation can be summarized by the following steps: (i) Presenting a set of values to the neuron, representing the input variables, (ii) Multiplying each input of the neuron to its corresponding synaptic weight, (iii) Obtaining the activation potential produced by the weighted sum of the input signals and subtracting the activation threshold, (iv) Applying a proper activation function to limit the neuron output, and (v) Compiling the output by employing the neural activation function in the activation potential [NHAB017].

Digital computers and modern theories of learning and neural processing occurred at about the same time, the 1940s. Since then, digital computers are tools for modelling individual neurons as well as clusters of neurons, which are called neural networks. Engineering is constantly looking for solutions to problems. Neuroscience provides concepts and ideas for developing and applying ANN in such quest. These models of ANN follow the general understanding of the brain and the behaviour of neurotransmitters in synaptic neural links [FrSk991]. In 1949, the first method for training ANNs was proposed; it was named Hebb's rule and was based on hypothesis and observations of neurophysiologic nature [NHAB017].

Between 1957 and 1958, Frank Rosenblatt developed the first neurocomputer called Mark I Perceptron, crafting the basic model of the Perceptron. In 1960, Widrow and Hoff developed a network called ADALINE, which is short for ADAptive LINEar Element. Later on, the MADALINE, the MultipleADALINE, was proposed. It consisted on a network whose learning is based on the Delta rule, also known as Least Mean Square (LMS) learning method [NHAB017].

Idan Segev remarks that in 1959 [ErRé959], Wilfrid Rall realized that dendrites are electrically distributed devices (rather than an isopotential "point"). This ignited the need to understand how the synaptic current spreads from the synaptic input site to other dendritic regions, in particular to the soma-axon where the output is generated. Rall's passive cable theory for dendrites has provided the theoretical foundation for this **biophysical** understanding. The experimental finding that the dendritic membrane is endowed with a rich repertoire of nonlinear voltage- and ligand-gated ion channels, and that synaptic inputs (inhibitory and excitatory) from different input sources target specific dendritic subdomains, suggested that dendrites (and their synapses) may empower neurons with enhanced computational capabilities. With this experimental and theoretical foundation, a new perspective emerged regarding the possibility that dendrites alone might implement computational functions. In 1964 Rall showed in a computational study that the soma voltage is sensitive to the temporal sequence of synaptic activation "swiping" over the dendritic tree, and that this property could be used to perform a fundamental computation, sensitivity to the direction of motion. It marked the beginning of the 50-year search for "dendritic computation". Remarkable theoretical ideas have been published regarding the role of dendrites in performing specific computations: (i) Detection of motion direction; (ii) their role in collision avoidance; (iii) storage and classification of multiple input features; (iv) calculation of position variables; (v) recovering input signals in the presence of

strong noise; or (vi) enhancing temporal resolution for coincidence detection. The first direct demonstration of dendritic computation was first provided by the in vivo study of Single and Borst in 1998. The anatomical appreciation of dendritic structure started 130 years ago, while the theoretical ideas about dendritic functions started 50 years ago [CuRT014].

The time window from 1969 to early 1980s is known for the setback/languish of ANN research. Some of the few works thereafter were the derivation of prediction algorithms using reverse gradients by Werbos in 1974, the development of ART by Grossberg in 1980, the formulation of the self-organized maps (SOMs) by Kohonen in 1982, and the recurrent network based on energy functions proposed by Hopfield in 1982. The latter is the work that brought to the ANNs area its original prestige from before 1969 [NHAB017].

Only after the end of the 1980s, scientists restored their interest in ANNs. The definitive comeback of ANNs is due to different reasons, such as the development of computers with enhanced processing and memory capabilities, the conception of more robust and efficient optimization algorithms, and finally, the novel findings about the biological nervous system. One of the fundamental works of that time was the publication of Rumelhart, Hinton and Williams' book "Parallel Distributed Processing" in 1986, which brought to the spotlight one algorithm that allowed the adjustment of weight matrices of networks with more than a single layer. Consequently, solving the old problem of learning patterns from the XOR logical function. The proposal of this algorithm, called "backpropagation," definitely revived and motivated research in ANN [NHAB017].

As the previous two paragraphs state, the modern renaissance of neural-network technology is due to the successful efforts of a handful of persistent researchers studying: Adaline and Madaline, backpropagation, bidirectional associative memory and Hopfield memory, simulated annealing, counterpropagation, SOMs, ART, spatiotemporal pattern classification, neocognition, Hebbian learning, DL, among other approaches. It shall be highlighted that it is often easy for scientists and engineers, in their pursuit of solutions to specific problems, to ignore completely the neurophysiological foundations of the ANN technology [FrSk991]. From all these approaches, this research focuses pragmatically in ART.

Pattern recognition, the act of taking in raw data and taking an action based on the "category" of the pattern, has been crucial for our survival. Pattern recognition is a complex environment that involves *modelling* of the subject, *pre-processing* of signals for keeping relevant information, *segmentation* of data, *feature extraction* seeking to measure "properties", and *classification*. Decision boundaries representing optimal trade-offs between performance on the training set and simplicity of classifier is one of the ultimate challenges in pattern classification for novel data. Classification aims to recover fundamentally the model that generated the patterns. Different classification techniques are useful depending on the type of candidate models themselves. *Statistical* pattern recognition, for instance, focuses on statistical properties of the patterns (generally expressed in probability densities). If instead the model consists of some set of crisp logical rules, then methods of *syntactic* pattern recognition, where rules or grammars describe decisions, are employed [DuHS001].

A central aspect in pattern recognition problems is achieving a "good" representation, which both reveals simply and naturally the structural relationships among the components and expresses the true (unknown) model of the patterns. A representation in which the patterns that lead to the same action are somehow "close" to one another, yet "far" from those that demand a

different action is sought. The extent to which a proper representation is either created or learned and how near and far apart are quantified determines the success of a pattern classifier. Representation favours small number of features because this might lead to simpler decision regions, and classifiers that are easier to train. Robust features (*e.g.*, relatively insensitive to noise or other errors) are desired in many scenarios. In practical applications a classifier may have to comply with constrains (*e.g.*, act quickly, use few electronic components, memory or processing steps) [DuHS001].

Feature extraction takes in a pattern and produces feature values. The number of features is virtually always chosen to be fewer than the total necessary to describe the complete target of interest, and this leads to a loss in information. In acts of *associative memory*, the system takes in a pattern and emits another pattern associative, which is representative of a general group of patterns. Feature extraction reduces the information memory somewhat, but rarely to the extent that pattern classification does. Because of the crucial role of a decision in pattern recognition information, it is fundamentally an information reduction process (*e.g.*, reduction from several thousands of bits representing colour of pixels to a single bit in a two class problem). All the steps involved in pattern recognition are subject to research and development. Many are domain or problem specific, and their solution depends upon the knowledge and insights of the designer. Learning comes in different forms (*e.g.*, supervised, unsupervised, and reinforcement) [DuHS001].

Anomalies are events, items or observations that do not conform to an anticipated pattern or other elements available in a dataset. Detection of anomalous activity addresses specific problems in engineering (*e.g.*, bank fraud, medical problems, locating errors in text or detecting DDoS attacks). Outliers, peculiarities, noise, deviations, surprise and exceptions are also termed as anomalies. Detection of anomalies is supported by two main fields of study: Statistical analysis and machine learning. Machine learning based detection is of special interest in this thesis because it offers a human-independent solution as compared with intrusion detection systems (IDSs) based on signatures. Signature based approaches require human intervention for creating, testing, and deploying the signatures. This may require hours or days, which is too long when dealing with rapid DDoS attacks. Machine learning based IDS can implement system capable of learning from data (examples/experience) and making fast decisions for test or unseen data [AsLa014]. Machine learning is based in advanced algorithms that extract patterns from input data. These patterns are identified efficiently by a ML algorithm compared to what a very skilled human could do when facing large amounts of information. A ML algorithm receives multiple instances and attributes in the training phase to form a model. This model is used to identify new instances when the ML algorithm is in the testing phase. There are a lot of prominent ML algorithms used in classification problems. Some of these algorithms inspire techniques from natural phenomena and could use adaptive learning [BAUM014].

Building a model based on a training data set that contains a collection of data examples or instances is a general approach carried in all detection problems in machine learning, as is the case of DDoS in SDN. This form of *supervised training* requires instances with a specific set of features (attributes) and the associated labels. The availability of labels defines the detection operating modes: (i) Supervised learning, (ii) unsupervised learning, and (iii) semi supervised Learning. The applicability of a specific technique for anomaly detection is determined based on the type of attributes [AsLa014]. Nevertheless, the problem for an assessment of DDoS attack

detection technique is the lack of suitable public DDoS attack datasets. The cause of this is the deep investigation of network traffic relates to covert information (*e.g.*, military related). Datasets available in cybersecurity are not a complete representative of all traffic phenomena in real networks [BoAy013]. Hence, this is a strong area that could benefit from modeling for setting benchmarks to evaluate different intrusion detection methods. Particularly, this is the approach taken in this research.

A *machine learning system* is a computer program that makes decisions based on accumulated experience from solving cases successfully. The goals of machine learning systems are twofold: Dealing with complex real-world decision-making problems and solving these problems for reaching correct conclusions [WeKu991].

From the systems design perspective, there are several reasons why there has been an increased interest in machine learning systems. New formal methods and new techniques of implementation have been developed. Both the cost and speed of running learning systems have improved dramatically over what was feasible in the late 1950s, when the first computer learning systems were developed [WeKu991].

Computer based decision-making approaches were among some of the earliest research programs in medical diagnosis, signal processing, image processing, and other pattern recognition applications. Statistical and heuristic approaches (including an early versions of neural nets) saw widespread implementation in the 1960s, and were augmented by expert knowledge-based approaches (codifying expert "rules of thumb") in the 1970s and 1980s [WeKu991]. At the time of the writing of this thesis, there are complex machine learning approaches: Deep learning, chaos-based neural networks, or quantum computing based data mining [Witt014].

The most prominent and basic task in machine learning is *prediction* via either *classification* (assigning an output as a label or a categorical identification) or *regression* (finding an output value in a continuous variable). Classification is the most widely used name, is often associated with statistical pattern recognition and has been also used to characterize applications in expert systems. In statistics the classification problem is sometimes called the *prediction problem*, and in the field of machine learning it is often called *concept learning*. The fundamental goal of empirical learning is to extract a decision rule from sample data that could be applicable to new data. A typical learning system is designed to work with a general model. "Learning" consists of choosing or adapting parameters within the model structure that work best with the available samples and others like them [WeKu991].

For classification problems, a learning system can be viewed as a higher-level system that helps build the decision-making system itself, called the *classifier*. A classifier can be represented as a black box that produces a decision for every admissible pattern of data that is presented to it. The classifier has available a finite set of samples of solved (labeled) cases. The data for each case consists of a pattern of observations and the corresponding correct classification. The general structure or classifier type must be selected by the person who has specified the problem. The objective of the learning system is to customize the classifier structure to the specific problem by finding a general way of relating any particular pattern of observations to one of the specified classes. The set of samples, *training set*, therefore contains the data that the learning system uses to find the generalized decision rules for the classifier. The set of potential observations relevant to a particular problem are also referred as *features*.

Features also go by a host of other names, including *attributes*, *variables*, tests, and measurements [WeKu991].

As only correct solved cases are used in building a specific classifier, the pattern of feature values for each case is associated with the correct classification or decision. Thus, learning in any of these systems can be viewed as a process of *generalizing* these observed empirical associations subject to the constraints imposed by the chosen classifier model. This generalization process is finding some solution that identifies essential patterns in the samples that are not overly specific to the sample data due to their limited availability or usually small size. If unlimited learning data were available, each pattern could be stored, and for a given pattern of observations one would simply look up the corresponding class that had previously been associated with it. Unfortunately, the number of possible combinations of values for even small sets of features is often huge, particularly with observations that are continuous numerical values. Even the most systematic and long-term record keeping is unlikely to cover all the possible combination of values that can arise in nature [WeKu991].

The simple requirement in classification methods is that the data is presented in the form of samples composed of patterns of observations with the *correct* classification. Although extensive computer processing dominates any learning system, people still have an important role to play in the design, selection, and implementation of any classifier. For a given problem, the relevant set of observations and conclusions must be described and defined [WeKu991].

One goal of a learning system is to extract decision rules from sample data. Samples are organized as cases, with each case consisting of measurements or feature values, and a simple indicator of the correct class. There are many different learning methods that can be applied to the same sample data. For a given application, some learning systems may do better than others. Any learning system is at the mercy of the sample data and the quality of the features. Even when no errors are made the data acquisition, the predictive capabilities of some features can be quite weak. Features that are no more predictive than chance can be considered *noise*. For any given application, features fall into those that are completely noisy, to somewhat noisy, to completely predictive. Features that appear noisy on their own may prove to be highly predictive when combined with other features. With a relatively large number of noisy features, the data can be called noisy data. Any learning system tries to extract the maximum amount of information from the sample data. However, the predictive capability of the features is fundamental to the success of any learning system [WeKu991].

An additional goal of a learning system is prediction on new cases, not *discrimination* between the existing sample cases. It is usually quite easy to find rules to discriminate, or separate, the sample cases from each other. Even with completely noisy data and hundreds of sample cases, classes can usually be distinguished with little difficulty. However, these distinctions would usually not hold up on new cases [WeKu991].

Some of the most known machine learning techniques used in classification are ANNs, support vector machines, genetic algorithms, fuzzy logic, Bayesian networks, and decision trees [AsLa014].

An artificial neural network contains optimization parameters and a set of interconnected and weighted processing elements, which transform a set of inputs to a set of desired outputs. The multilayer perceptron (MLP) is a well-known ANN that allows building a non-linear classification decision boundary (discriminate function) in the feature space to perform as non-

linear. Each layer between the output and the input considers a number of neurons. An MLP is trained (weights are modified) usually through back propagation, which follows a gradient descent method that calculates an error function (usually MSE). The error function considers the difference between the calculated output by the ANN and the desired output. Successful learning is achieved when the output of the ANN is brought close to the desired output by reducing the value of the error function through the continuous back-propagation of the error to the previous layer of neurons, which in turns adjust the weights of the neurons connections [AsLa014].

SVMs use a set of training samples, marked as either normal or abnormal, for classification. A SVM model extracts the samples attributes in the training to perform classification. Classification methods based on SVMs provide good ability of learning when applied to small sample datasets. SVMs have been extensively applied in cybersecurity [AsLa014].

Decision trees short the instances down the tree (from the root node to some leaf node). A node in the tree denotes a test of some attribute of the instance. A branch descending from a node corresponds to a possible value for an attribute. DTs perform well with large data sets, which is advantageous for real-time detection in SDN given the large amounts of data. Also, DT are robust to noisy data and construct easily interpretable models [AsLa014].

Genetic algorithms employs a search method, based on hill climbing from an arbitrary number of genes, to find an approximate solution for an optimization task. A GA creates new chromosomes from one or two parents by a crossover mechanism [BAUM014]. GAs use a mutation mechanism to prevent selecting any local results by search algorithm, which adds new chromosomes with high fitness function to the initial population. Chromosomes fitted to the answer survive [BAUM014]. GAs have been used in IDS for detecting intrusions that are either novel or those based on past behaviour. The latter requires a baseline for normal behaviour, which is then used by the GA for learning and taking decisions when unseen patterns appear. A chromosome contains genes related to attributes (*e.g.*, service, flags, login status, or superuser attempts). GAs accurately detect attacks that are based in common attributes [AsLa014].

Fuzzy logic is based on fuzzy set theory, which provides reasoning termed as an approximation rather than a crisp value. Fuzzy based techniques have been used in anomaly detection because the considered features for solving a problem are manipulated as fuzzy variables. In fuzzy logic, an object simultaneously fits into different classes. This flexibility is useful when it is difficult distinguishing classes, which is the case for IDS when differentiating between the normal and anomalous classes. Fuzzy logic is effective (*e.g.*, against probes and port scans), but its main disadvantages are high resource consumption and large time consumed in training [AsLa014].

Problems posed by cybersecurity demand computational intelligence methodologies that have capabilities of self- adaptation, configuration, diagnosis, optimization, organization, parametrization, prediction, and even generation of self-written code [NiBe016]. Hence, this research considers advanced computational intelligence algorithms that could fit as many of these demands. ANN are biologically inspired computational models attempting to simulate decision processes in networks of nerve cells in the *central nervous system* [Grau013], which consist of processing elements (*neurons*), and connections between them with coefficients (*weights*) bound to the connections. These connections constitute the neuronal structure and attached to this structure are training and recall algorithms. Neural networks are known as

connectionist models because of the connections found between the neurons. Neural network modeling is an attempt to fit a line, place or hyperplane through a set of points that can through feature extraction be completely or almost independent. Some characteristics found in real and ANN are learning and adaptation, generalization, massive parallelism, robustness, associative storage of information, and spatiotemporal information processing [ShSa016].

For centuries, people have debated whether the idea of optimization can help us understand the human mind. Aristotle proposed that all human efforts and thought are ultimately based on the pursuit of (maximizing) happiness - a kind of inborn "telos", meaning an ultimate object or aim. ***Functionality*** of the brain is about making choices which yield better results. ***Intelligence*** is about ***learning*** how to make better choices. Simple animals may be born with fixed rules about what actions to take, as a function of the state of their environment as they see it. More advanced animals, instead, have an ability to select actions based on the ***results*** that the actions might have. The brain optimization theory implies that the brain combines ***incremental learning*** with ***learning to be more creative*** - to improve the "stochastic search" of available options. According to Paul Werbos ([Werb007] and [Werb009]), some researchers in evolutionary computing or stochastic search claim that their algorithms are guaranteed to find the global optimum, eventually. However, such guarantees are not realistic because, for a system of realistic complexity, they require astronomical time to truthfully get to the optimum ([Werb007] and [Werb009]).

C.1 Deep Learning

C.1.1 History

Deep learning has had a long and rich history, known by many names, reflecting different philosophical viewpoints. It has become more useful as the amount of training data has increased. Deep learning models have grown in size over time as computer infrastructure for DL has improved. Also, deep learning has solved complicated applications with increasing accuracy over time [GoBC016].

Historically, deep learning dates back to the 1940s. Deep learning only appears to be new, because it was relatively unpopular for several years preceding its current popularity, and because it has gone through many different names, only recently being called "deep learning." The field has been rebranded many times, reflecting the influence of different researchers and different perspectives. Deep learning has been successfully used in commercial applications since the 1990s but was often regarded as being more of an art than a technology [GoBC016].

There have been three waves of development: Deep learning known as **cybernetics** (simple *linear models* motivated from a *neuroscientific* perspective) in the 1940s–1960s, deep learning known as **connectionism** ([Hebb005] and [ToMi985]) (in the context of cognitive science, which is an approach to understanding the mind where many models related to cognition spawned) or **parallel distributed processing** ([McRH995] and [RuMG986]) in the 1980s–1990s, and the current resurgence under the name **deep learning** beginning in 2006 [GoBC016].

Some of the earliest learning algorithms recognized today were intended to be *computational models* of *biological learning* as it happens or could happen in the brain. One of the names that DL has gone by is ANNs. The corresponding perspective on DL models is that

they are engineered systems inspired by biological brains. Some kinds of neural networks (generally not designed to be realistic models of biological function) used in machine learning have sometimes been used to understand brain functionality ([GoBC016] and [HiSh991]).

The *neural* perspective on DL is motivated by two main ideas: (i) The brain provides a *proof by example* that intelligent behaviour is possible, and a conceptually straightforward path to building intelligence is to *reverse engineer computational principles behind the brain and duplicate its functionality*, and (ii) it would be deeply *interesting to understand the brain and the principles that underlie human intelligence*, so machine learning models that shed light on these basic scientific questions are useful apart from their ability to solve engineering applications [GoBC016].

The modern term "deep learning" goes *beyond the neuroscientific perspective* on the current breed of machine learning models. It appeals to *a more general principle of learning multiple levels of* ***composition***, which can be applied in *machine learning frameworks* that are not necessarily neurally inspired. Hence, today *neuroscience* is regarded as *an important source of inspiration* for DL researchers, but it is *no longer the predominant guide* for the field because there is not enough information about the brain to use it as a guide. One should not view DL as an attempt to simulate the brain. Modern deep learning draws inspiration from many fields, especially applied math fundamentals like linear algebra, probability, information theory, and numerical optimization. While some deep learning researchers cite neuroscience as an important source of inspiration, others are not concerned with neuroscience at all.

It is worth noting that the effort to understand how the brain works on an algorithmic level is alive and well. This endeavour is primarily known as *computational neuroscience* and is a separate field of study from *deep learning*. Researchers frequently move back and forth between both fields. The field of *deep learning* is primarily concerned with how to *build computer systems* that are able to successfully solve tasks *requiring intelligence*, while the field of *computational neuroscience* is primarily concerned with *building more accurate models of how the brain actually works* [GoBC016].

Several key concepts arose during the **connectionism** movement of the 1980s that remain central to DL. Of significant interest is the concept of **distributed representation** [HiSe986], which conveys the idea that each input to a system should be represented by many features, and each feature should be involved in the representation of many possible inputs. A major accomplishment of the connectionist movement was the successful use of *back-propagation* to train deep neural networks ([Cun987] and [RuHW986]). During the 1990s, researchers made important advances in modelling sequences with neural networks ([BeSF994], [GoBC016] and [Hoch991]).

In 2006, Geoffrey Hinton [HiOT006] showed that a kind of neural network called a *deep belief network* could be efficiently trained using a strategy called greedy layer-wise pretraining. Alternate research groups showed that the same strategy could be used to train many other kinds of deep networks ([BLPL006] and [RPCL006]) and systematically helped to improve generalization on test examples. This wave of neural networks research popularized the use of the term "deep learning" to emphasize that researchers could train deeper neural networks, and to focus attention on the theoretical importance of depth ([BeCu007], [DeBe011], [Mont014], [MPCB014], and [PGCB014]). Within this wave, deep neural networks outperformed competing AI systems based on other machine learning technologies as well as hand-designed functionality.

This wave began with a focus on unsupervised learning techniques and the ability of generalization from small datasets (*e.g.*, [Ande935], [Fish936], [Gars900], and [Goss008]). There is also interest in much older supervised learning algorithms and the ability of deep models to leverage large labelled datasets (*e.g.*, the public Street View House Numbers (SVHN) dataset [NWCB011] and statistical machine translation datasets [Stat014]) [GoBC016].

Deep learning is now used by many top technology companies, including Apple, Facebook, Google, IBM, Microsoft, and NVIDIA in the United States of America, whilst Alibaba, Baidu, and Tencent in China. Deep learning has also made contributions to other sciences (*e.g.*, visual processing models in neuroscience, prediction of molecules interaction in pharmacy, searching for subatomic particles in quantum mechanics, parsing microscope images to construct a 3-D map of the human brain, and so on and so forth) [GoBC016].

C.1.2 Concepts

A hierarchy of concepts enables the computer to learn *complicated concepts*, by constructing them, out of *simpler* ones. A graph showing how these concepts are built on top of each other has multiple or many *layers*. Hence, the graph is *deep*. This approach to AI is also known as **deep learning** [GoBC016].

A computer can reason automatically about statements in formal languages that use logical inference rules. This is known as the **knowledge base** approach to AI. Some projects have sought to *hard-code knowledge* about the world in formal languages, but unfortunately none has led to a major success [GoBC016].

Systems that rely on hard-coded knowledge suggest that AI systems need the ability to acquire their own knowledge, by extracting patterns from raw data. This capability is known as **machine learning**. Machine learning enables computers tackling problems involving knowledge of the real world and making decisions that appear subjective [GoBC016].

The performance of *simple* machine learning algorithms depends heavily on the **representation** of the data they are fed with. Each piece of information mapped into the representation is known as a **feature**. Machine learning algorithms discover how features **correlate** with diverse outcomes. It is significant expressing that feature design is not a trivial undertaking [GoBC016].

The use machine learning to discover not only the mapping from representation to output but also the representation itself is known as **representation learning**. Learned representations often result in much better performance than the obtained with hand-designed representations and also enable AI systems to rapidly adapt to new tasks, with minimal human intervention. The quintessential example of a representation learning algorithm is the **autoencoder**. An autoencoder is the combination of an encoder function, which converts the input data into a different representation, and a decoder function, which converts the new representation back into the original format [GoBC016].

When designing features or algorithms for learning features, the goal is usually to separate the **factors of variation** that explain the observed data. The word “factors” refers the sources of influence, which are usually not combined by multiplication and are often non directly observable quantities. These factors may exist as: (i) Either unobserved objects or unobserved forces in the physical world that affect observable quantities, and (ii) constructs in the human mind that provide useful simplifying explanations or inferred causes of the observed data. These

factors are analogous to concepts or abstractions that help making sense of the rich variability in the data (*e.g.*, in a speech recording the factors of variation include age, sex, accent, and the spoken words).

A difficulty in many real-world AI applications is that many of the factors of variation influence every single observed piece of data. These AI applications require disentanglement of the factors of variation and discard the non-relevant ones. **Deep learning** solves this central problem in representation learning by introducing representations expressed in terms of simpler representations. Consequently, deep learning enables the computer building complex concepts out of simpler concepts [GoBC016].

The quintessential example of a DL model is the feedforward deep network, or **multilayer perceptron**. A multilayer perceptron is a mathematical function mapping some set of input values to output values. The function is formed by composing many simpler functions [GoBC016].

The idea of learning the right representation for the data provides one perspective on DL. Another perspective on DL is that *depth enables the computer to learn a* **multistep computer program**. Each layer of the representation is then analogous to the state of the computer's memory after executing a pipeline/set of instructions in parallel. Networks with greater depth can execute more instructions in sequence. Sequential instructions offer great power because later instructions could refer back to results of earlier instructions. According to this view of DL, not all the information in a layer's activations necessarily encodes factors of variation that explain the input. The representation also stores *state information* that helps to *execute a program making sense of the input*. This state information could be analogous to the *program counter* in a microprocessor or the **pointer** in a traditional computer program. This has nothing to do with the content of the input specifically, but it helps the model to organize its processing [GoBC016].

C.1.3 Awareness of Depth

Conventionally, there are *two* main ways of measuring the **depth of a model**. The *first view* is based on the *number of sequential instructions* that must be executed to evaluate the architecture. This is analogous to the *length of the longest path* through a flow chart that describes how to compute each of the model's outputs given its inputs. Similar to equivalent computer programs having different lengths depending on the language used to write the program. The *second view*, used by deep probabilistic models, regards the depth of a model as the *depth of the graph describing how concepts are related to each other* [GoBC016].

Since it is not always clear which of these two views—depth of the computational graph, or depth of the probabilistic modelling graph—is most relevant, and because different sets of smallest elements are chosen from which to construct graphs, there is no single correct value for the depth of an architecture, just as there is no single correct value for the length of a computer program. Also, there is no consensus about how much depth a model requires to qualify as "deep." However, *deep learning can be safely regarded as the study of models that involve a greater amount of* ***composition*** *of either learned* ***functions*** *or learned* ***concepts*** *than traditional machine learning does* [GoBC016].

C.1.4 Deep Learning Definition

Succinctly, *deep learning* (*e.g.*, MLP) is a kind of *representation learning* (*e.g.*, shallow autoencoders), which is in turn is a kind of *machine learning* (*e.g.*, logistic regression), which is used for many but not all approaches to *artificial intelligence* (*e.g.*, knowledge bases). Within *representation learning*, a *deep learning* **implementation** mechanism works by extracting *simple features* from the *input data*, then *additional layers* extracting more *abstract features*, and finally *mapping* the *highest abstract features* to *outputs* [GoBC016].

Deep learning is an approach to machine learning that has drawn heavily on the knowledge of the human brain, statistics and applied math as it developed over the past several decades [GoBC016].

C.1.5 Big Data

The size of benchmark datasets has expanded remarkably over time, from *hundreds* (10^2) *or thousands* (10^3) *of samples* (manually compiled) used by statisticians in the 1900s to *tens of millions* (10^7) *of samples* in the 2010s. This astronomical increate in the availability of data is driven by the increasing *digitization* of society. Since more and more of human activities take place on computers, more and more is *recorded*. As computers are increasingly *networked* together, it becomes easier to centralize *records* and curate them into *datasets* appropriate for machine learning applications. This phenomenon is "Big Data", which has made machine learning much easier because the key burden of statistical estimation, generalizing well to new data after observing only a small amount of data, has been considerably lightened. As of 2016, a rough rule of thumb is that a supervised deep learning algorithm achieves acceptable performance with around *five thousand* $(5x10^3)$ labelled samples per category and would match or exceed human performance when trained with a dataset containing at least *ten million* (10^7) labelled samples [GoBC016].

C.1.6 Model Sizes

One of the main insights of *connectionism* is that animals become intelligent when many of their neurons work together. Biological neurons are not especially densely connected. Machine learning models have had a number of connections per neuron within an order of magnitude of even mammalian brains for decades. Neural networks have been astonishingly small, in terms of the number if neurons, until quite recently. Since the introduction of hidden units, ANNs have doubled in size roughly every 2.4 years. This growth is driven by faster computers, larger memories, and the availability of larger datasets. Larger ANNs achieve higher accuracy on more complex tasks. Unless new technologies enable faster scaling, ANNs are expected to have the same number of neurons as the human brain until at least the 2050s. It is worth noting that if biological neurons could represent more complicated functions than current artificial neurons, which are simplified models of reality, then biological neural networks could be even larger than what *connectionism* has proposed [GoBC016].

Model size for DL increased over time, due to the availability of faster CPUs, the advent of general purpose GPUs, faster network connectivity, better software infrastructure for distributed computing. The increase in model size is expected to continue [GoBC016].

C.1.7 Accuracy, Complexity, and Real-World

Since the 1980s, DL has consistently improved in its ability to provide *accurate* recognition and prediction. Moreover, deep learning has consistently been applied with success to broader sets of applications [GoBC016].

Early deep models, in the 1980s, recognize individual objects in tightly cropped, extremely small images. Since then there has been a gradual increase in the size of images ANN could process. Modern ANN process rich high-resolution photographs and do not have a requirement for the photo be cropped near the object to be recognized. Earliest ANN could recognize only two kinds of objects (in some cases, the absence or presence of a single kind of object), while modern ANN typically recognize at least 1,000 objects [GoBC016].

At the same time that the scale and accuracy of deep networks have increased so has the complexity of the tasks that they can solve. Deep learning has succeeded in applications where neural networks, utilizing neural Turing machines, can learn simple programs from examples of desired behavior. This self-programming technology is still in its infancy, but in the future it could in principle spawn to more applications [GoBC016].

Another *crowning achievement* of DL is its extension to the domain of **reinforcement learning**. In this context, an autonomous agent learns to perform a task by trial and error, without any guidance from a human. DeepMind demonstrated that a reinforcement learning system based on DL is capable of learning to play Atari video games, reaching human-level performance on many tasks. Deep learning has significantly improved the performance of reinforcement learning for robotics [GoBC016]. Selected computational intelligence approaches are presented next.

C.2 Backpropagation Neural Networks

Backpropagation, proposed in the 70s along with adaptive dynamic programming (ADP) (explained in detail in [Werb009]), is a neural technique that has a range of useful properties that converted it in a mainstay in contemporary pattern recognition research. Backpropagation neural networks (BNN) is one of the approaches of multilayer neural networks (MNN), where the parameters governing the nonlinear mapping are learned at the same time as those governing the linear discriminant. MNN overcome the drawbacks and limitations of two-layer networks and at least in principle provide the optimal solution to an arbitrary classification problem. At their base, MNN implement linear discriminants, but in a space where the inputs have been mapped nonlinearly. The flexibility of MNN is admitting simple algorithms allowing learning the nonlinearity shape from training data. Thus, the models are extremely powerful, have nice theoretical properties, and apply well to a vast array of real-world applications. The *backpropagation* algorithm, based on gradient descent in error, is one of the most popular and simple (even for complex models with hundreds or thousands of parameters) methods for training MNN. Neural networks are the most accessible technique for performing statistical pattern recognition. A number of tricks (scaling of input values and initial weights, and desired output values) in backpropagation are often used to improve performance and increase training speed. Network architecture or topology, problem dependant, plays an important role for neural net classification. Knowledge of the problem domain, of an informal or heuristic nature, can be incorporated into network architectures through choices in the number of hidden layers, units, or

feedback connections. Whereas the number of inputs and outputs is given by the feature space and number of categories, the total number of weights or parameters in the network is not (or at least not directly). If too many free parameters are used, generalization would be poor; conversely if too few parameters are used, the training data cannot be learned adequately. It is crucial to know that neural networks do not exempt designers from intimate knowledge of the data and problem domain [DuHS001].

Nonlinear multilayer networks (with input units, hidden units and output units) have greater computational or *expressive power* (more functions can be implemented) than similar networks lacking hidden units. In fact, given sufficient number of hidden units of a general type *any* function can be represented. This claim was proven by Kolmogorov and it is sustained that any continuous function from input to output can be implemented in a three-layer net, given sufficient number of hidden units, proper nonlinearities, and weights. An intuitive analogy to the universal expressive power of three-layer nets is inspired by Fourier's Theorem that any continuous function can be approximated arbitrarily closely by a (possibly infinite) sum of harmonic functions. Imagine a network whose *hidden units* implement harmonic functions and proper *hidden-to-output weights* related to the coefficients in a Fourier synthesis would then enable the full network to implement the desired function. Informally speaking, harmonic functions for Fourier-like synthesis of a desired function are not required to be built up, but a sufficiently large number of "bumps" at different input locations, of different amplitude and sign, can be put together to give the desired function [DuHS001].

C.3 Supervised Learning

While the computer can carry out many different forms of analysis, much of the potential for successful classification and prediction lies with the person that selects the observations for analysis in the first place. Technically, this form of learning is called supervised learning because the system learns from a set of known correctly classified cases, which have been produced by the human expert that "supervises" the choice of learning cases [WeKu991].

Any method that incorporates information from training samples in the design of a classifier employs learning. Learning refers to some form of algorithm for reducing the error on a set of training data. A range of *gradient descent* algorithms that alter a classifier's parameters in order to reduce an error measure now permeate the field of statistical pattern recognition [DuHS001].

An IDS inspects network traffic to discover security threats (*e.g.*, unauthorized access) based on predefined signatures (rules), collected information from attack packets and stored log files. DDoS attacks have been previously classified on packet threshold, attack duration, packet rate, heuristic identification of source IP address of IP-spoofed internet attacks, source and destination IP addresses, protocols, TTL values, least significant bytes of source IP addresses. However, the signature-based IDS are insufficient in detecting new or modified attacks for which no signature exists. This is the motivation for training machine learning algorithms so that novel forms of attacks would be detected and classified successfully. Network traffic analysers (*e.g.*, Bro or Corsaro) provide traffic information in packets, which is used for feature selection (pre-processing) and further fed into classifiers. The most informative 12 features used in classifiers according to the literature are: IP source, source country, IP source port, IP destination port, protocol, SYN Flag, ACK Flag, RST flag, packet length, packet time-to-live, delta time, and

alert. Supervised learning uses labelled training data, where classes are known, for creating a model to classify new instances into discrete categories. DT and naive Bayes are popular supervised machine learning classifiers used in DDoS detection [BaAZ014].

C.4 Hybrid Machine Learning

Inefficiency in terms of accuracy and computational cost in machine learning algorithms, has led to the creation of hybrid machine learning methods for detecting DDoS attacks. Specifically, a GA used for feature selection and decreasing dimensionality, and an ANN for attack detection for improving the detection rate [BAUM014].

Sensors are implemented to collect the traffic (incoming, related to external accessible servers, and internal) passing through the network. Then, datasets including attack traffic are used for the evaluation of experiments [BAUM014].

Selection of the most significant feature sets to be used in classification is always a challenge in any classification problem as is the case of DDoS. Feature selection as part of pre-processing could be generated by packet headers, payload, or protocol handshaking. Feature selection decreases the dimensionality of data and improves the classification performance in term of speed. Whereas, using all features from the dataset or traffic being monitored can cause large memory and disk usage and delay significantly the detection phase. Hence, the feature selection purpose is choosing representative features with high discriminative power [BAUM014].

C.5 Unsupervised Machine Learning

The absence of a human expert in the training of a learning system is known as “unsupervised” learning or clustering, where solved cases are not known, so no classification can be given, and the samples consist only of observables. In this situation the goal is to identify clusters of patterns that are similar, thus identifying potential classes. This type of problem, also known classically as numerical taxonomy is far less structured, and its potential for success is much more limited, as it involves much more guessing [WeKu991].

C.6 Nonparametric Machine Learning

From a statistical pattern recognition perspective, most of these newer techniques fall into the class of *nonparametric* methods. That is, they make no assumptions about the mathematical functional form of the underlying population density distribution, such as that of a Gaussian curve. Each of the methods does assume a certain form of underlying model for the classifier or its learning capabilities, but within this model there are typically many possible choices [WeKu991].

APPENDIX D

ENSEMBLES OF CLASSIFIERS

Advanced statistical learning approaches in classification consider ensembles of classifiers. This scheme makes linear ensembles (collections) of model fitting methods, multi-classification, instead of using a single fit of the method in question. Hence, multi-classification is a form of majority logic in which a voting mechanism is implemented. *Bootstrap aggregation* and *AdaBoost* (short for adaptive boosting) group are known methods for making classifiers ensembles. These utilize a base learning algorithm many times with various training sets [BoAy013].

Ensemble learning, aka multiple classification systems, have many real-world applications, including object detection and tracking, scene segmentation and analysis, image recognition, information retrieval, bioinformatics, data mining, feature selection, confidence estimation, missing feature, incremental learning, error correction, class-imbalanced data, or learning concept drift from nonstationary distributions. Viola and Jones equipped modern digital cameras with face detection technology through ensemble learning. Similar machine learning technologies are used by the tracking algorithm adopted in the Xbox Kinect sensor. However, applications in cybersecurity are few. Ensemble learning reduces the variance, thereby improving the accuracy, of an automated decision-making system. However, forms of ensemble-based decision systems (*e.g.*, consulting several doctors before a major medical operation, reading user reviews before a purchase, or calling references before hiring a job applicant) are second nature for the society and some (*e.g.*, essence of democracy) have been around perhaps as long as the civilized communities existed. The original goal for using ensemble systems is analogous to the use of such mechanisms in our daily lives, improving confidence about making the right decision by weighing distinct opinions and combining them through some thought process to reach a final decision. The three pillars of the ensemble systems are: Diversity, training ensemble members, and combining ensemble members [ZhMa012].

Any classification error has two components that can be controlled: *Bias*, the accuracy of the classifier; and *variance*, the precision of the classifier when trained on different training sets. These two components have a trade-off relationship: Classifiers with low bias tend to have high variance and vice versa. It is known that averaging has a smoothing (variance-reducing) effect. The goal of ensemble systems is to create several classifiers with relatively fixed (or similar) bias and then combining their outputs (by averaging) to reduce the variance. In the context of ensemble systems, there are many ways of combining ensemble members, of which averaging the classifier outputs is only one method. Combining the classifier outputs may not necessarily lead to a classification performance that is guaranteed to be better than the best classifier in the ensemble. Rather, it reduces the likelihood of choosing a classifier with a poor performance. A

representative illustration of the variance reduction ability of the ensemble of classifiers is shown in Fig. D.1.

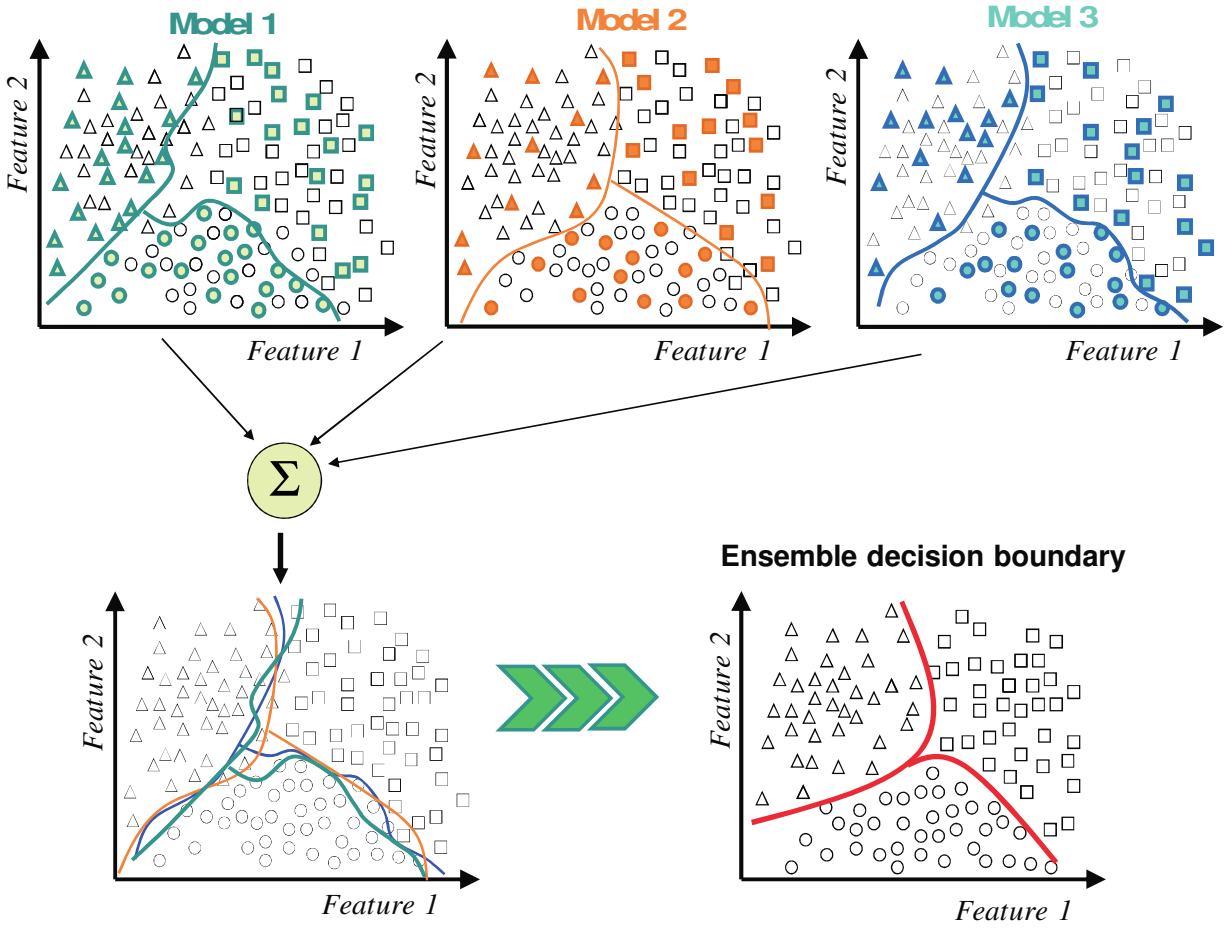


Fig. D.1. Variability reduction using ensemble systems. From [ZhMa012].

Dasarathy and Sheela's work in 1979 is one of the earliest examples of ensemble systems, which focused on partitioning the feature space using multiple classifiers. About a decade later, Hansen and Salamon showed that an ensemble of similarly configured ANNs can be used to improve classification performance. However, it was Schapire's work that demonstrated through boosting that a strong classifier with an arbitrarily low error on a binary classification problem, can be constructed from an ensemble of classifiers, the error of any of which is merely better than that of random guessing. This theory of boosting paved the path for the subsequent suite of AdaBoost (short for adaptive boosting) algorithms, most popular ensemble-based algorithms, extending the boosting concept to multiple class and regression problems. Due to these seminal works ensemble-based algorithms appeared under different names: Bagging, random forests (an ensemble of decision trees), composite classifier systems, mixture of experts (MoE), stacked generalization, consensus aggregation, combination of multiple classifiers, dynamic classifier selection, classifier fusion, committee of neural networks,

or classifier ensembles. Ensemble-based systems typically differ from each other in three pillars: (i) Selection of training data for individual classifiers, (ii) the specific procedure used for generating ensemble members, and/or (iii) the combination rule for obtaining the ensemble decision [ZhMa012].

D.1 Data Sampling Selection: Diversity

Making different errors on any given sample is of paramount importance in ensemble-based systems. If all ensemble members provide the same output, there is nothing to be gained from their combination. Therefore, diversity in the decisions of ensemble members is required, particularly when an error is made. Ideally, classifier outputs should be independent or negatively correlated. Diversity in ensembles can be achieved through several strategies, although using different subsets of the training data is a common approach, as illustrated in Fig. D.1. Different sampling strategies lead to different ensemble algorithms (*e.g.*, using bootstrapped replicas of the training data leads to bagging, sampling from a distribution that favors previously misclassified samples is the core of boosting algorithms, different subsets of the available features to train each classifier leads to *random subspace methods*, less common approaches include using different parameters of the base classifier like training an ensemble of multilayer perceptrons with a different number of hidden layer nodes or even using different base classifiers as the ensemble members) [ZhMa012].

D.2 Training Member Classifiers

The core of any ensemble-based system is the strategy used to train individual ensemble members. Numerous competing algorithms have been developed for training ensemble classifiers; nevertheless, bagging (and related algorithms arc-x4 and random forests), boosting (and its many variations), stack generalization and hierarchical MoE remain as the most commonly employed approaches [ZhMa012].

D.3 Combining Ensemble Members

The mechanism used to combine the individual classifiers is the last step in any ensemble-based system. The strategy used in this step depends partly on classifiers used as ensemble members (*e.g.*, *SVM* provide only discrete-valued label outputs where (simple or weighted) majority voting fits, *multilayer perceptron* or (naïve) Bayes classifier provide continuous valued class-specific outputs that are interpreted as the support given by the classifier to each class where a wider array of options is available, such as arithmetic (sum, product, or mean) combiners or more sophisticated decision templates) [ZhMa012].

Ensemble members are used in one of two general settings: *Classifier selection* where each classifier is trained as a local expert in some local neighborhood of the entire feature space, and *classifier fusion* where all classifiers are trained over the entire feature space, and then combined to obtain a composite classifier with lower variance (and hence lower error). Bagging, random forests, arc-x4, and boosting/AdaBoost are examples of the latter approach.

Bootstrap aggregation or bagging, the training sets $S_{B\{1,2,\ldots n\}}$ are bootstrap copies, extracted uniformly, from the original training set S_O. AdaBoost is a machine learning metaheuristic algorithm formulated by Yoav Freund and Robert Schapire, from AT&T Bell

Laboratories, [FrSc997] and was given the Gödel Prize in 2003. The Gödel Prize, given jointly by the European association for theoretical computer science (EATCS) and the Association for Computing Machinery (ACM) Special Interest Group on Algorithms and Computational Theory (SIGACT), recognizes outstanding papers in the area of theoretical computer science. AdaBoost can be used with many types of learning algorithms (possibly weak) to improve their individual performance. AdaBoost assigns a set of weights W to the original training set S_O and adaptively modifies the weights after each classifier is trained by the main learning algorithm. The adaptive modifications increase the weight of misclassified instances and reduce the weight of properly classified instances [BoAy013]. Unlike ANNs and SVMs, the AdaBoost training process selects only those features known to improve the predictive power of the model.

The features fed to a classifiers ensemble can also be selected by a machine learning algorithm (*e.g.*, SVM), when these are many. The architecture of classifiers ensemble-based detection of DDoS is integrated by these layers: Features set (classifiers inputs), features subset (when features are many), classifiers, classifiers outputs, and ensemble output [BoAy013].

The execution of the ensemble starts with a classifier acting upon the training data features provided as inputs. Then, the classification weights, 1 or 0, are assigned to misclassified instances or proper instances respectively. The next classifier (trained differently) uses the training data features for the misclassified instances (with value 1 in the weight field) and produces new outputs for them. This process of weights updating continues for all the classifiers that are part of the classifiers ensemble. This boosting technique reduces the processing spent in each classifier progressively and diminishes the overall execution time. This time reduction is a consequence of a smaller set of input instances (those misclassified) for each classifier in the ensemble [BoAy013].

The classification decision to the known status (attack or normal) of the instance is defined as: *False positive* (FP) if a normal traffic instance is classified as an attack, *false negative* (FN) if an attack instance is classified as normal traffic, *true negative* (TN) if a normal traffic instance is classified successfully as normal traffic, and *true positive* (TP) if an attack instance is classified successfully as an attack [BoAy013].

The algorithm classification precision for detecting DDoS attacks is defined as the ratio of TP to the sum of TP and FP.

$$\text{Precision=True Alarm Rate (TAR)}=TP/(TP+FP) \quad \text{(D.1)}$$

The false alarm rate is defined as 1-precision.

$$\text{False Alarm Rate (FAR)}=1-\text{Precision}=FP/(TP+FP) \quad \text{(D.2)}$$

The accuracy, representing the ensemble correctness, is expressed as:

$$\text{Accuracy Ratio (AcR)}=(TP+TN)/(TP+TN+FP+FN) \quad \text{(D.3)}$$

Classifiers ensembles have the advantage of higher accuracy and lower false alarms compared to other widely used machine learning schemes in IDSs [BoAy013].

D.4 Fuzzy Logic in Ensemble Classifiers

A fuzzy classifier functions with input membership functions that process the data features fed into the classifier. In fuzzy classifiers, the membership functions can be modified or trained, utilizing optimization methods like least-squares or BP gradient descent, analogously to the connection weights in ANN.

APPENDIX E

GEOMETRIC INTERPRETATION OF FUZZYART

LEARNING WITH COMPLEMENT CODING

Let the input patterns be two-dimensional vectors $\mathbf{a}=(a_1,a_2)$. By complement coding, the effective input vectors are four-dimensional $\mathbf{I}=(a_1,a_2,1-a_1,1-a_2)$. In this case, each category *j* is represented by a four-dimensional weight vector [SeLA012]

$$\mathbf{z}_j=(\mathbf{u}_j,\mathbf{v}_j^c) \tag{E.1}$$

where $\mathbf{u}_j$ and $\mathbf{v}_j$ are two-dimensional vectors. Consider a rectangle R_j with corners defined by vectors $\mathbf{u}_j$ and $\mathbf{v}_j$. The size of the rectangle R_j can be defined as [SeLA012]

$$|R_j|=|\mathbf{v}_j-\mathbf{u}_j| \tag{E.2}$$

Assuming that the system is in the fast learning mode, that is, $\beta=1$ in the fuzzy learning rule. When a category becomes committed for the first time by an input pattern $\mathbf{I}=(\mathbf{a},\ \mathbf{a}^c)$, that category learns the template

$$\mathbf{z}_j(new)=(\mathbf{a},\ \mathbf{a}^c) \tag{E.3}$$

so that $\mathbf{u}_j=\mathbf{v}_j=\mathbf{a}$ and the rectangle R_j is just point $\mathbf{a}$ and it has zero size [SeLA012].

When a new input pattern $\mathbf{I}=(\mathbf{b},\ \mathbf{b}^c)$ is added to a category j, rectangle R_j is expanded (according to the fuzzy learning rule given with $\beta=1$) to [SeLA012]

$$\begin{aligned}\mathbf{z}_j(new) &= (\mathbf{u}_j(new),\ \mathbf{v}_j^c(new))\\ &= (\mathbf{u}_j(old)\wedge\mathbf{b},\ \mathbf{v}_j^c(old)\wedge\mathbf{b}^c)\\ &= (\mathbf{u}_j(old)\wedge\mathbf{b},\ (1-\mathbf{v}_j(old))\wedge(1-\mathbf{b}))\\ &= (\mathbf{u}_j(old)\wedge\mathbf{b},\ (\mathbf{v}_j(old)\vee\mathbf{b})^c)\end{aligned} \tag{E.4}$$

where the symbol $\vee$ denotes the component wise fuzzy MAX operator. Therefore [SeLA012],

$$\begin{aligned}\mathbf{u}_j(new)&=\mathbf{u}_j(old)\wedge\mathbf{b}\\ \mathbf{v}_j(new)&=\mathbf{v}_j(old)\vee\mathbf{b}\end{aligned} \tag{E.5}$$

Rectangle R_j may be expanded when category *j* incorporates a new input vector. Rectangle R_j is expanded by the minimum size needed to incorporate the new input vector $\mathbf{b}$ inside the rectangle. In particular, if $\mathbf{b}$ is an input vector inside R_j no weight change occurs during the weight update [SeLA012].

The maximum size that a rectangle R_j can reach is limited by the vigilance parameter ρ. This can be reasoned as follows. If an input vector $\mathbf{I}=(\mathbf{b}, \mathbf{b}^c)$ activates a category *j*, this category resets whenever [SeLA012],

$$|\mathbf{I}\wedge\mathbf{z}_j| < \rho|\mathbf{I}| \tag{E.6}$$

Since input vectors are two dimensional and complement coding is used, $|\mathbf{I}|=N=2$. Hence, the reset condition becomes [SeLA012],

$$|\mathbf{I}\wedge\mathbf{z}_j| < 2\rho \tag{E.7}$$

but,

$$\begin{aligned} |\mathbf{I}\wedge\mathbf{z}_j| &= |(\mathbf{b}, \mathbf{b}^c)\wedge(\mathbf{u}_j, \mathbf{v}_j^c)| \\ &= |(\mathbf{b}\wedge\mathbf{u}_j)+(\mathbf{b}\vee\mathbf{v}_j)^c| \\ &= |(\mathbf{b}\wedge\mathbf{u}_j)+2-|\mathbf{b}\vee\mathbf{v}_j| \\ &= |\mathbf{u}_j(new)|+2-|\mathbf{v}_j(new)_j| \\ &= 2-|R_j(new)| \end{aligned} \tag{E.8}$$

Therefore, the category resets whenever [SeLA012],

$$|R_j(new)| > 2(1-\rho) \tag{E.9}$$

and the maximum size of the rectangles is limited by $2(1-\rho)$ in the 2-dimensional case. For input vectors with *N* components (2*N* after complement coding) the maximum size rectangle is limited by $N(1-\rho)$. Consequently, the closer ρ is to '1' the smaller the size of the rectangles R_j is and the smaller the number of input patterns coded in each category is [SeLA012].

The fact that *rectangles grow during learning* and that their *maximum size is bounded* allows the existence of a *stable category learning theorem*, which guarantees that no category proliferation occurs. If no complement coding is used, the input space '*rectangle covering*' may be substituted by a '*triangle covering*'. But it turns out that the resulting triangles have a size which depends directly on the norm of their weight templates $|\mathbf{z}_J|$. This means that as $\mathbf{z}_J$ shrinks its associate triangle shrink as well, and thus triangles close to the origin are small. Consequently, the number of triangles needed to '*cover*' the input space close to the origin grows. This together with the fact that triangles may shrink during learning produces the category proliferation problem if input patterns are not normalized [SeLA012].

APPENDIX F

DISTRIBUTED DENIAL-OF-SERVICE DATASET

Access to the distributed denial-of-service dataset used in this research has been obtained through the Protected Repository for the Defense of Infrastructure Against Cyber Threats, PREDICT. An agreement with the United States Department of Homeland Security (DHS) PREDICT Project listing the terms and conditions of use for the DDoS dataset has been signed. Abiding to these regulations, this DDoS dataset cannot be shared, sent, transmitted, or otherwise cause the Data to be transported to any country or location outside of the United States of America that is not on the approved PREDICT country list. Among extensive computer security considerations and Canada being part of the PREDICT country list, a dataset with real network traffic containing a DDoS attack has been made available for this research. It is important to highlight that this DDoS dataset is confidential. Hence, unauthorized or inadvertent use, disclosure dissemination, or publication of the raw dataset is prohibited. The access to this DDoS dataset is limited (based on an expiry date), non-exclusive, revocable, and non-transferable. Any attempt to translate, unlock, override, reverse engineer, or otherwise take any steps to defeat any anonymization or obfuscation methods or tools that may have been applied to the dataset in order to determine the identity of a specific individual shall not take place. If any product results from analysis related to this dataset, such product shall not include sensitive information derived from the dataset. Any publication product of research related to this DDoS dataset shall be forwarded to the United States Department of Homeland Security's PREDICT Project [Depa013]. Nevertheless, researchers interested in accessing this DDoS dataset are encouraged to contact the United States Department of Homeland Security's PREDICT Project directly to follow proper steps for security clearance for grating access.

Domain name system amplification reflection attacks involve an attacker sending a flood of DNS ANY requests. These requests focus on asking the DNS to provide ALL the information about the domain, which may include mail servers, MX records, IP addresses, A records, and so on and so forth. Attackers queries like this to maximize the response sent to the victim. These ANY requests are sent to one or several DNS servers, while spoofing source address to that of the intended target. A poorly configured recursive DNS server sends a much larger reply to the target, thus amplifying the attack. This DDoS dataset contains an attack between two sites: (i) The Information Sciences Institute at the University of Southern California (ISI/USC) located in Marina del Rey California, California and (ii) The Colorado State University (CSU) located in Fort Collins, Colorado. The ISI/USC hosted one attacker system (IP address: 145.233.157.236, which probably is not present in the traces) and six recursive DNS servers (IP addresses: 145.233.157.224, 145.233.157.228, 145.233.157.232, 145.233.157.233, 145.233.157.234, and 145.233.157.235), while the CSU provided a single system as an intended target (IP address:

144.154.222.228). All data files containing the DDoS attack recording are in extensible record format (ERF), compressed with bzip2 [Depa013].

In this DDoS attack, all non-attack traffic has been anonymized and scrubbed. Since the attack traffic was generated only as part of this experiment (completely under control), it is known to not have any privacy concerns, and we preserve payloads of traffic specific to the attack [Depa013].

Attack queries are replayed at 400 packets per second, each packet containing a UDP DNS query, which is directed to one of the six ISI/USC servers in a round-robin fashion. Each IP packet is 64 bytes long, thus the bit rate of the attack before amplification/reflection is 64*400*8=205 Kbps. The data collection starts at 21:52:45 and ends at 22:25:32 on 17 June 2013. Within this window of time, the DDoS attack starts at 22:00:12 and concludes at 22:15:34. This dataset contains 59,928,921 packets from which the following relevant information is available: Time stamp, source, destination, length, and protocol [Depa013].

APPENDIX G

RESULTS OF SELECTED PRIMARY OPERATORS APPLIED THROUGH MULTISCALORS

G.1 Variance

The results obtained by the variance multiscalor visually resembles in all its components (from first ($m_{2|||^1}$) to seventh ($m_{2|||^7}$)) both of the DDoS attacks, the DNS amplification and the H&R, that are present in the dataset and can be seen in Figs. L.1 to L.7 respectively.

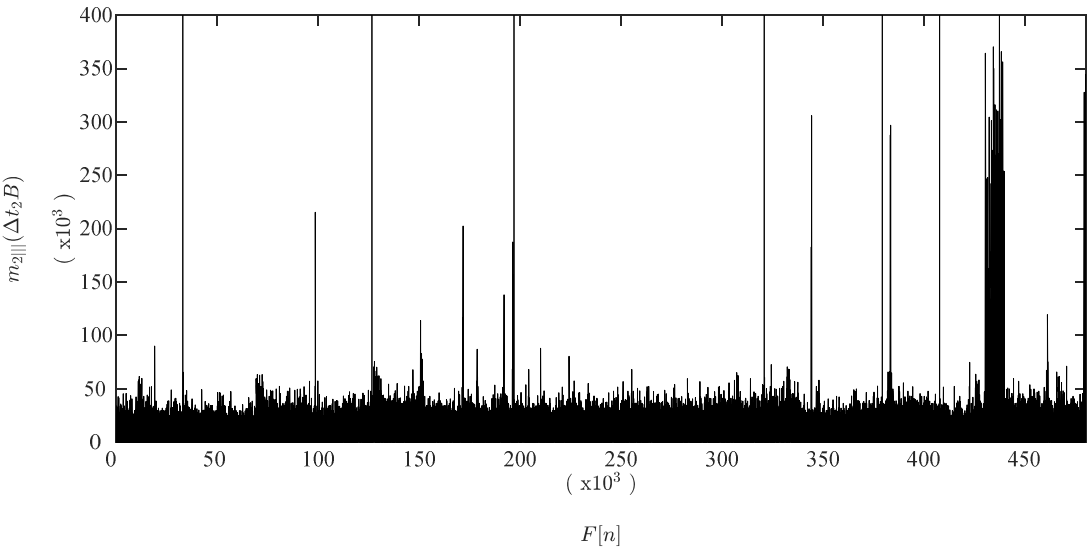


Fig. G.1. Variance multiscalor 1st component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 2 ($\Delta t_2 B$) samples are used.

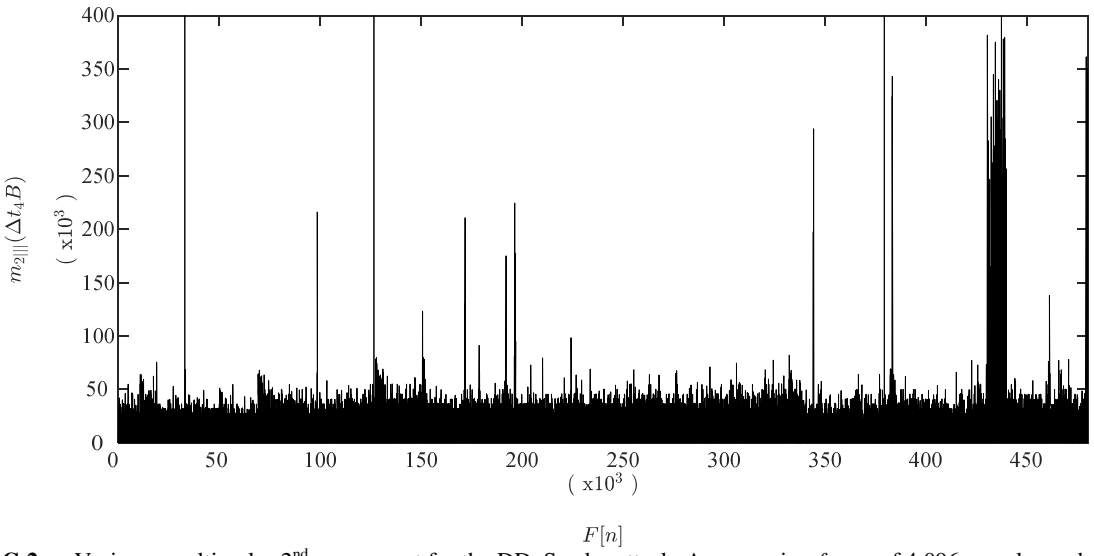


Fig. G.2. Variance multiscalor 2nd component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 4 $(\Delta t_4 B)$ samples are used.

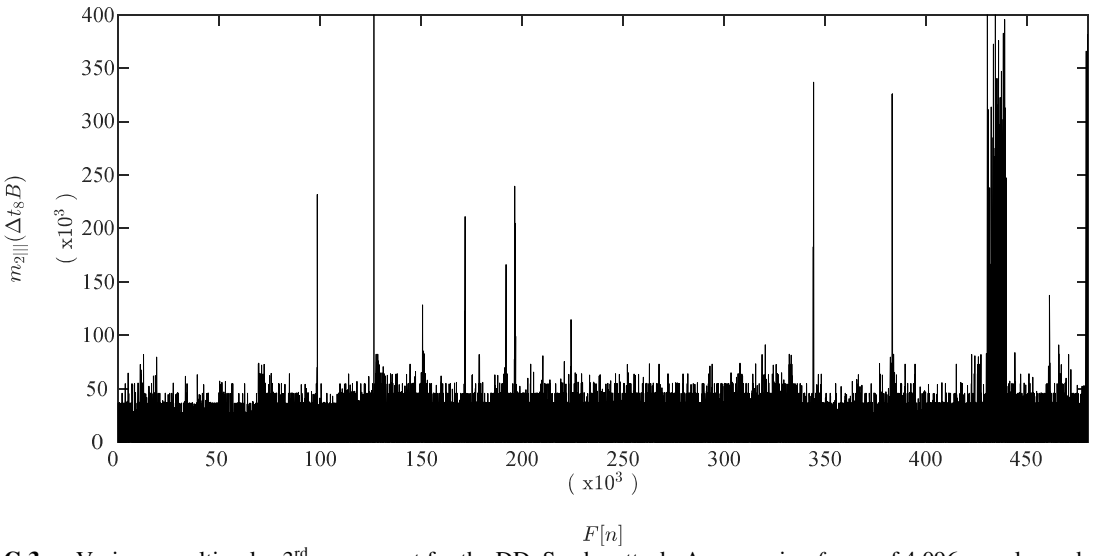


Fig. G.3. Variance multiscalor 3rd component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 8 $(\Delta t_8 B)$ samples are used.

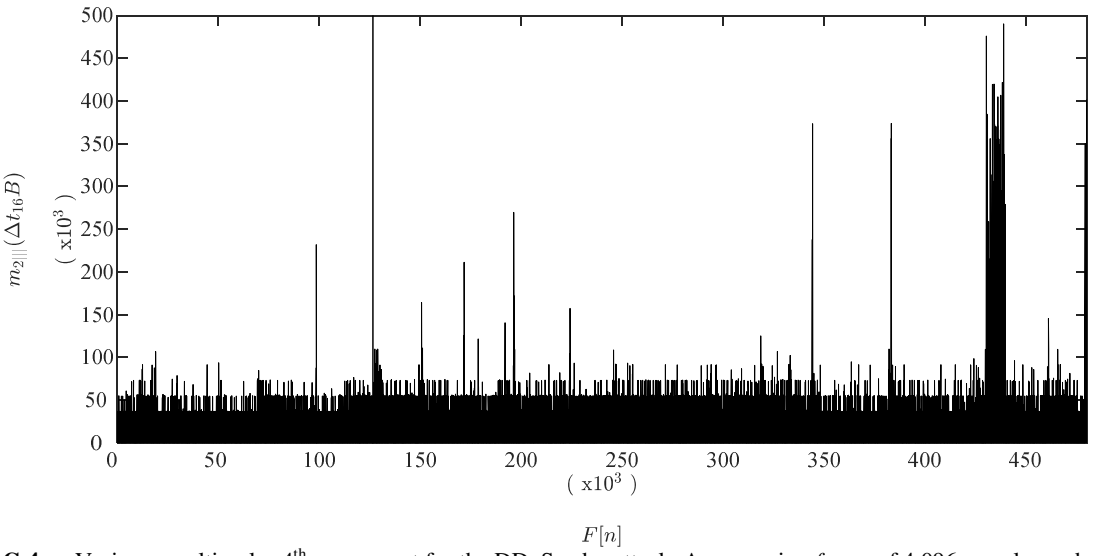


Fig. G.4. Variance multiscalor 4th component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 16 $(\Delta t_{16}B)$ samples are used.

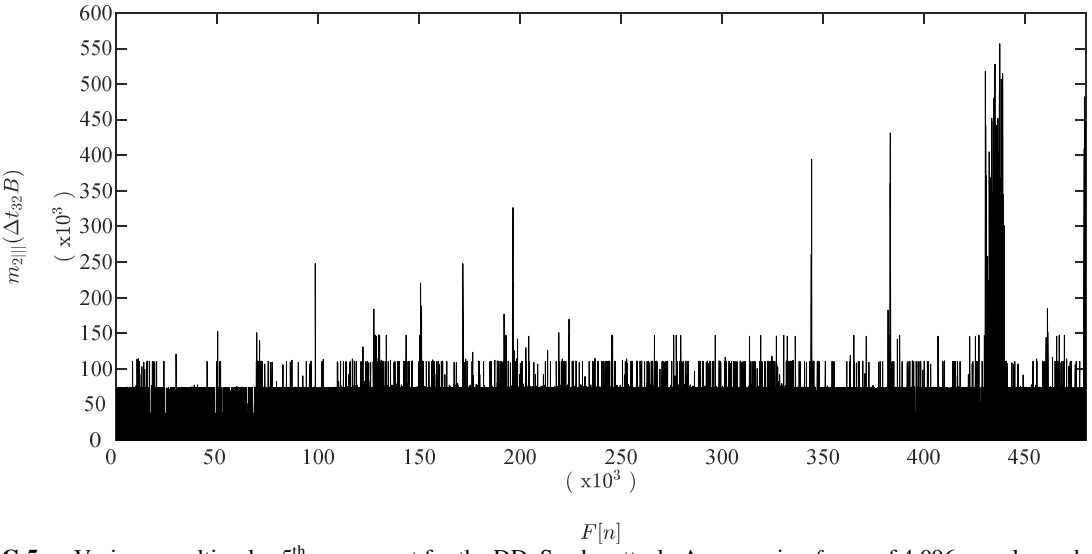


Fig. G.5. Variance multiscalor 5th component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 32 $(\Delta t_{32}B)$ samples are used.

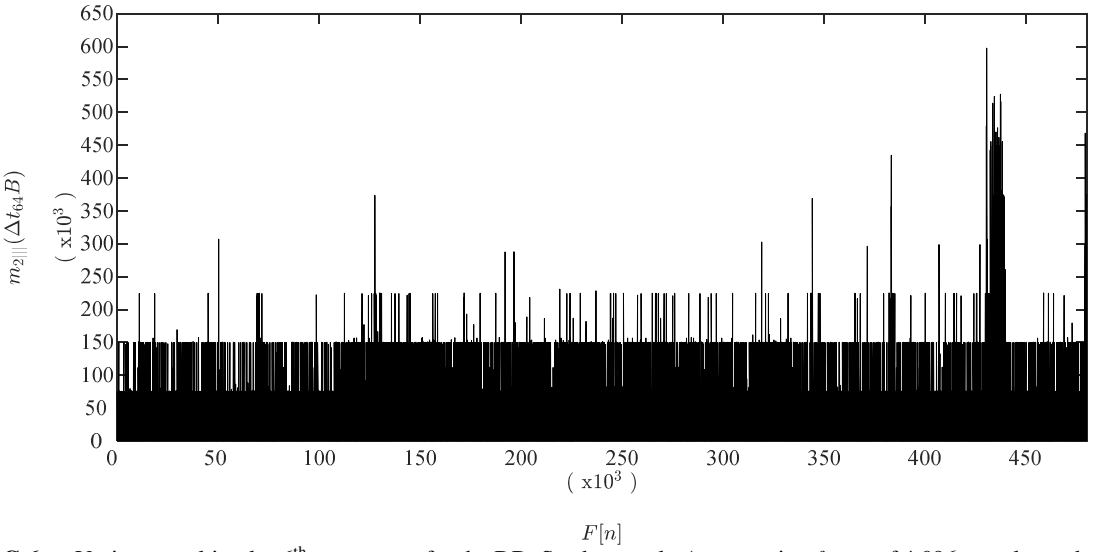


Fig. G.6. Variance multiscalor 6th component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 64 $(\Delta t_{64} B)$ samples are used.

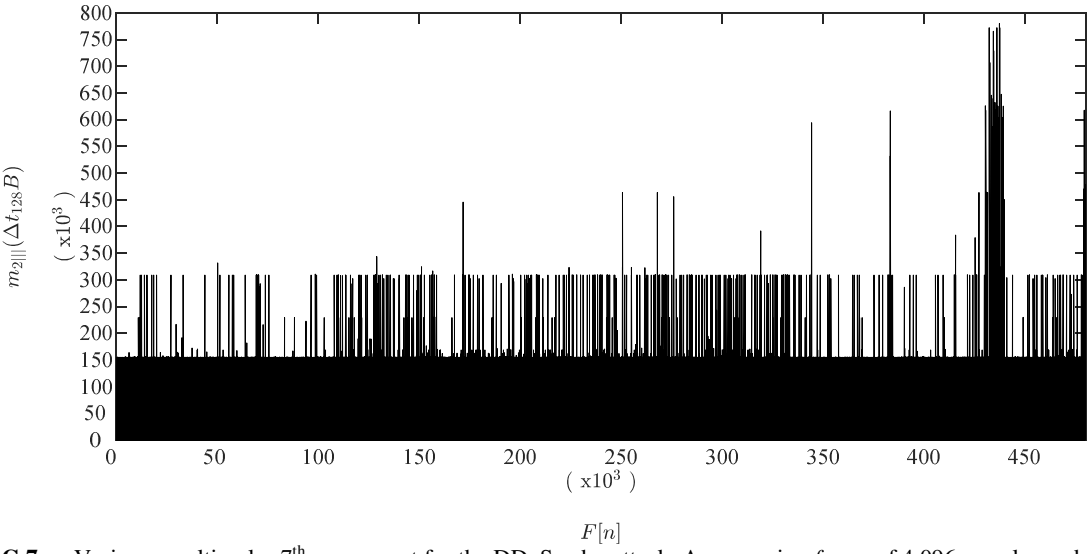


Fig. G.7. Variance multiscalor 7th component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 128 $(\Delta t_{128} B)$ samples are used.

G.2 Skewness

The skewness multiscalor visual insights are not as clear as the ones provided by the variance multiscalor previously. Specifically, the DNS amplification DDoS attack is visibly identifiable in the first ($m_{3|||^1}$), second ($m_{3|||^2}$), and third ($m_{3|||^3}$) skewness multiscalor components while the H&R DDoS attack is remarkably weak in any components of the skewness multiscalor.

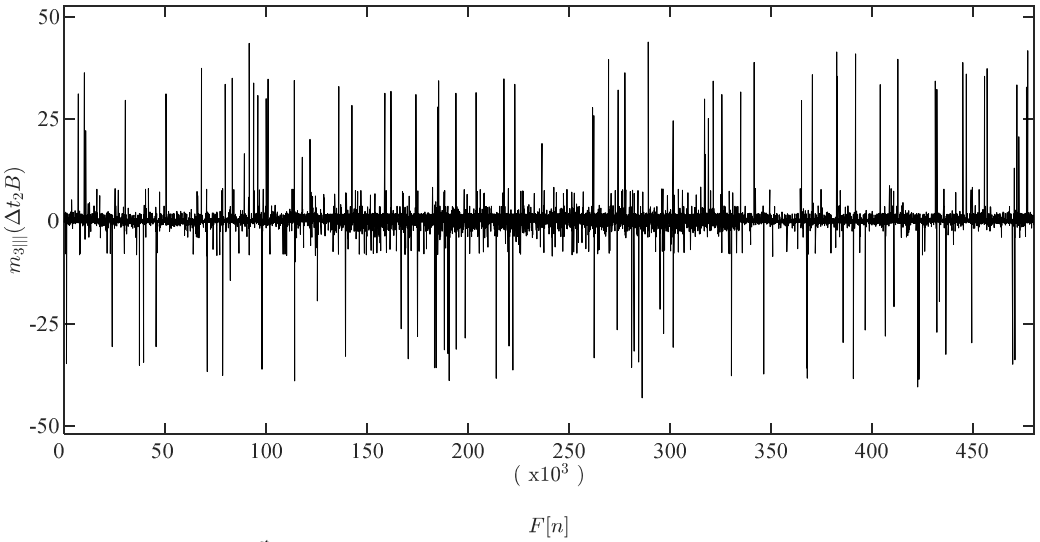


Fig. G.8. Skewness multiscalor 1st component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 2 ($\Delta t_2 B$) samples are used.

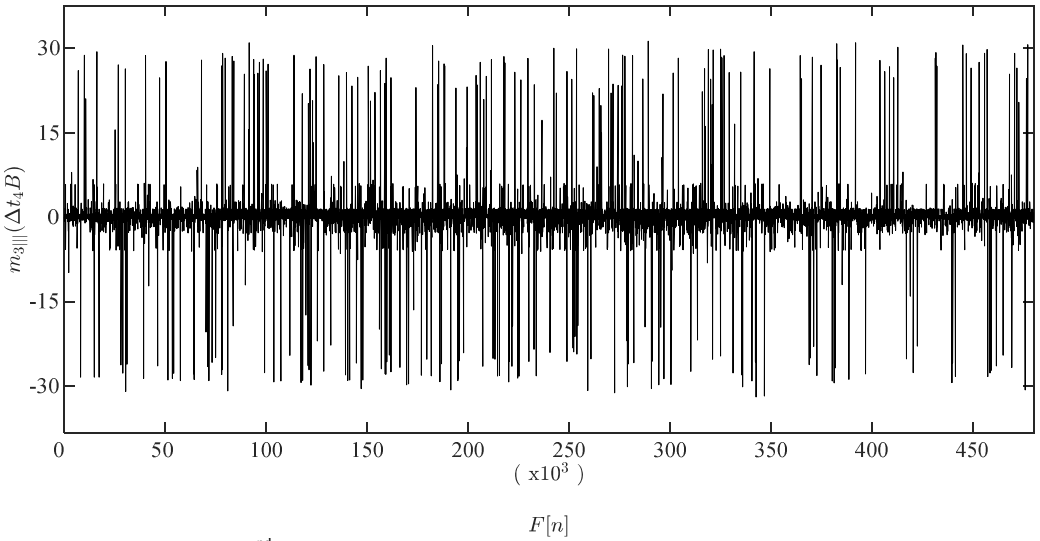


Fig. G.9. Skewness multiscalor 2nd component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 4 ($\Delta t_4 B$) samples are used.

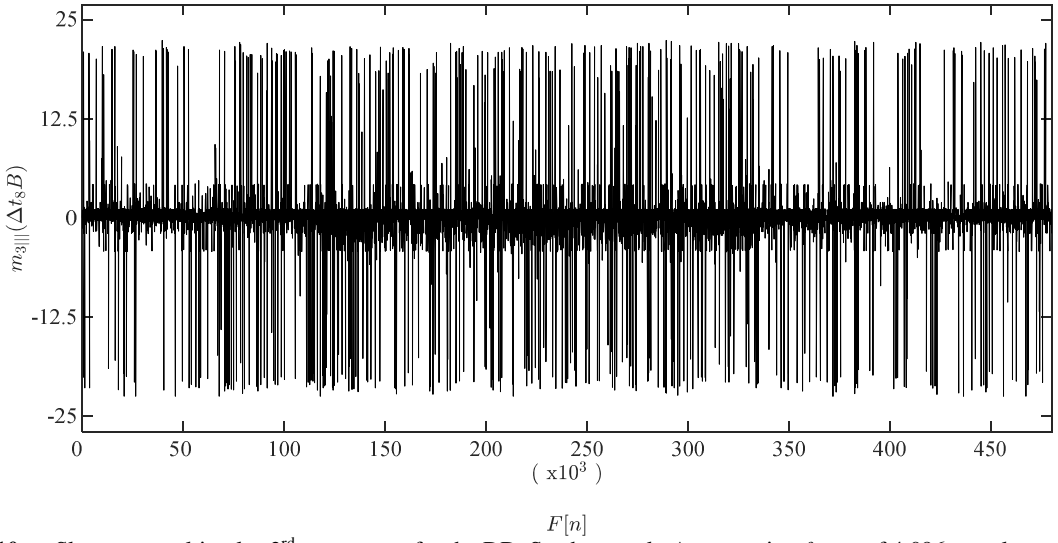


Fig. G.10. Skewness multiscalor 3rd component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 8 $(\Delta t_8 B)$ samples are used.

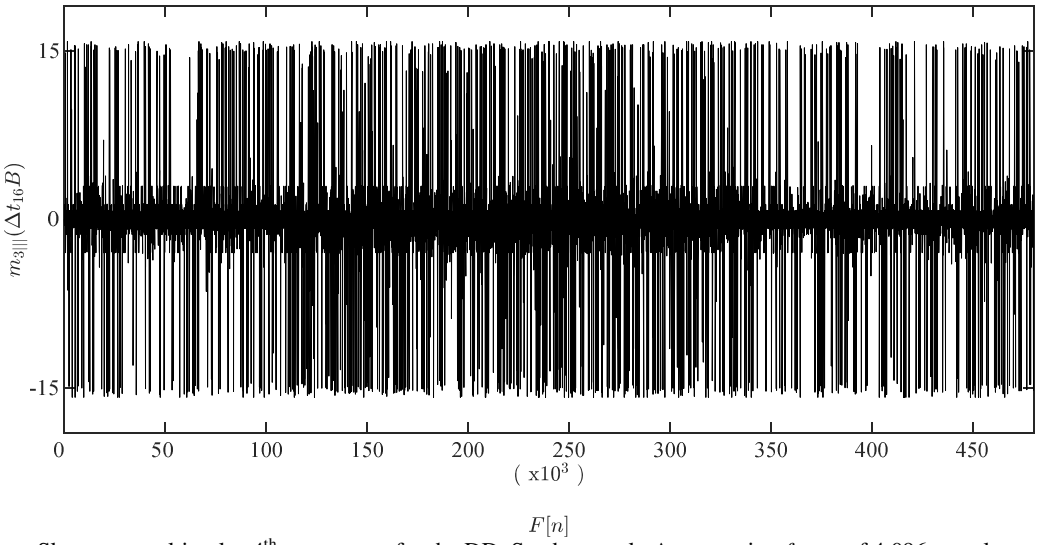


Fig. G.11. Skewness multiscalor 4th component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 16 $(\Delta t_{16} B)$ samples are used.

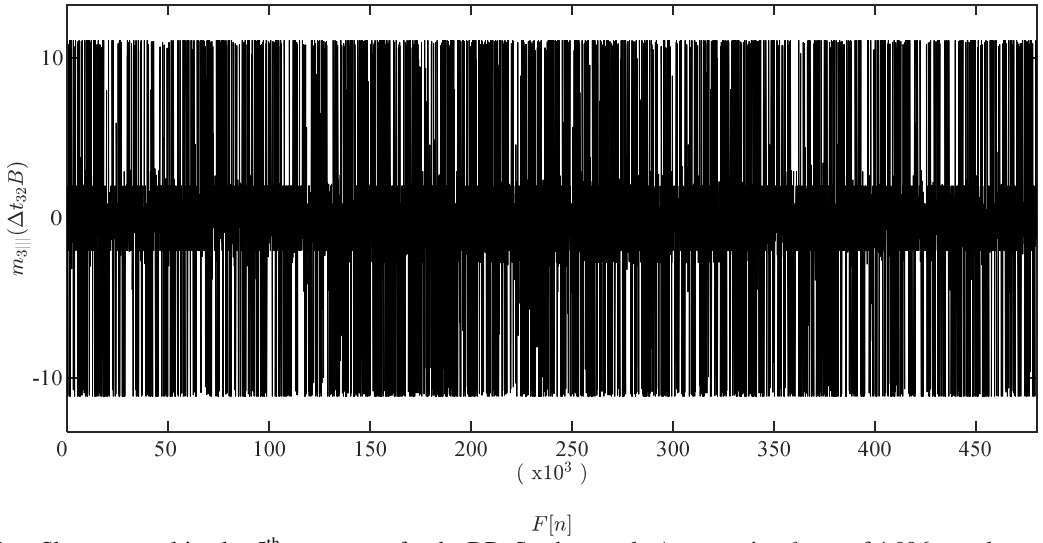


Fig. G.12. Skewness multiscalor 5th component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 32 $(\Delta t_{32}B)$ samples are used.

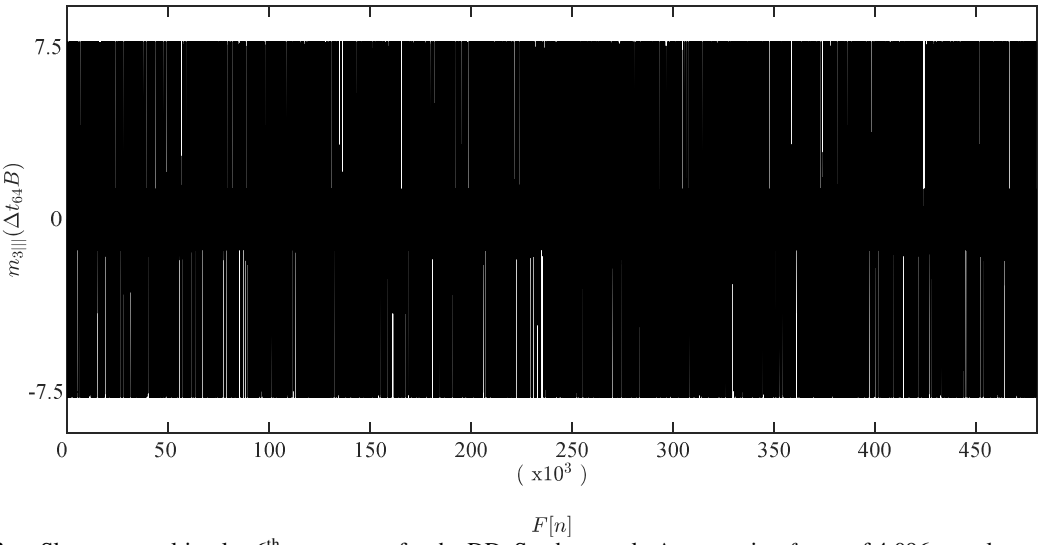


Fig. G.13. Skewness multiscalor 6th component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 64 $(\Delta t_{64}B)$ samples are used.

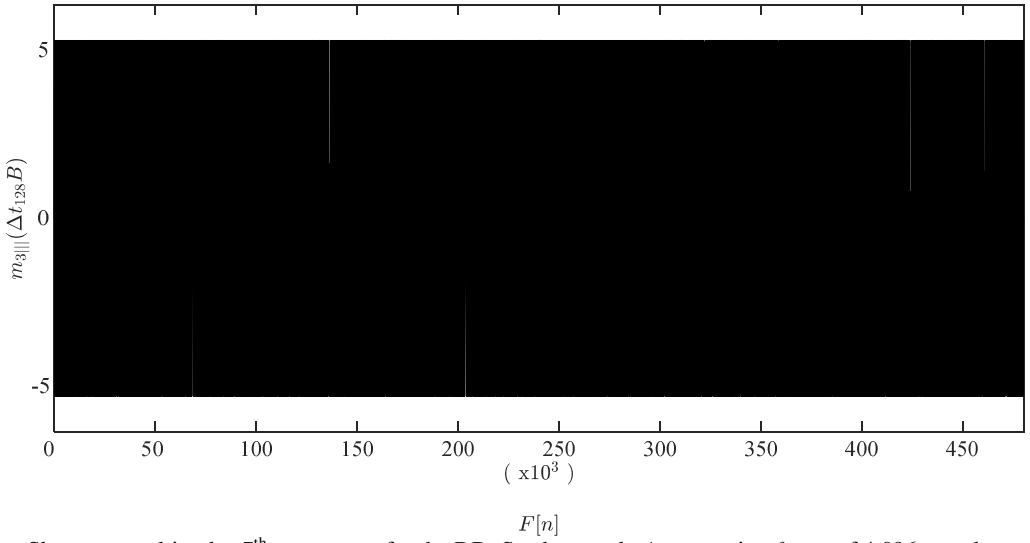


Fig. G.14. Skewness multiscalor 7th component for the DDoS cyberattack. A processing *frame* of 4,096 samples and a *vel* size of 128 $(\Delta t_{128}B)$ samples are used.

APPENDIX H

RESULTS OF SELECTED SECONDARY OPERATORS APPLIED TO VARIANCE MULTISCALOR

H.1 Cumulative Sum

H.1.1 Cumulative Sum Applied to Variance Multiscalor Components

The cumulative sum applied to each variance multiscalor component ($S[m_{2\text{III}^n}]$) provides meaningful results for all variance multiscalor components (from first, $m_{2\text{III}^1}$, to seventh, $m_{2\text{III}^7}$) as seen in Figs. H.1 to H.7. The two DDoS attacks, DNS amplification and H&R, can be seen in the seven components of the variance multiscalor. The visual quality of the results appears to decrease as one traverses from the first multiscalor component, $m_{2\text{III}^1}$, to the seventh, $m_{2\text{III}^7}$. The latter showing bigger fluctuations in amplitude. Nonetheless, the shapes of both DDoS attacks are preserved within all results of the cumulative sum run on all variance multiscalor components. A number of minor spikes are visible across the traffic. These would correspond to either traffic transients or DDoS attack precursors (tests carried prior to launching a full force attack).

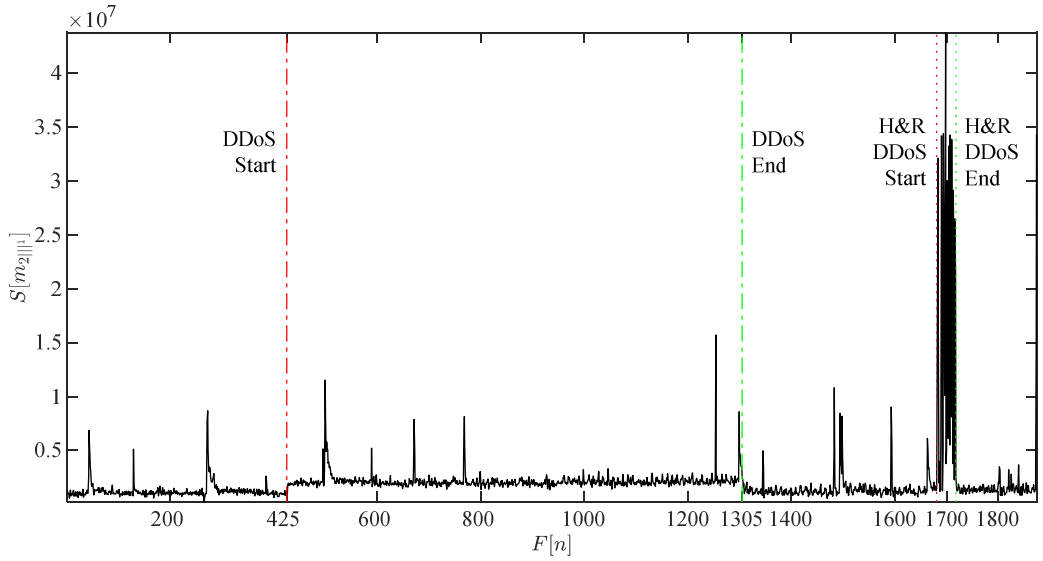


Fig. H.1. Cumulative sum *S* applied to the variance multiscalor 1st component ($m_{2||^1}$). A processing *frame* of 256 samples is used.

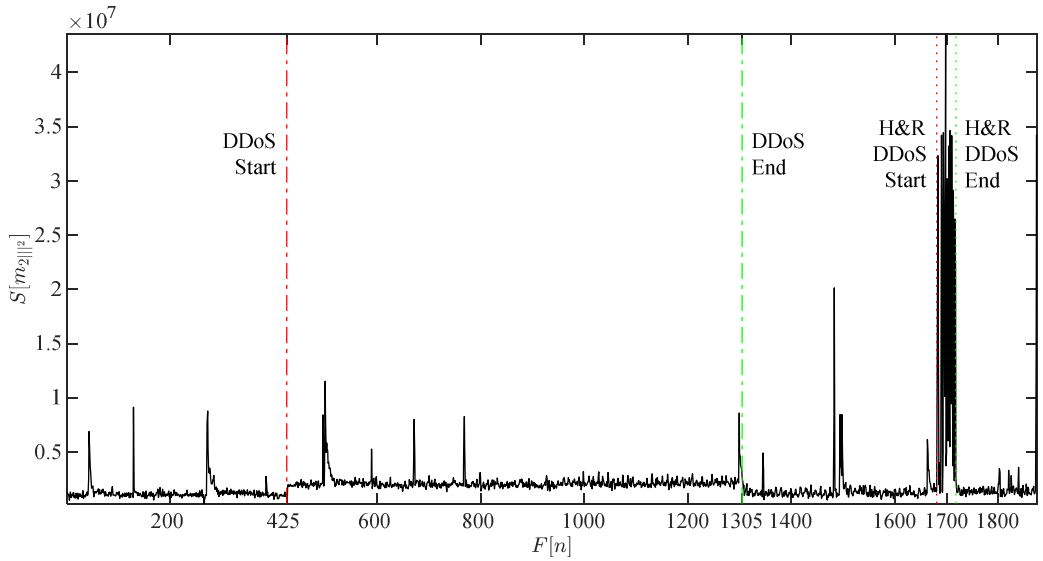


Fig. H.2. Cumulative sum *S* applied to the variance multiscalor 2nd component ($m_{2||^2}$). A processing *frame* of 256 samples is used.

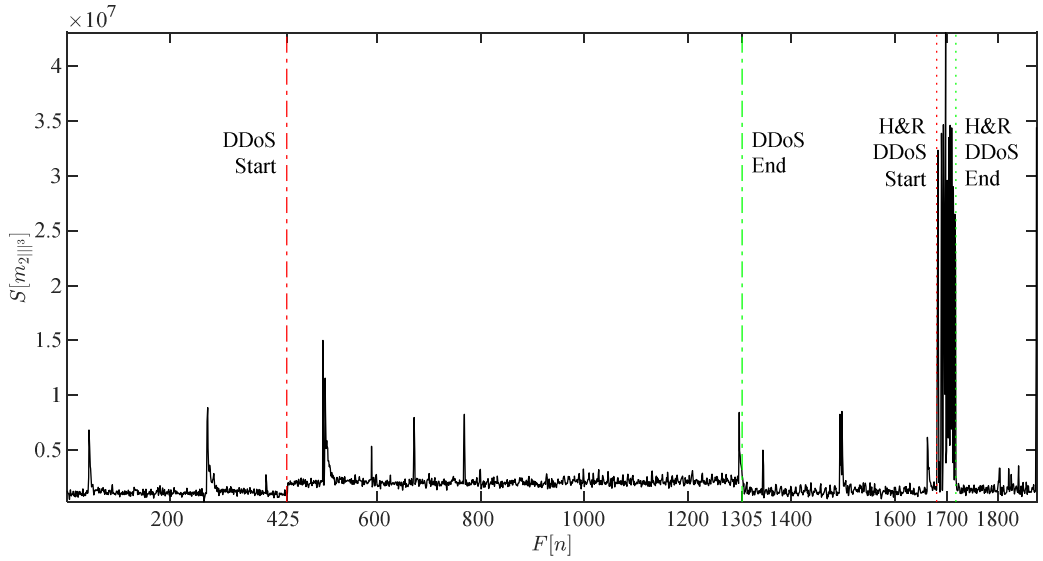


Fig. H.3. Cumulative sum *S* applied to the variance multiscalor 3rd component $(m_{2\|\|^3})$. A processing *frame* of 256 samples is used.

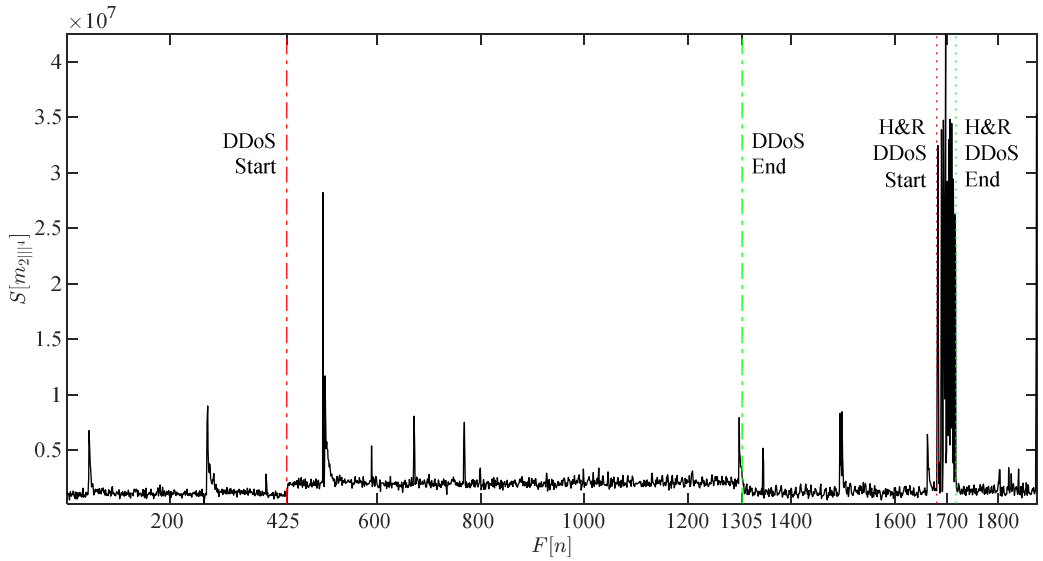


Fig. H.4. Cumulative sum *S* applied to the variance multiscalor 4th component $(m_{2\|\|^4})$. A processing *frame* of 256 samples is used.

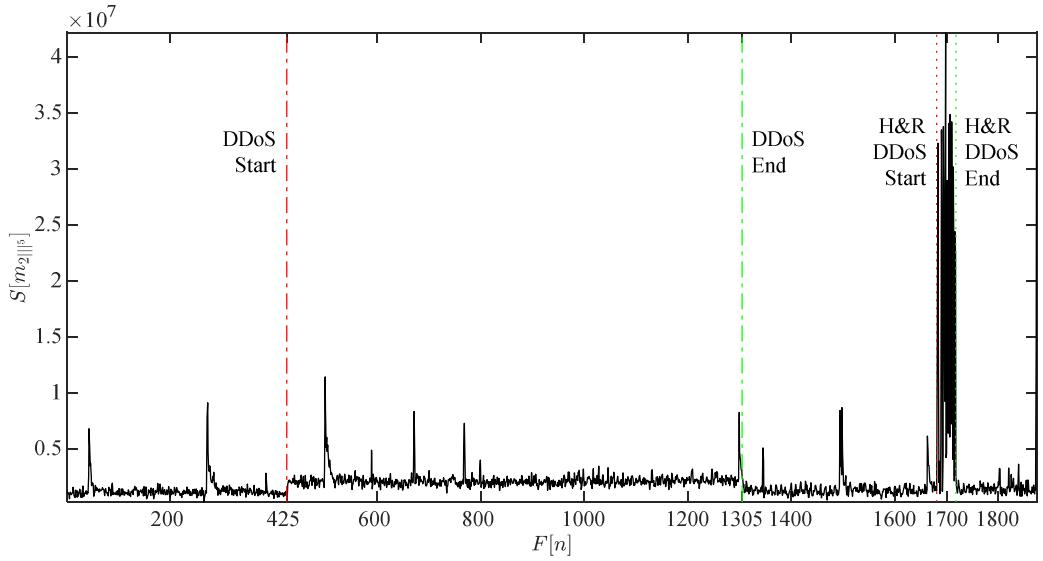


Fig. H.5. Cumulative sum *S* applied to the variance multiscalor 5th component ($m_{2\|\|^5}$). A processing *frame* of 256 samples is used.

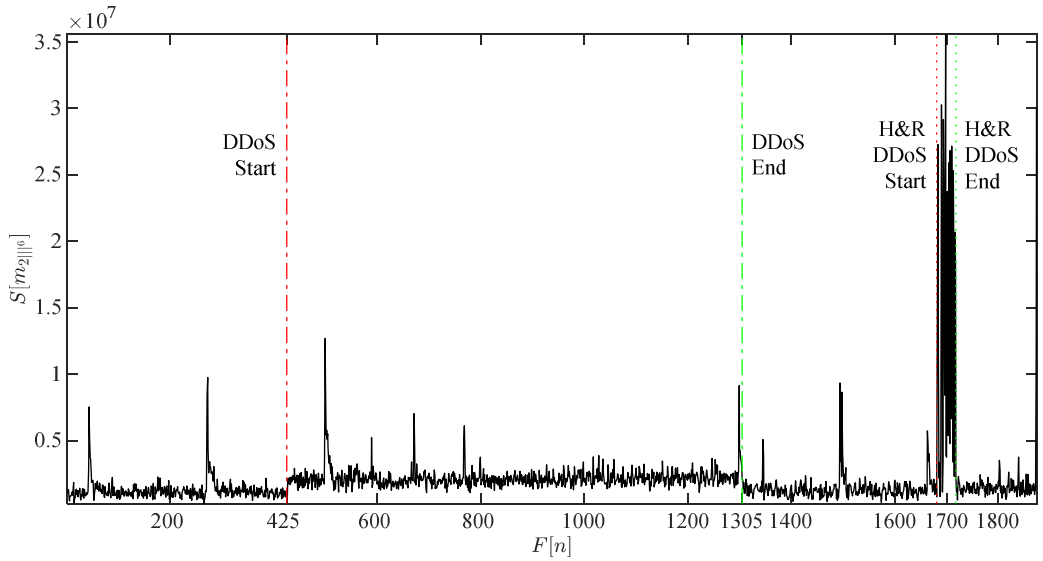


Fig. H.6. Cumulative sum *S* applied to the variance multiscalor 6th component ($m_{2\|\|^6}$). A processing *frame* of 256 samples is used.

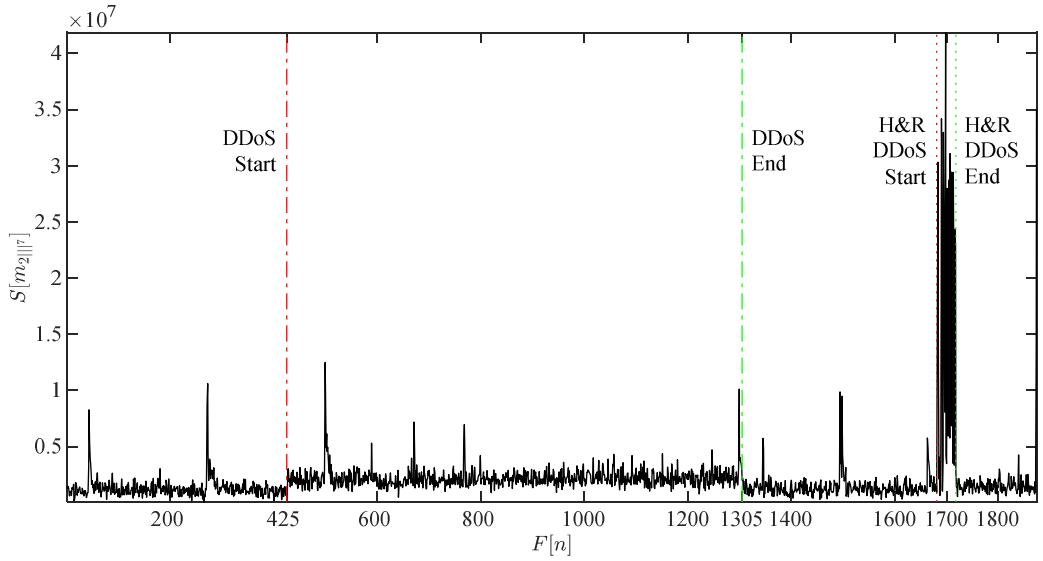


Fig. H.7. Cumulative sum *S* applied to the variance multiscalor 7th component ($m_{2|||^7}$). A processing *frame* of 256 samples is used.

H.1.2 Cumulative Sum Applied to Variance Multiscalor Components After Donoho's Denoising

Figures H.8 to H.14 depict Donoho's denoising applied to the outcome of cumulative sum after processing variance multiscalor components. It is seen that noise has been reduced from the first, $m_{2|||^1}$, to seventh, $m_{2|||^7}$, components. Both DDoS attacks, DNS amplification and H&R, are seen more nitidly in the seven components of the variance multiscalor. The visual quality in the results presented in Figs. H.8 to H.14 appears to be constant from the first multiscalor component, $m_{2|||^1}$, to the seventh, $m_{2|||^7}$. The shape of both attacks has been maintained. The minor spikes across the traffic are still visible.

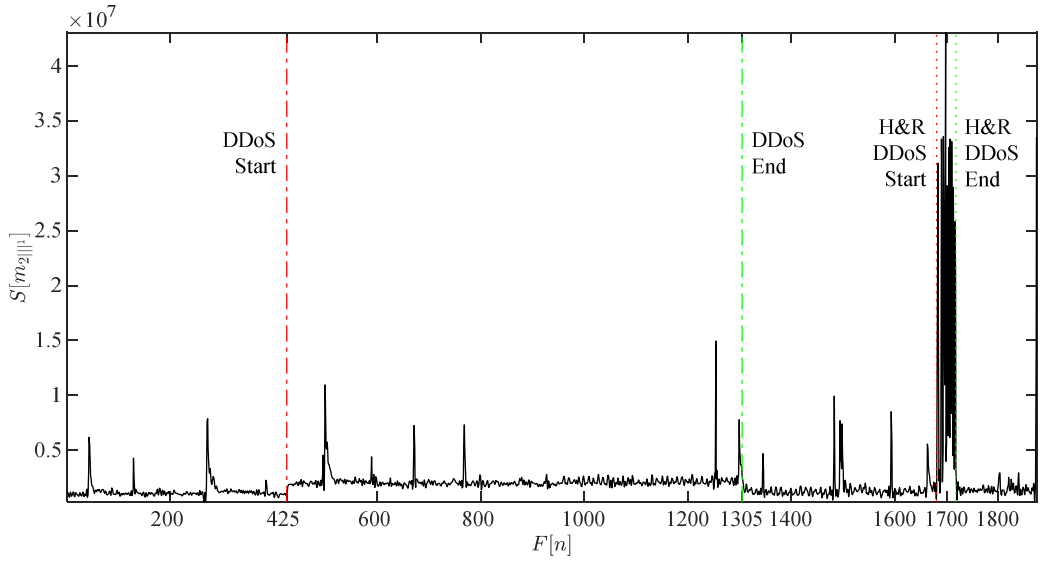


Fig. H.8. Cumulative sum *S* applied to the variance multiscalor 1st component ($m_{2\|\|^1}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

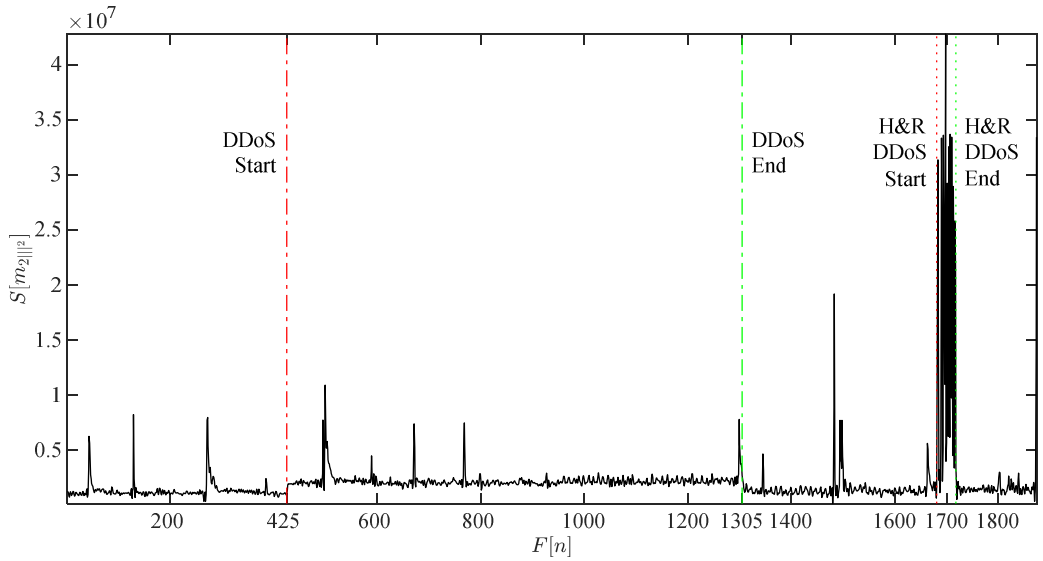


Fig. H.9. Cumulative sum *S* applied to the variance multiscalor 2nd component ($m_{2\|\|^2}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

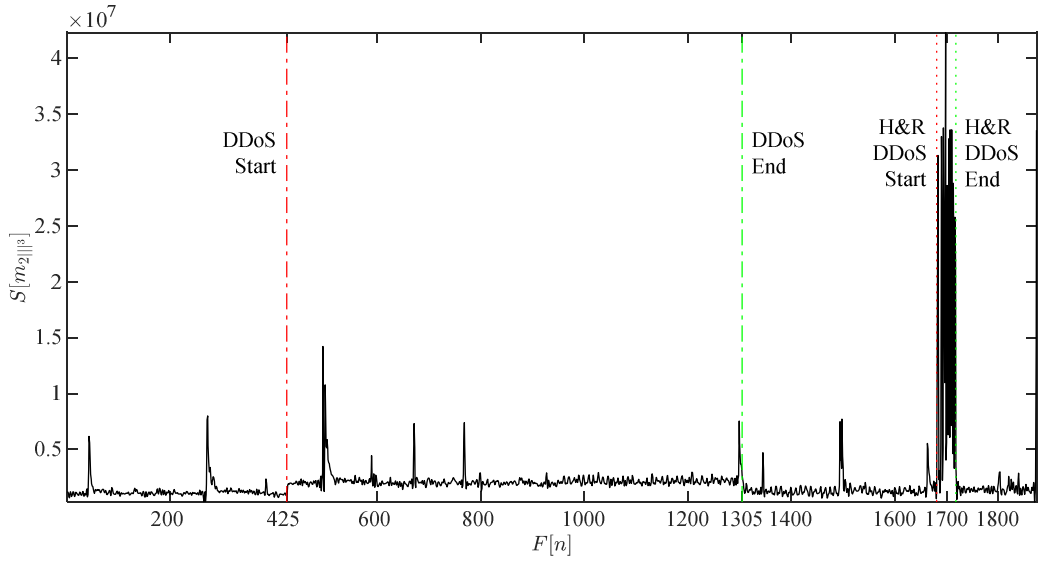


Fig. H.10. Cumulative sum *S* applied to the variance multiscalor 3rd component ($m_{2||^3}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

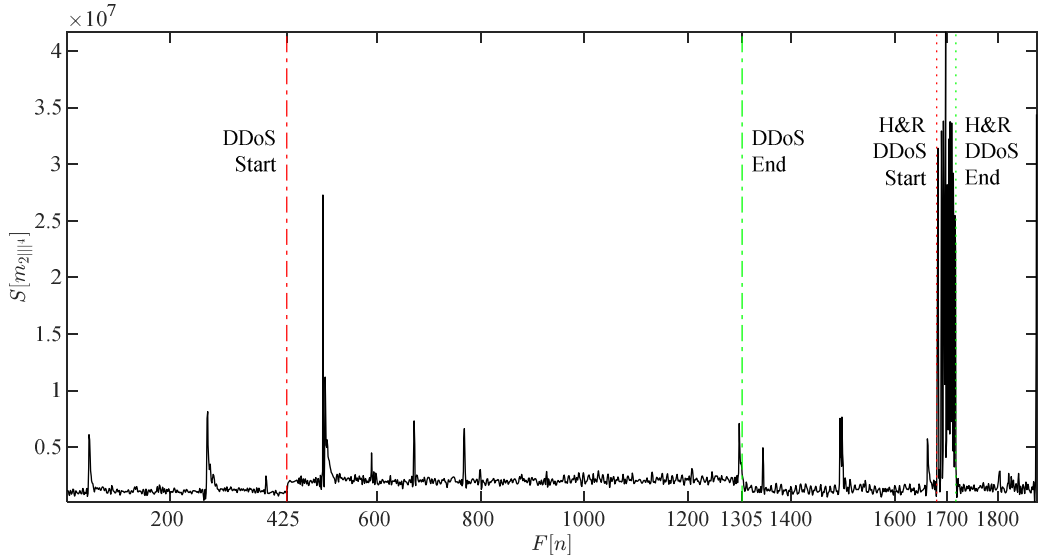


Fig. H.11. Cumulative sum *S* applied to the variance multiscalor 4th component ($m_{2||^4}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

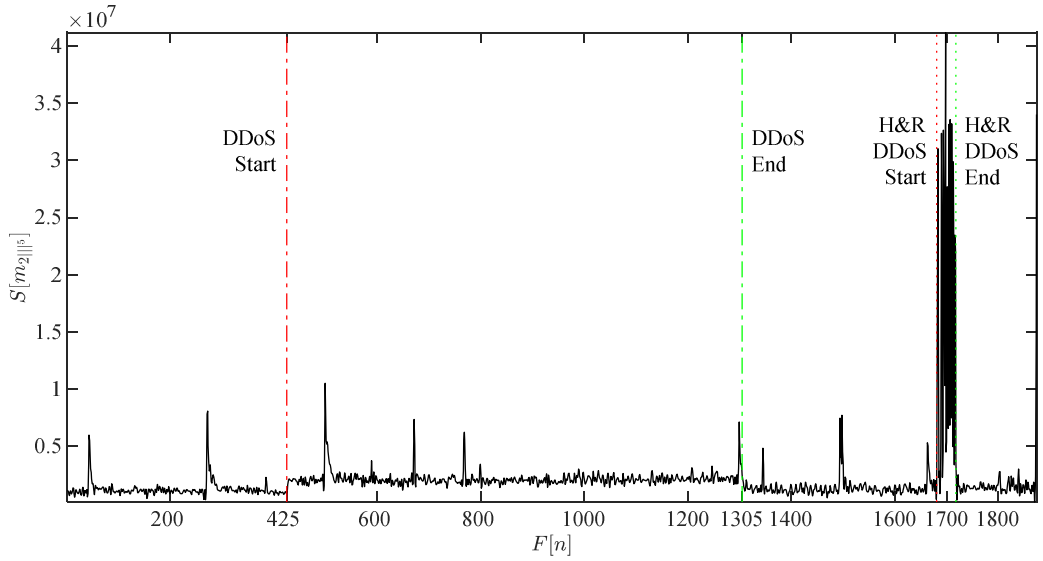


Fig. H.12. Cumulative sum *S* applied to the variance multiscalor 5th component ($m_{2||^5}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

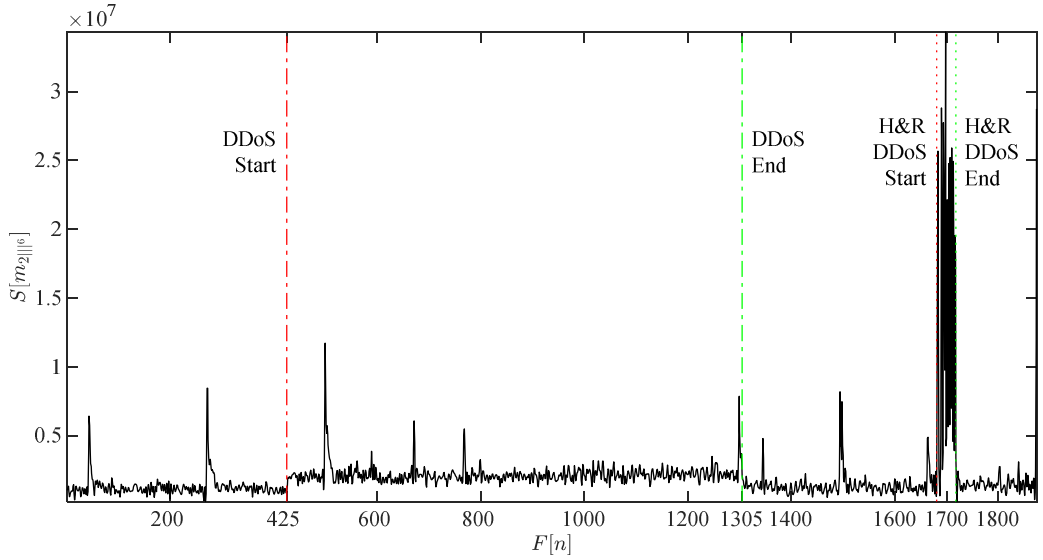


Fig. H.13. Cumulative sum *S* applied to the variance multiscalor 6th component ($m_{2||^6}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

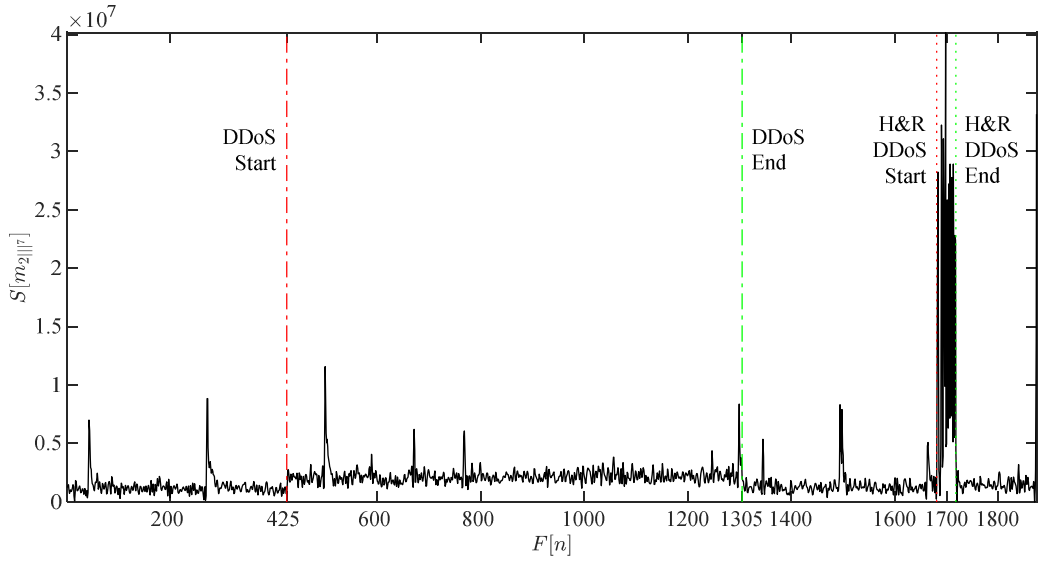


Fig. H.14. Cumulative sum *S* applied to the variance multiscalor 7th component ($m_{2|||^7}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

H.1.3 Cumulative Sum Applied to Variance Multiscalor Components Non-Linearly Filtered After Donoho's Denoising

A subsequent nonlinear filtering stage is applied to the variance multiscalor components after Donoho's denoising. Figures H.15 to H.21 show the outcome of this processing stage for the first, $m_{2|||^1}$, to seventh, $m_{2|||^7}$, components respectively. A further reduction in noise is seen for all cases and all waveforms appear smooth. Both DDoS attacks, DNS amplification and H&R, are seen clearly in the seven components of the variance multiscalor. The visual quality in the outcomes introduced in Figs. M.15 to M.21 appears to be persistent from the first variance multiscalor component, $m_{2|||^1}$, to the seventh, $m_{2|||^7}$. Nonetheless, the seventh variance multiscalor component, $m_{2|||^7}$, is wigglier than the previous six counterparts. The shape of both attacks has been maintained. Some of the minor spikes across the traffic have been removed.

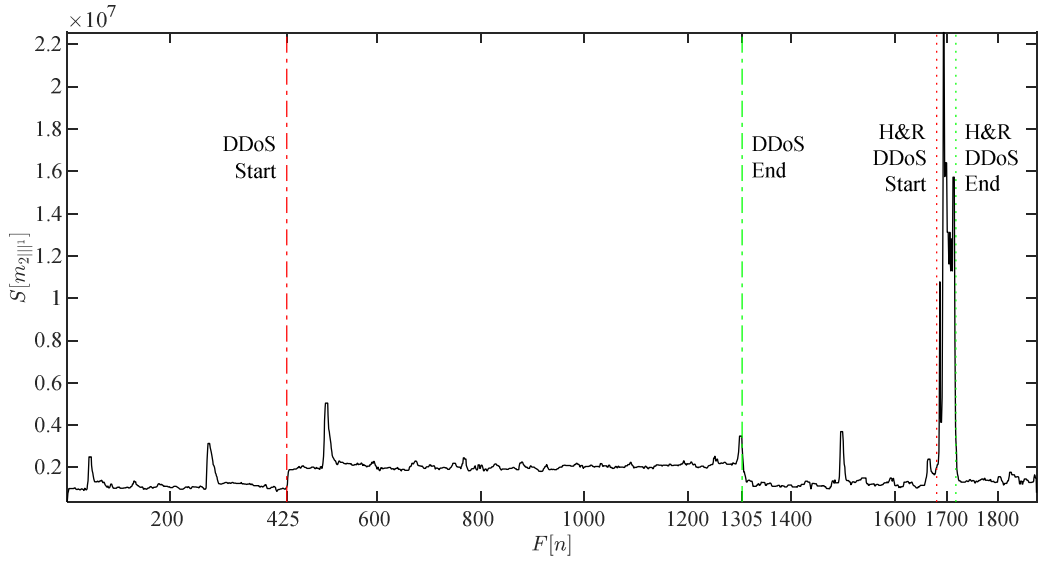


Fig. H.15. Cumulative sum *S* applied to the variance multiscalor 1st component ($m_{2||^1}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

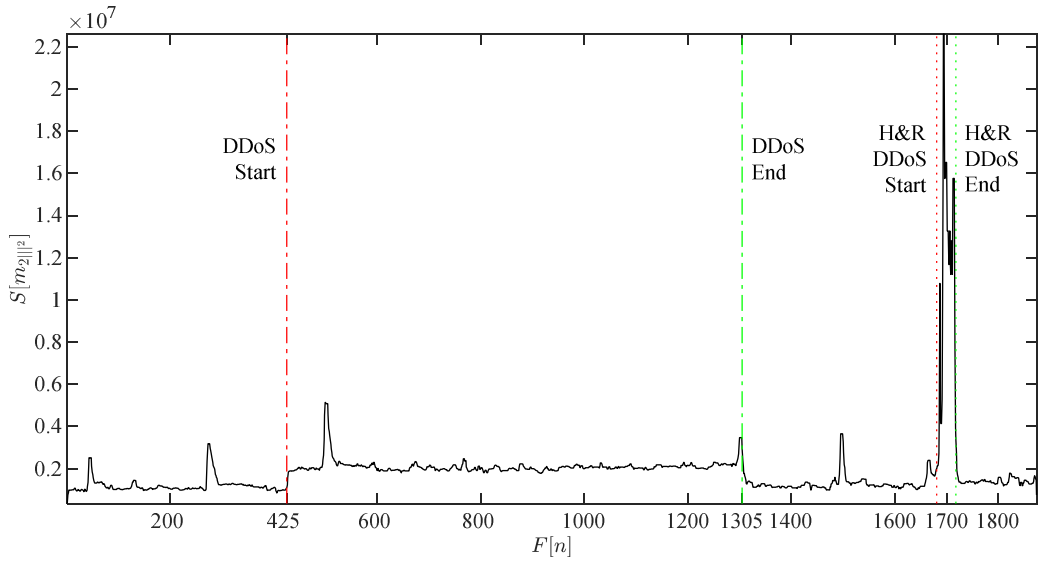


Fig. H.16. Cumulative sum *S* applied to the variance multiscalor 2nd component ($m_{2||^2}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

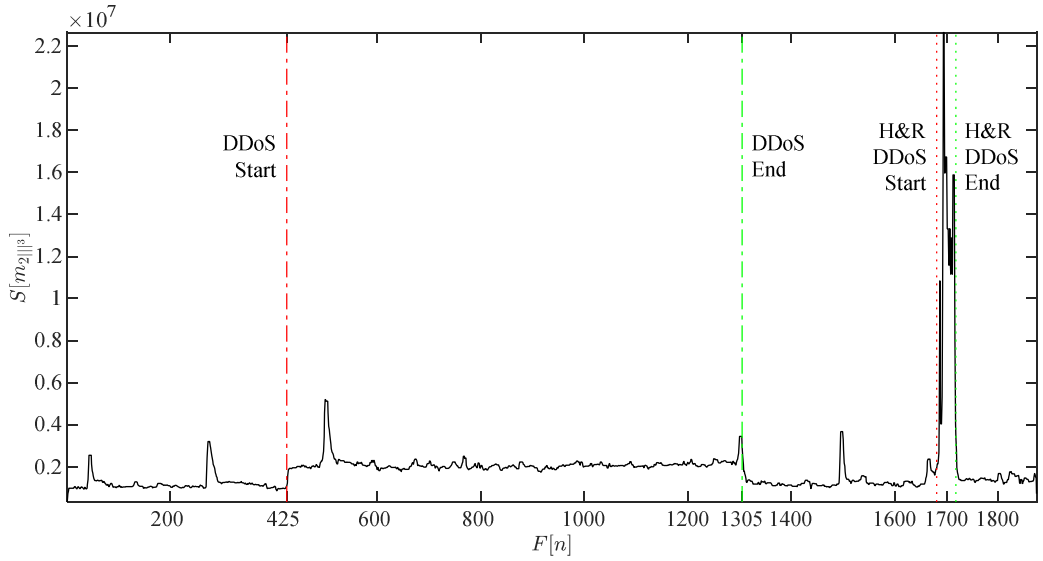


Fig. H.17. Cumulative sum *S* applied to the variance multiscalor 3rd component ($m_{2||^3}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

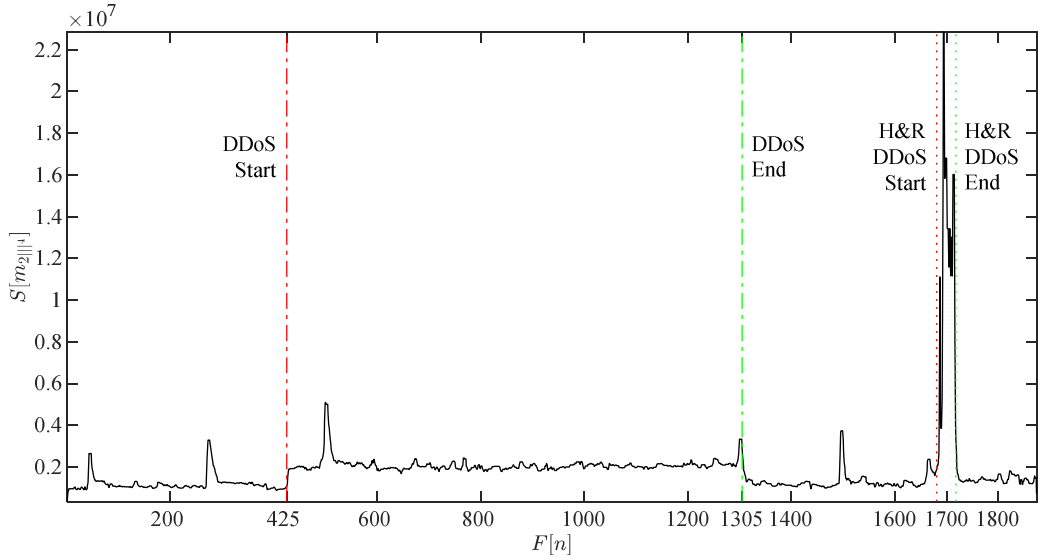


Fig. H.18. Cumulative sum *S* applied to the variance multiscalor 4th component ($m_{2||^4}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

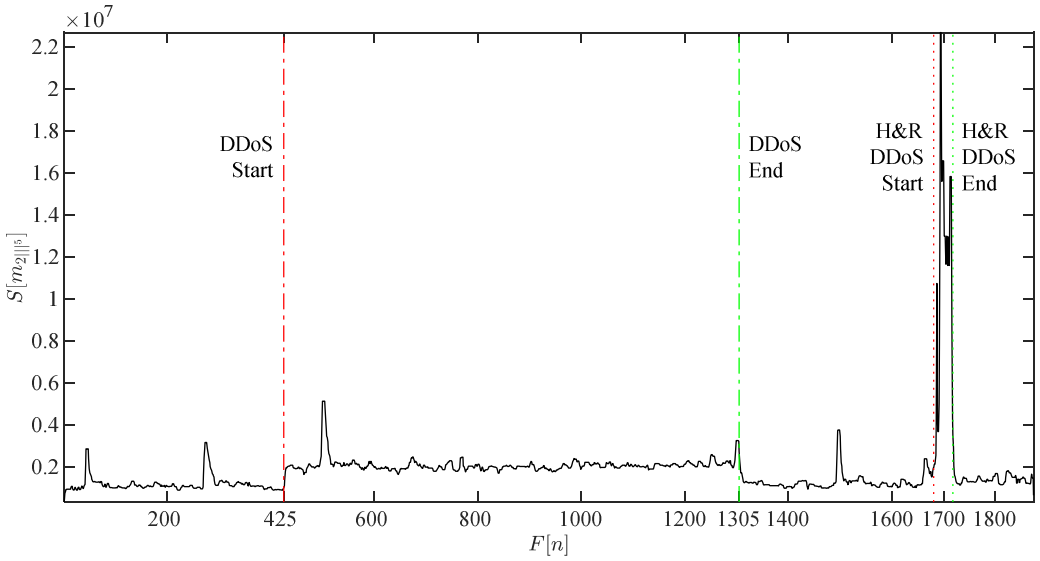


Fig. H.19. Cumulative sum *S* applied to the variance multiscalor 5th component ($m_{2||^5}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

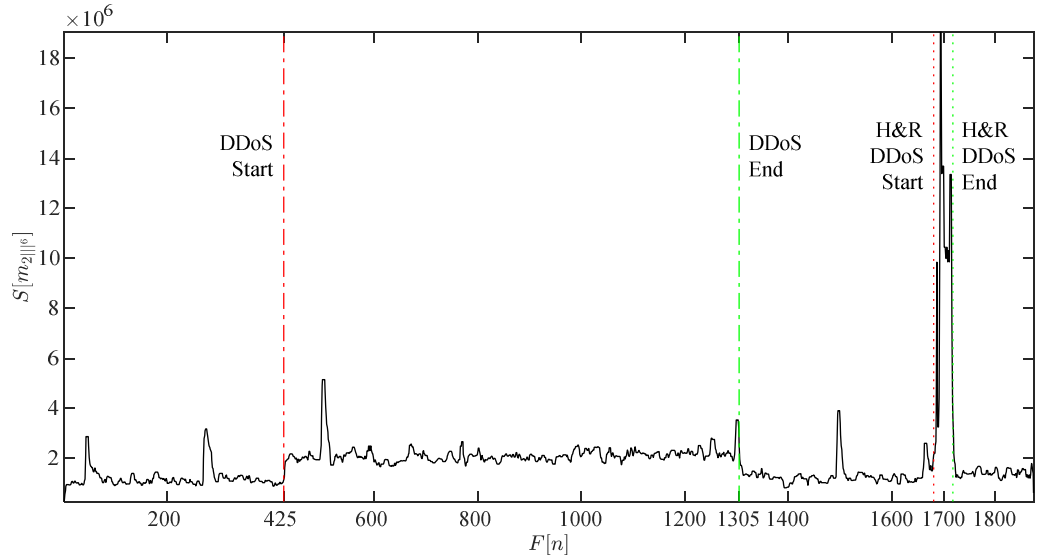


Fig. H.20. Cumulative sum *S* applied to the variance multiscalor 6th component ($m_{2||^6}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

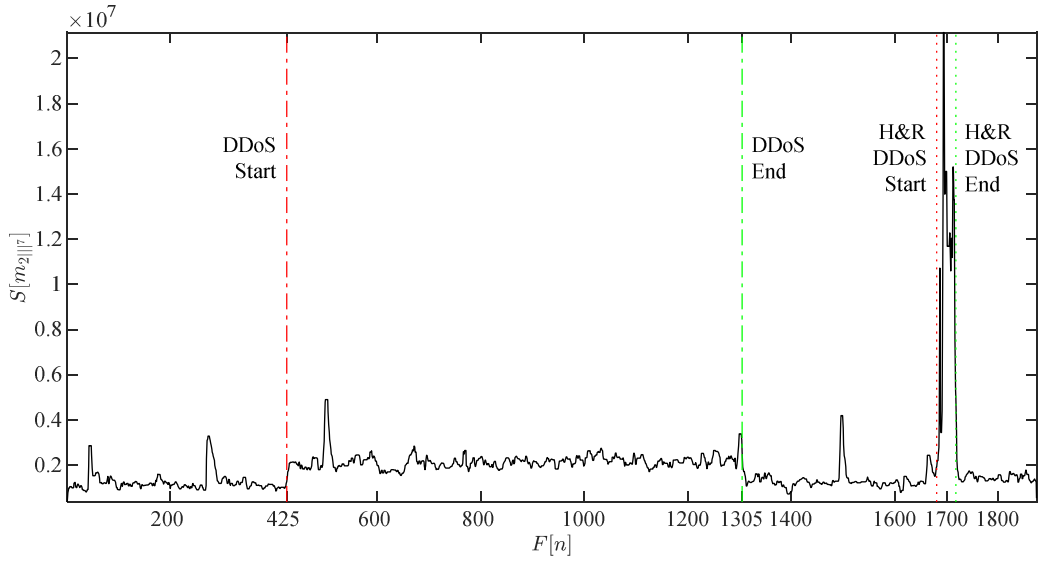


Fig. H.21. Cumulative sum *S* applied to the variance multiscalor 7th component ($m_{2|||^7}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

H.1.4 Cumulative Sum Applied to Variance Multiscalor Components Quantization of Non-Linear Filtering After Donoho's Denoising

In order to ready the extracted features for machine learning processing through ART1, a quantization phase to aid the conversion of the waveform into defined amplitude level codes, which can easy the translation to binary codes, is required. Quantization through Lloyd's methodology is shown in Figs. H.22 to H.28 for the first, $m_{2|||^1}$, to seventh, $m_{2|||^7}$, components respectively, where both DDoS attacks, DNS amplification and H&R, appear remarkably clear from the first, $m_{2|||^1}$, to the fifth, $m_{2|||^5}$, variance multiscalor components. The last two variance multiscalor components sixth, $m_{2|||^6}$, and seventh, $m_{2|||^7}$, also show the dynamics for both DDoS attacks. These last two components show oscillations in two of the quantized amplitude levels as a consequence of them being wigglier. Remarkably, the shape of both attacks appears very clear. Some of the minor spikes across the traffic have been translated into narrow pulses.

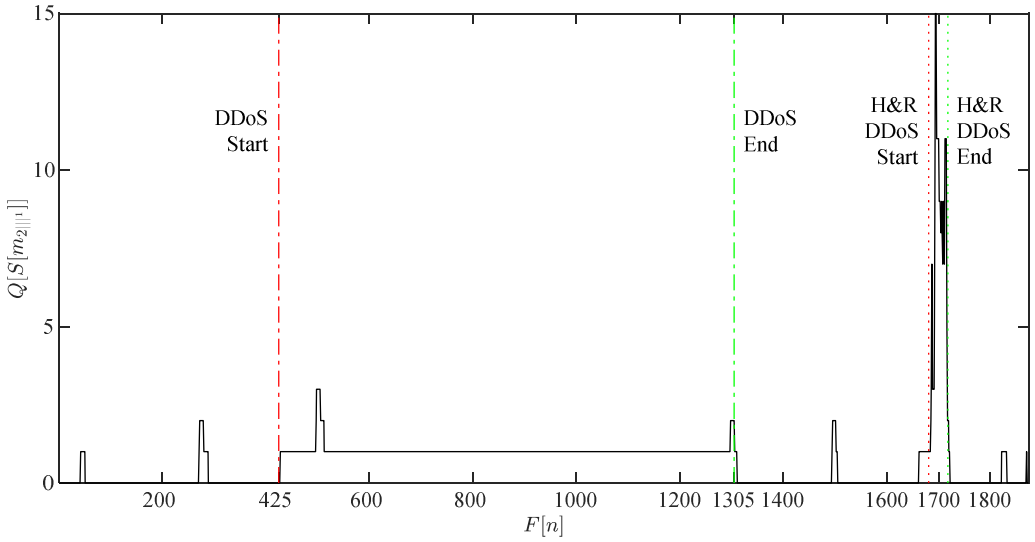


Fig. H.22. Cumulative sum *S* applied to the variance multiscalor 1st component ($m_{2\|\|^1}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

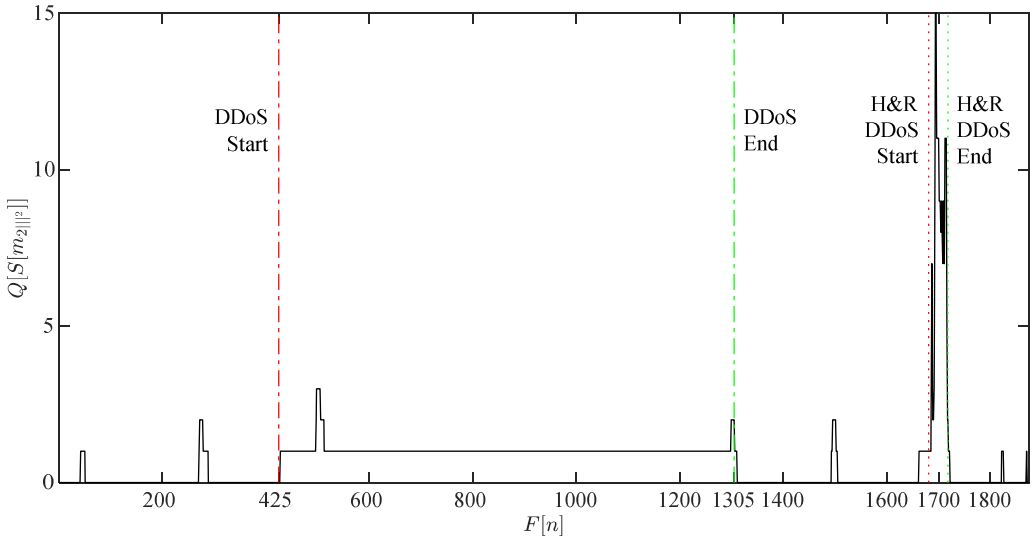


Fig. H.23. Cumulative sum *S* applied to the variance multiscalor 2nd component ($m_{2\|\|^2}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

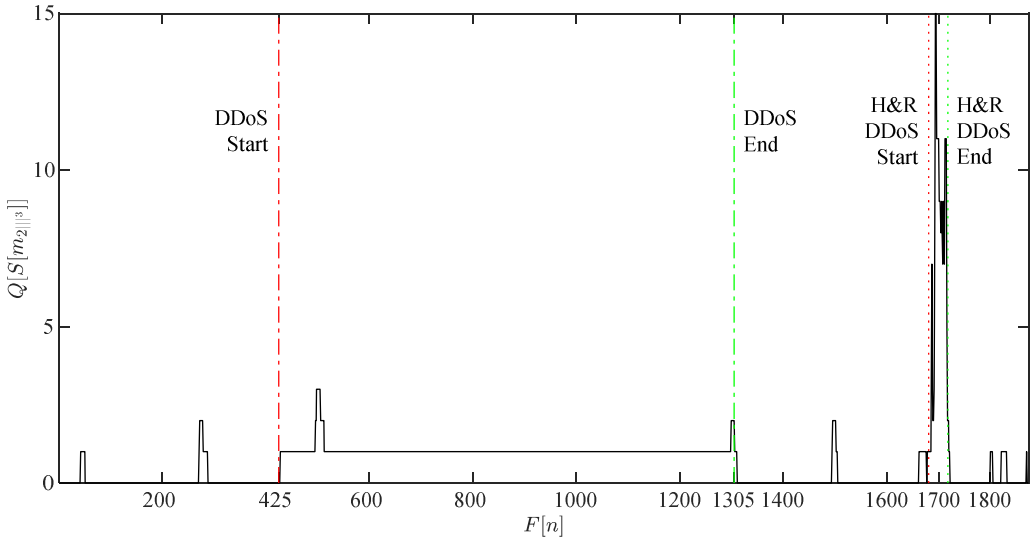


Fig. H.24. Cumulative sum *S* applied to the variance multiscalor 3rd component ($m_{2\|\|^3}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

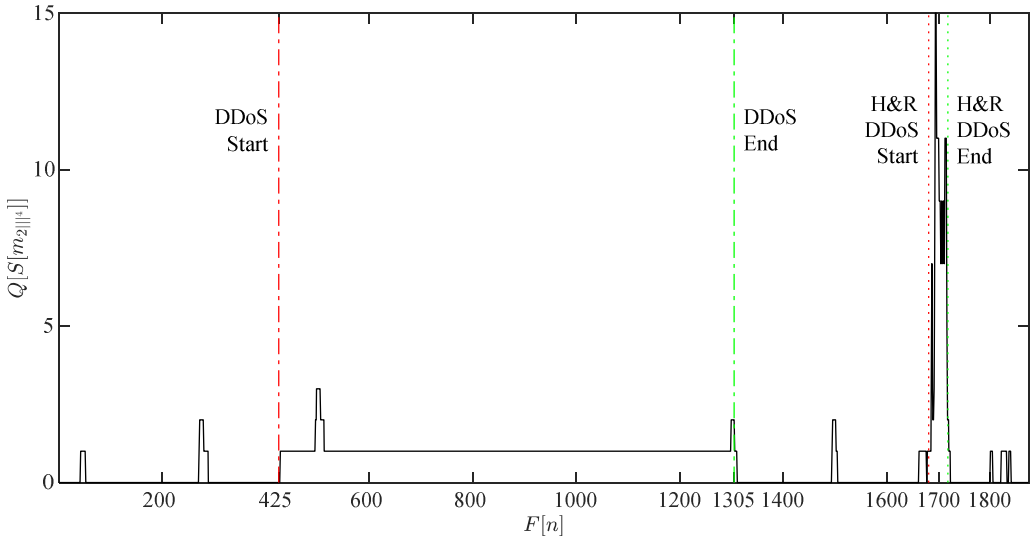


Fig. H.25. Cumulative sum *S* applied to the variance multiscalor 4th component ($m_{2\|\|^4}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

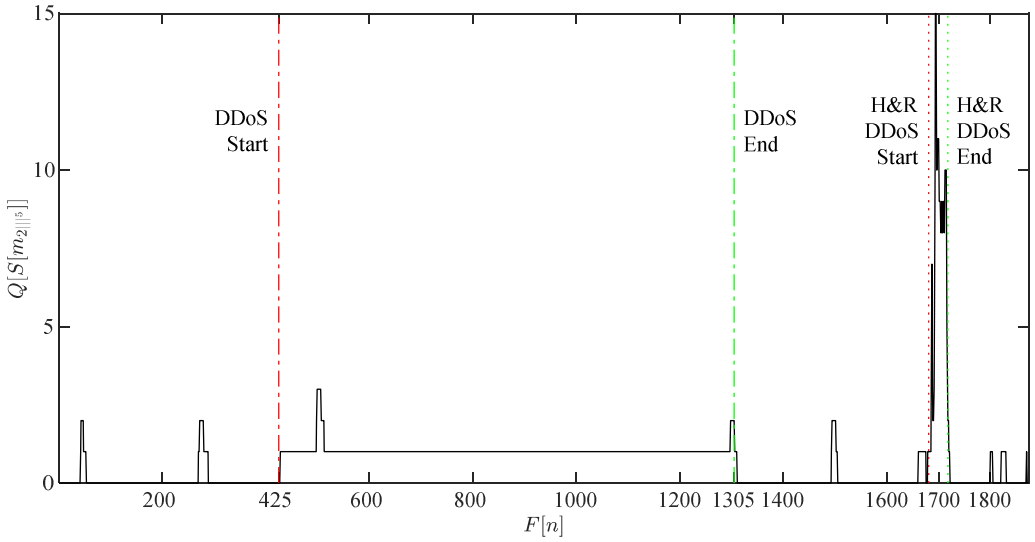


Fig. H.26. Cumulative sum *S* applied to the variance multiscalor 5th component ($m_{2||^5}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

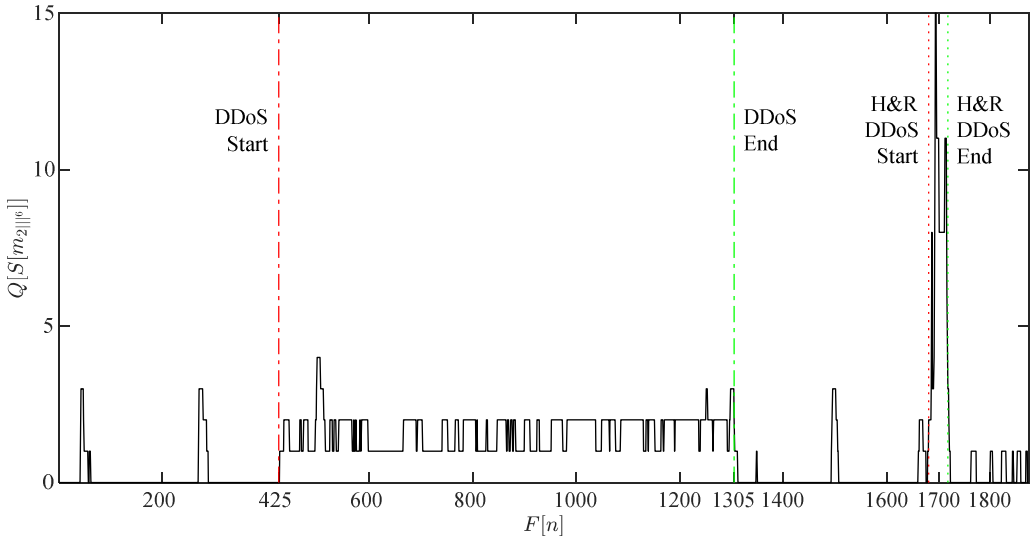


Fig. H.27. Cumulative sum *S* applied to the variance multiscalor 6th component ($m_{2||^6}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

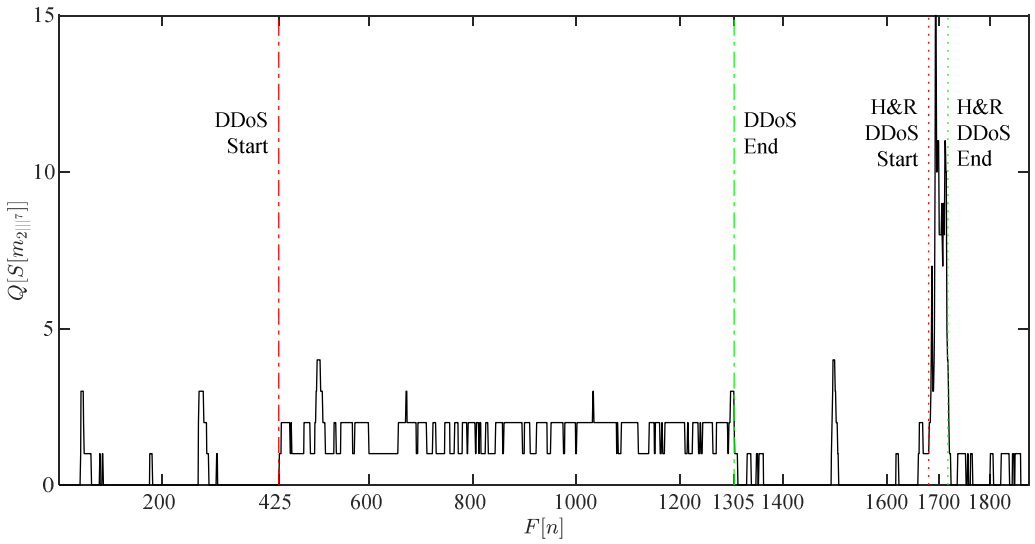


Fig. H.28. Cumulative sum *S* applied to the variance multiscalor 7th component ($m_{2|||^7}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

H.2 Zero-Crossing Rate

H.2.1 Zero-Crossing Rate Applied to Variance Multiscalor Components

The ZCR applied to each variance multiscalor component ($Z_n[m_{2|||^n}]$) provides results that are not as straightforward as the cumulative sum. However, the dynamics for both DDoS attacks are maintained as seen in Figs. H.29 to H.35 for all variance multiscalor components (from first, $m_{2|||^1}$, to seventh, $m_{2|||^7}$). The visual quality of the results appears to increase as one traverses from the first multiscalor component, $m_{2|||^1}$, to the seventh, $m_{2|||^7}$. It is interesting to observe that the H&R DDoS attack appears inverted from the first, $m_{2|||^1}$, to forth, $m_{2|||^4}$ and becomes positive for the rest of the components. Besides these occurrences, the shapes of both DDoS attacks are preserved for the rest of the results for the ZCR analysis on the variance multiscalor components.

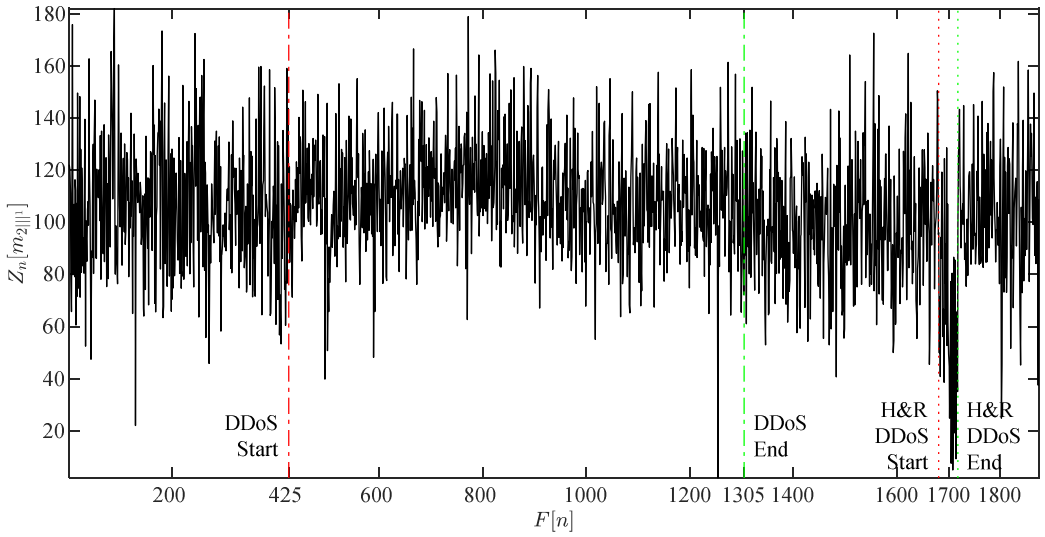


Fig. H.29. ZCR Z_n applied to the variance multiscalor 1st component ($m_{2|||^1}$). A processing *frame* of 256 samples is used.

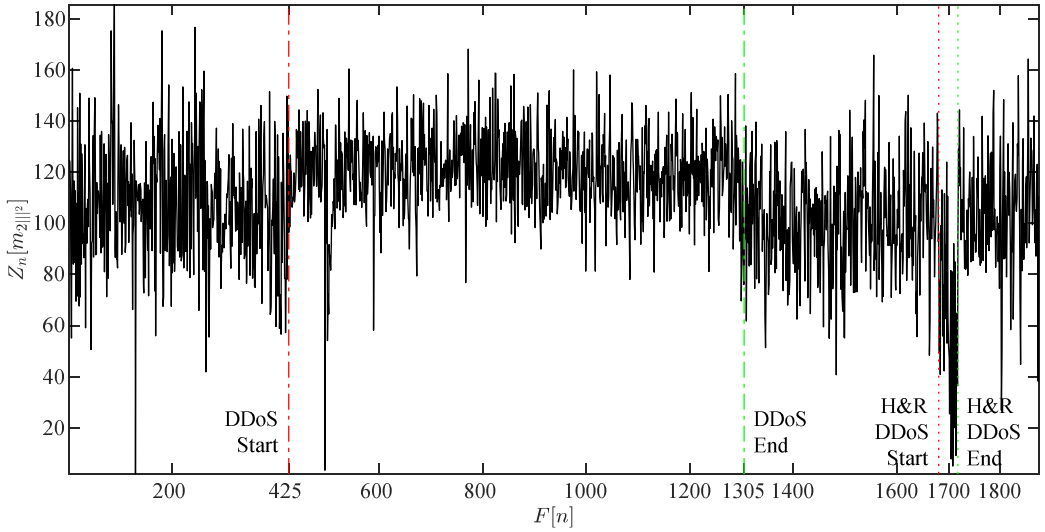


Fig. H.30. ZCR Z_n applied to the variance multiscalor 2nd component ($m_{2|||^2}$). A processing *frame* of 256 samples is used.

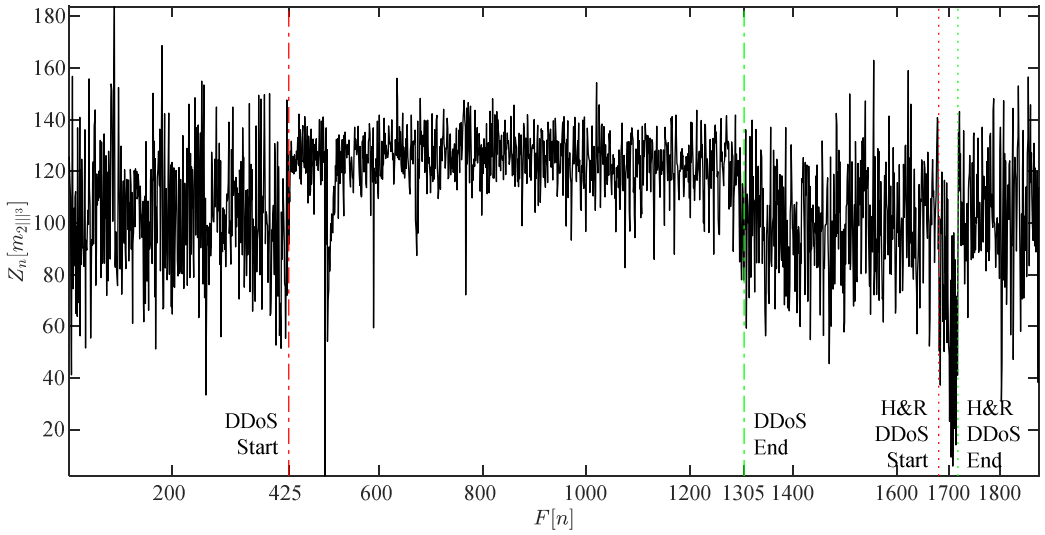


Fig. H.31. ZCR Z_n applied to the variance multiscalor 3rd component $(m_{2|||^3})$. A processing *frame* of 256 samples is used.

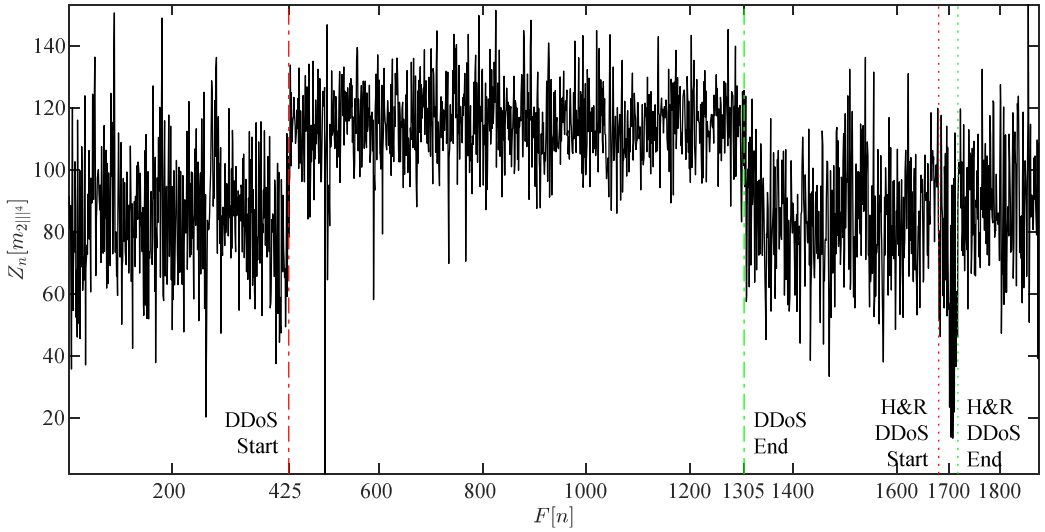


Fig. H.32. ZCR Z_n applied to the variance multiscalor 4th component $(m_{2|||^4})$. A processing *frame* of 256 samples is used.

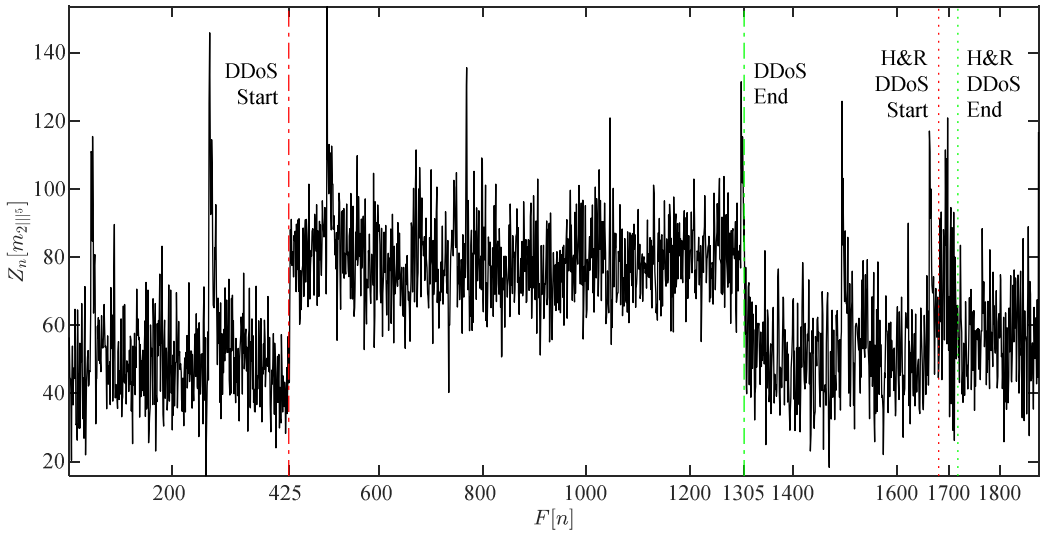


Fig. H.33. ZCR Z_n applied to the variance multiscalor 5th component $(m_{2\|\|^5})$. A processing *frame* of 256 samples is used.

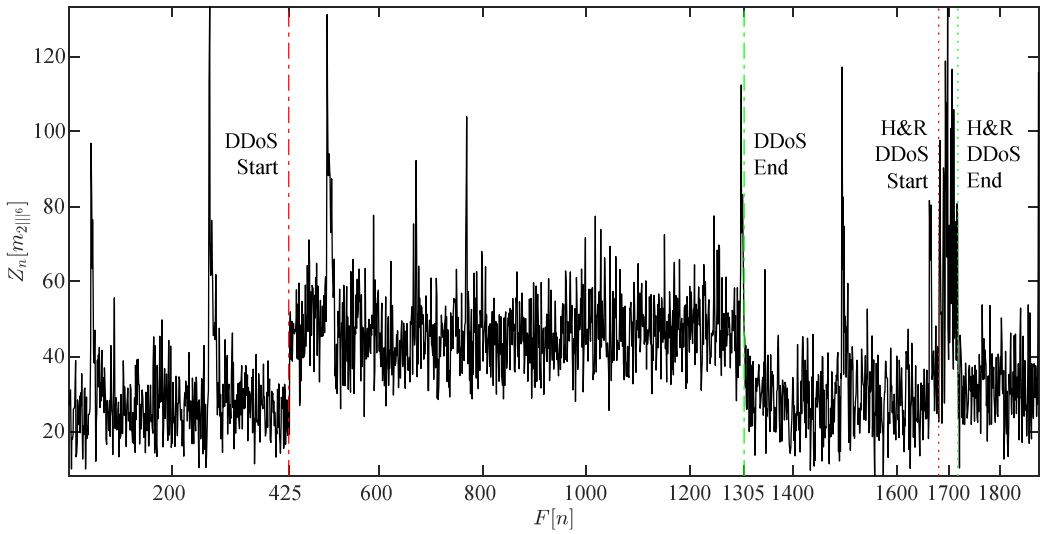


Fig. H.34. ZCR Z_n applied to the variance multiscalor 6th component $(m_{2\|\|^6})$. A processing *frame* of 256 samples is used.

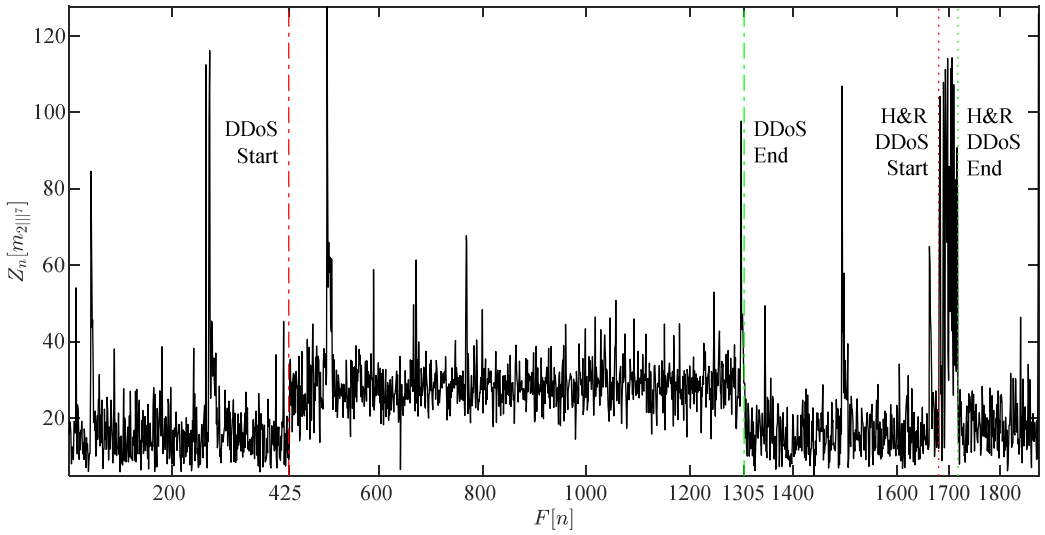


Fig. H.35. ZCR Z_n applied to the variance multiscalor 7th component $(m_{2|||^7})$. A processing *frame* of 256 samples is used.

H.2.2 Zero-Crossing Rate Applied to Variance Multiscalor Components After Donoho's Denoising

Figures H.36 to H.42 depict Donoho's denoising applied to the outcome of ZCR after processing the seven variance multiscalor components. It is seen that noise has been reduced from the first, $m_{2|||^1}$, to seventh, $m_{2|||^7}$, components. Both DDoS attacks, DNS amplification and H&R, are seen more nitidly from the second, $m_{2|||^2}$, to the seventh, $m_{2|||^7}$, variance multiscalor components. The visual quality in the results presented in Figs. M.36 to M.42 appears to improve from the first multiscalor component, $m_{2|||^1}$, to the seventh, $m_{2|||^7}$. Excluding the H&R DDoS attack shape from the first, $m_{2|||^1}$, to forth, $m_{2|||^4}$, components, the shape of both DDoS attacks has been preserved.

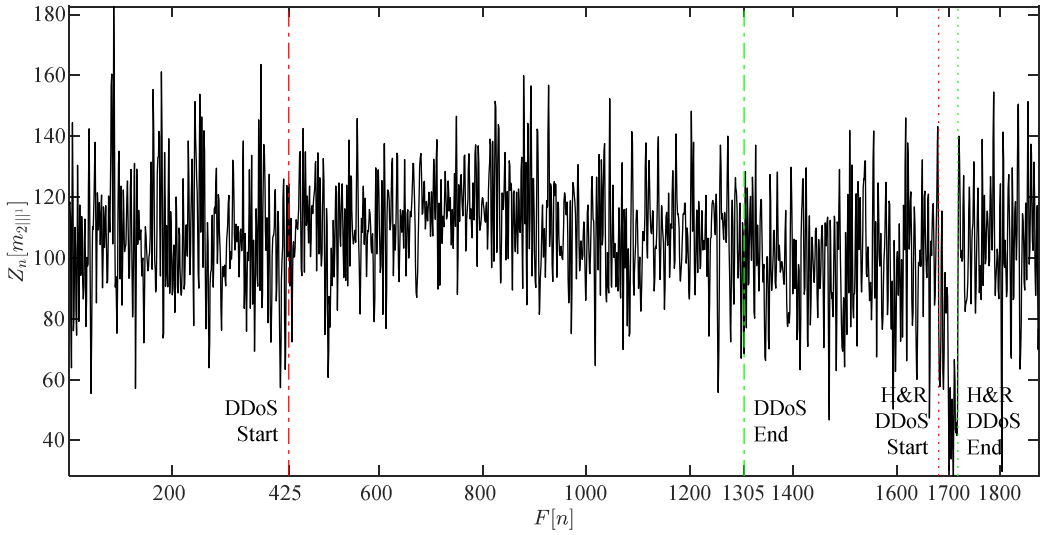


Fig. H.36. ZCR Z_n applied to the variance multiscalor 1st component $(m_{2\|^1})$ after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

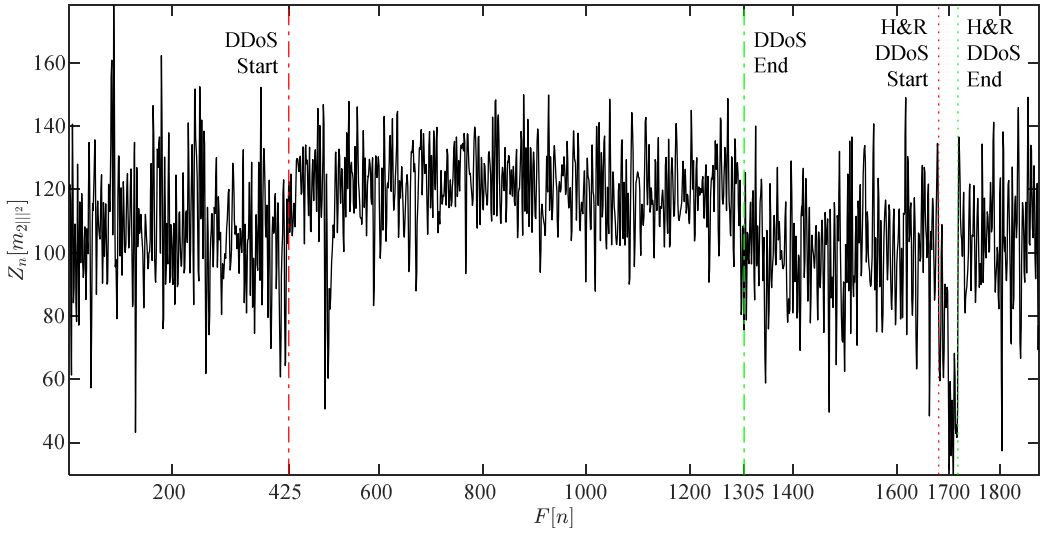


Fig. H.37. ZCR Z_n applied to the variance multiscalor 2nd component $(m_{2\|^2})$ after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

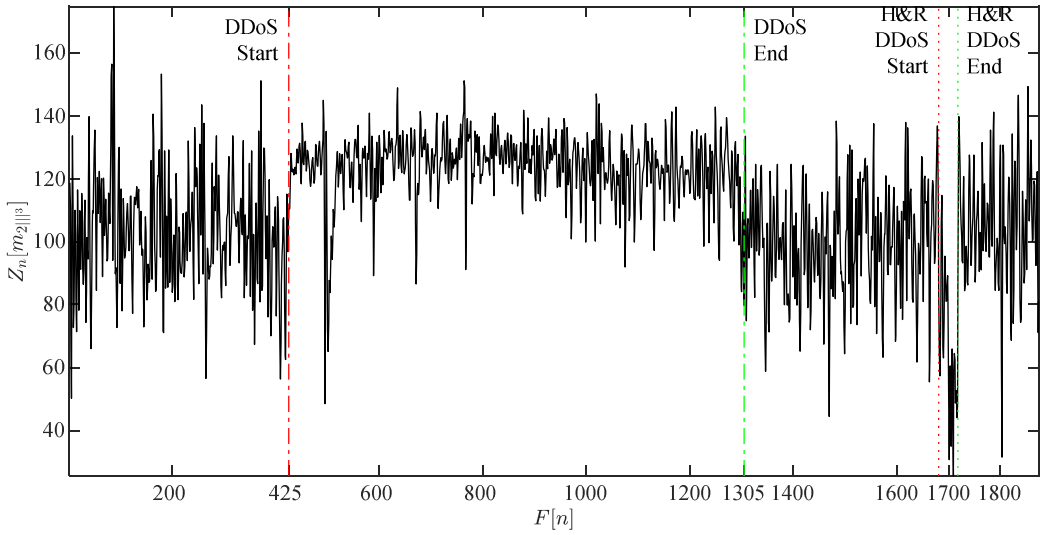


Fig. H.38. ZCR Z_n applied to the variance multiscalor 3rd component ($m_{2\|\|^3}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

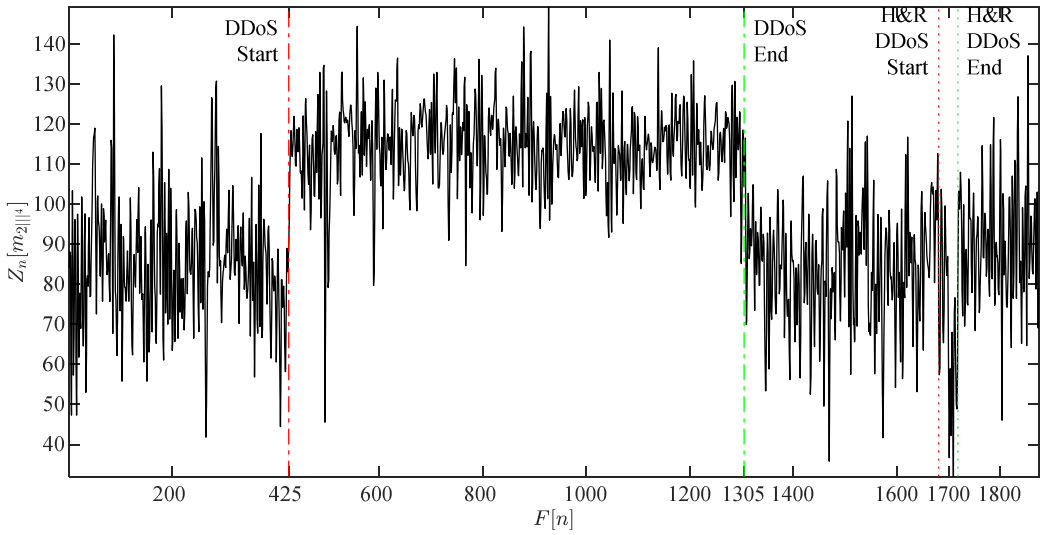


Fig. H.39. ZCR Z_n applied to the variance multiscalor 4th component ($m_{2\|\|^4}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

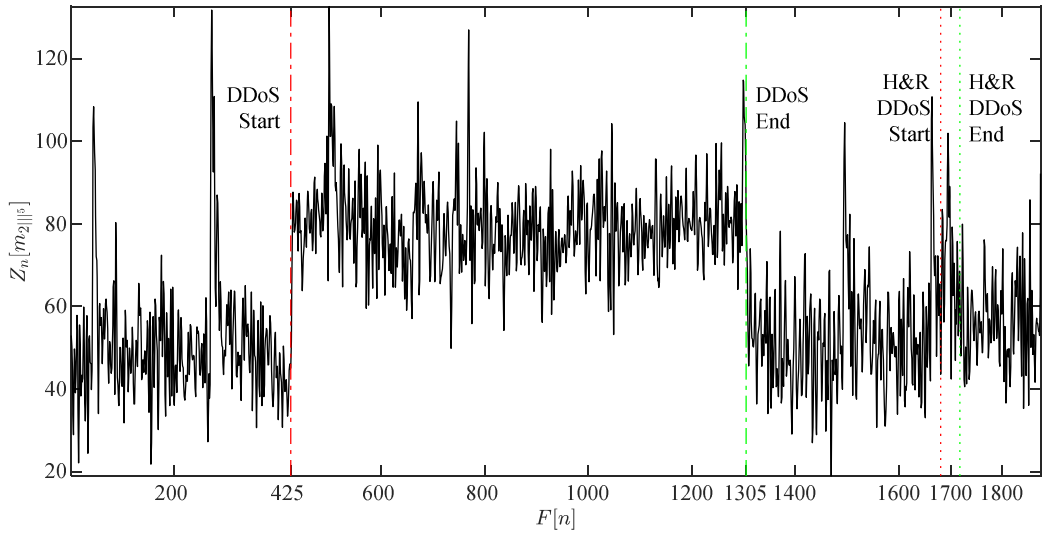


Fig. H.40. ZCR Z_n applied to the variance multiscalor 5th component ($m_{2\|\|^5}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

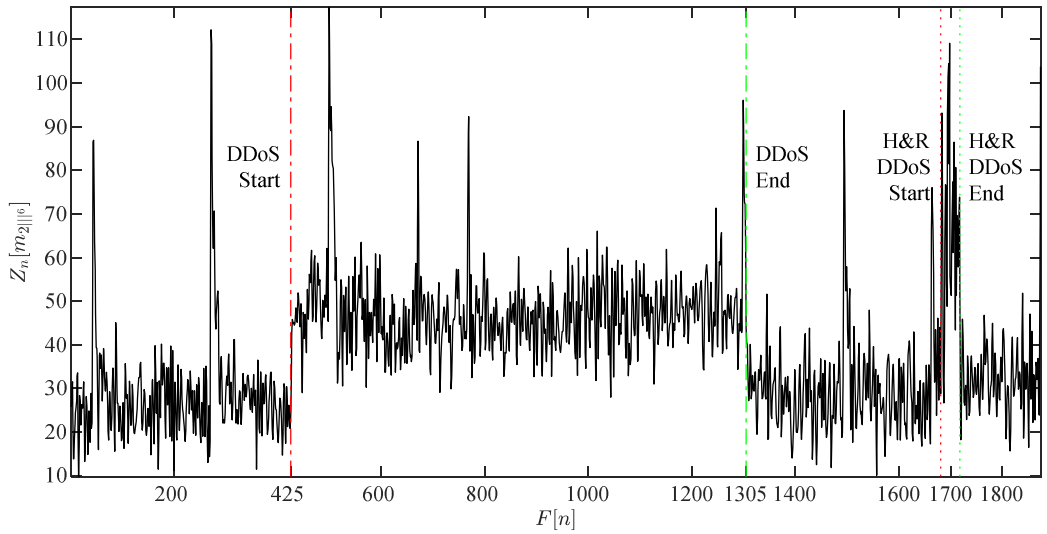


Fig. H.41. ZCR Z_n applied to the variance multiscalor 6th component ($m_{2\|\|^6}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

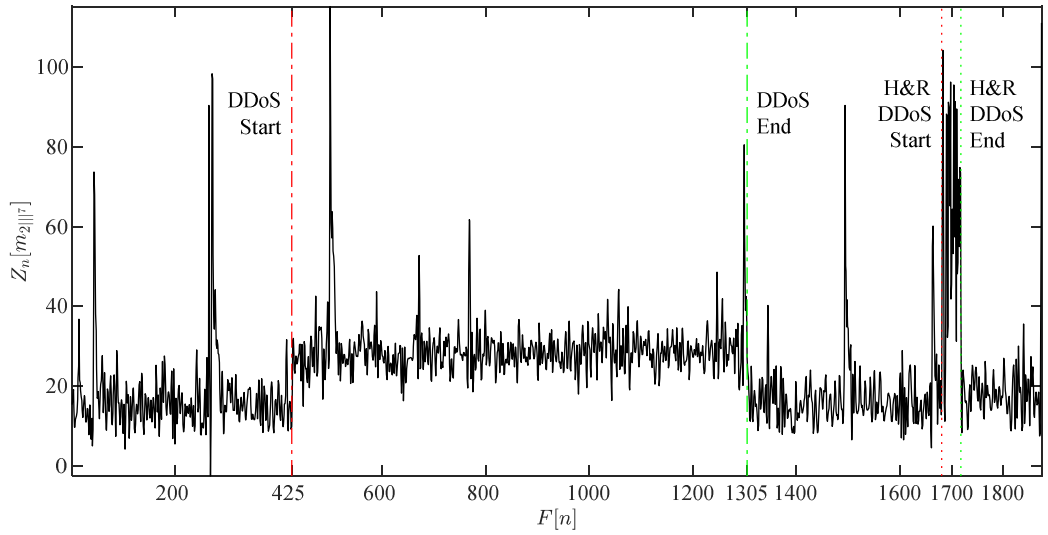


Fig. H.42. ZCR Z_n applied to the variance multiscalor 7th component ($m_{2|||^7}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

H.2.3 Zero-Crossing Rate Applied to Variance Multiscalor Components Non-Linearly Filtered After Donoho's Denoising

The subsequent nonlinear filtering stage is applied to the variance multiscalor components after Donoho's denoising. Figures H.43 to H.49 show the outcome of this processing stage for the first, $m_{2|||^1}$, to seventh, $m_{2|||^7}$, components respectively. A significant reduction in noise is seen for the seven components and all waveforms appear smooth. Both DDoS attacks, DNS amplification and H&R, are seen clearly from the second, $m_{2|||^2}$, to the seventh, $m_{2|||^7}$, components of the variance multiscalor. The inversion of the H&R DDoS attack present from the first, $m_{2|||^1}$, to the fourth, $m_{2|||^4}$, is now more noticeable. The visual quality increment in the outcomes introduced in Figs. H.43 to H.49 from the first variance multiscalor component, $m_{2|||^1}$, to the seventh, $m_{2|||^7}$, is also more noticeable. The characteristic shape of both attacks, DNS amplification and H&R, has been maintained with the exclusion of the H&R DDoS attack shape being inverted from the first, $m_{2|||^1}$, to forth, $m_{2|||^4}$, components.

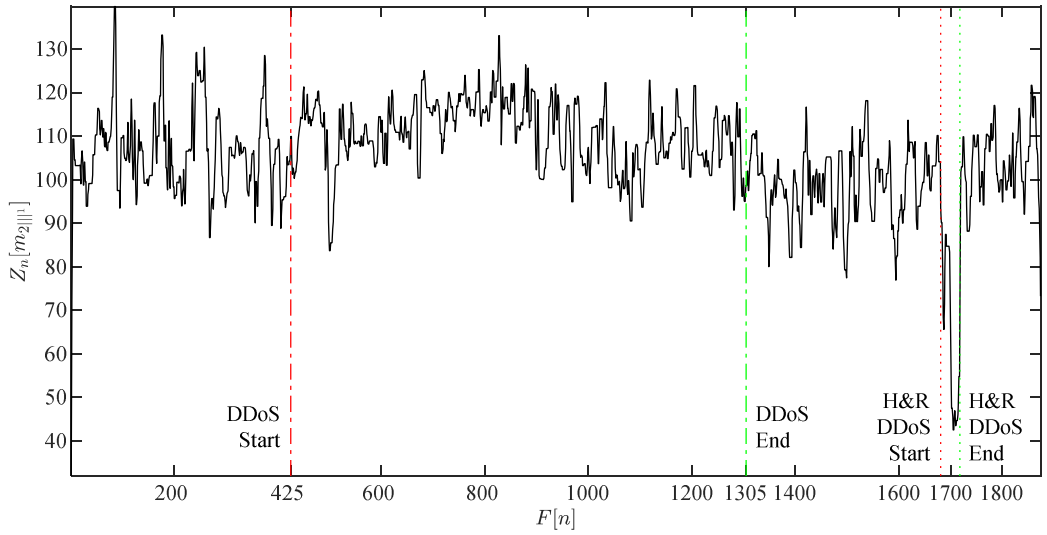


Fig. H.43. ZCR Z_n applied to the variance multiscalor 1st component $(m_{2|||^1})$ median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

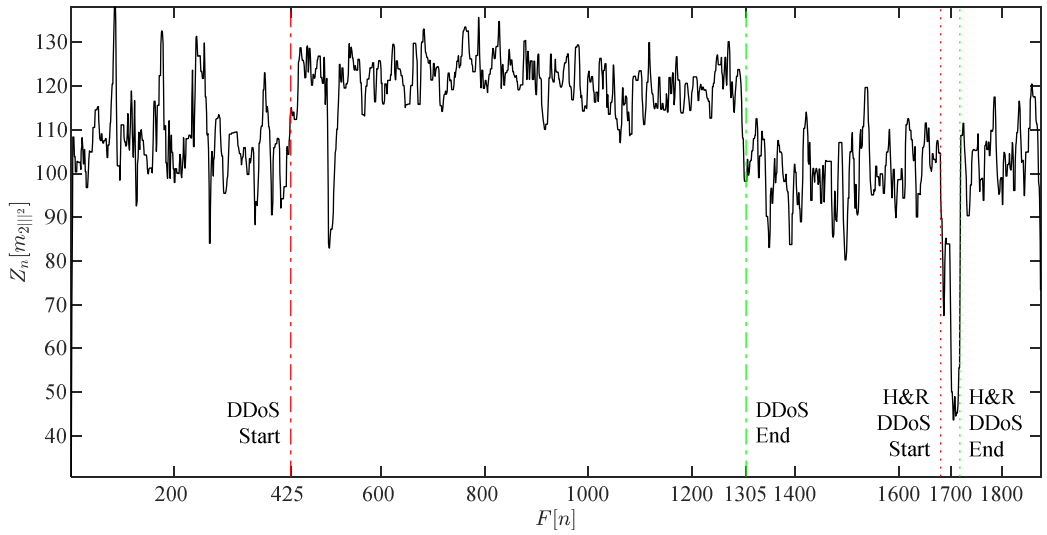


Fig. H.44. ZCR Z_n applied to the variance multiscalor 2nd component $(m_{2|||^2})$ median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

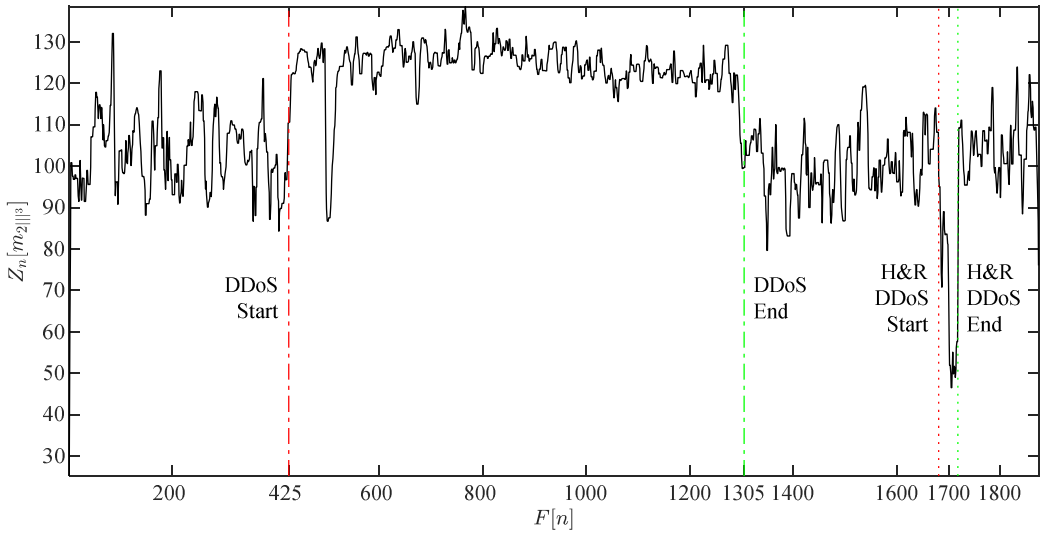


Fig. H.45. ZCR Z_n applied to the variance multiscalor 3rd component $(m_{2|||^3})$ median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

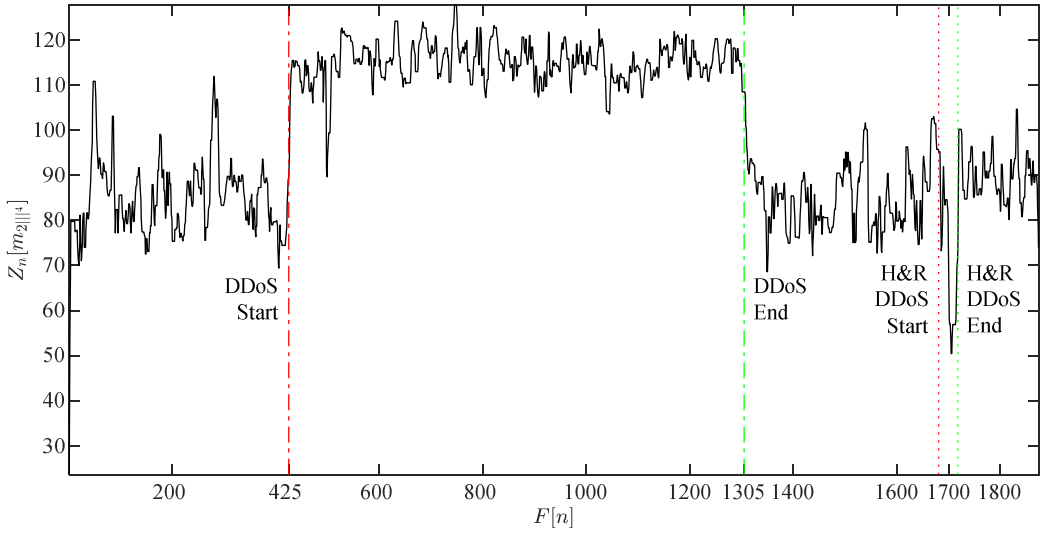


Fig. H.46. ZCR Z_n applied to the variance multiscalor 4th component $(m_{2|||^4})$ median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

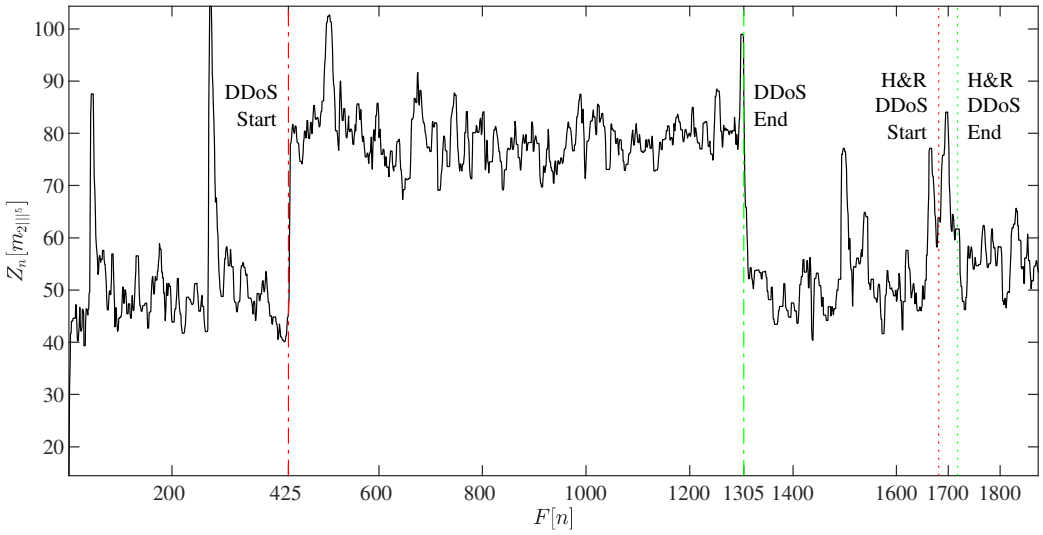


Fig. H.47. ZCR Z_n applied to the variance multiscalor 5th component $(m_{2||^5})$ median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

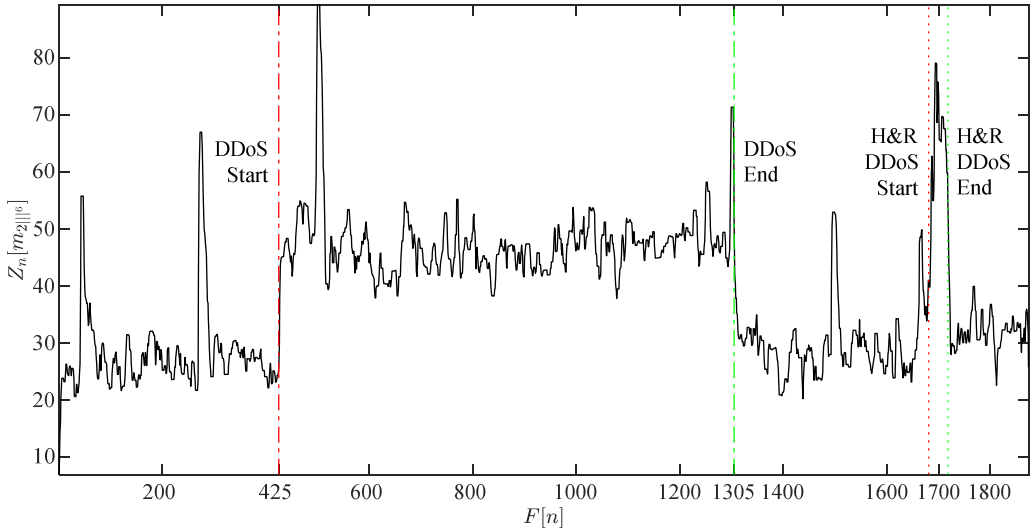


Fig. H.48. ZCR Z_n applied to the variance multiscalor 6th component $(m_{2||^6})$ median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

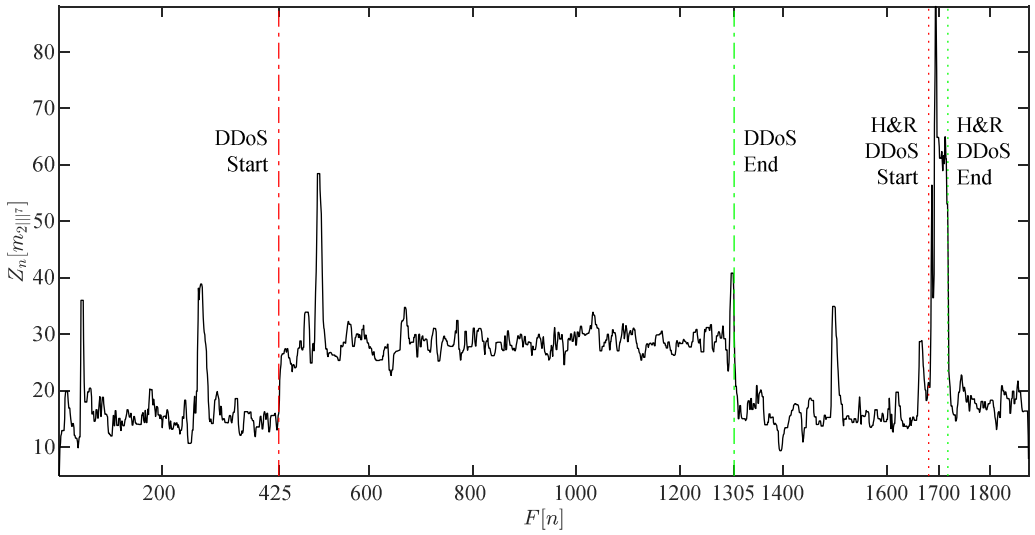


Fig. H.49. ZCR Z_n applied to the variance multiscalor 7th component ($m_{2|||^7}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

H.2.4 Zero-Crossing Rate Applied to Variance Multiscalor Components Quantization of Non-Linear Filtering After Donoho's Denoising

Analogously to the cumulative sum case, the extracted features through ZCR are readied for further machine learning processing through ART1, the quantization phase required for converting the component waveforms analysed through ZCR into defined amplitude level codes. The quantization through the Lloyd's methodology is shown in Figs. H.50 to H.56 for the first, $m_{2|||^1}$, to seventh, $m_{2|||^7}$, components respectively. The visual quality improvement increases from the first, $m_{2|||^1}$, to seventh, $m_{2|||^7}$, components is noticeable, but not of the same visual quality as the one obtained with the cumulative sum. Both DDoS attacks shapes, DNS amplification and H&R, are noticeable from the first, $m_{2|||^1}$, to the seventh, $m_{2|||^7}$, variance multiscalor components. The DNS amplification DDoS attack exhibits a peculiar sinusoidal like occurrence (shown in Fig. H.50) in the first, $m_{2|||^1}$, variance multiscalor component. The inverse peaks corresponding to the H&R DDoS attack present from the first, $m_{2|||^1}$, to the fourth, $m_{2|||^4}$, variance multiscalor components are clearly visible after the progression of data processing starting with ZCR and concluding with its quantization. Besides the four inverted peaks, the rest of the DDoS attacks instances are preserved and resemble the results obtained with the cumulative sum, but ZCR does not match them in quality. It is important to notice that the DNS amplification DDoS attack exhibits a discontinuity shortly at the beginning from the first, $m_{2|||^1}$, to the third, $m_{2|||^3}$, variance multiscalor components.

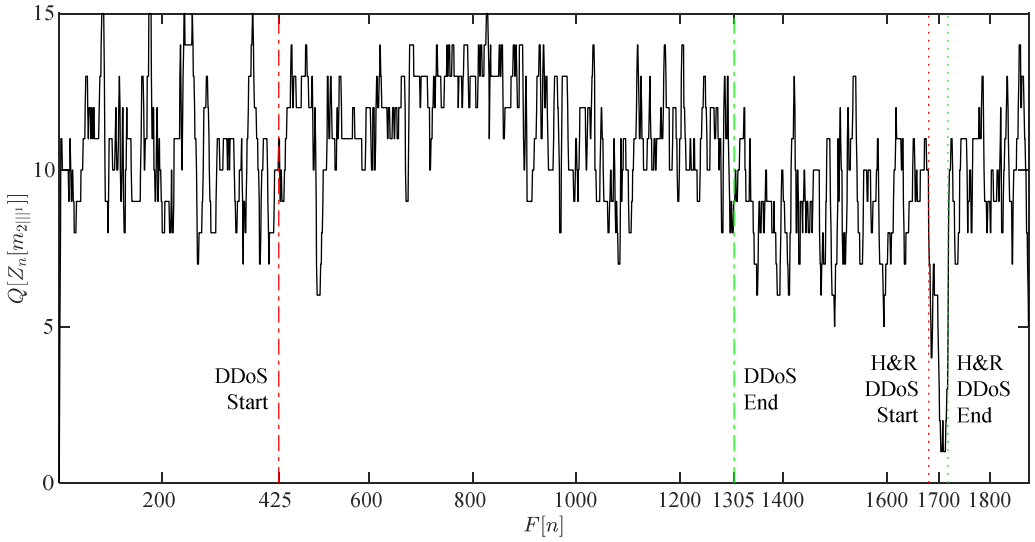


Fig. H.50. ZCR Z_n applied to the variance multiscalor 1st component $(m_{2|||^1})$ quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

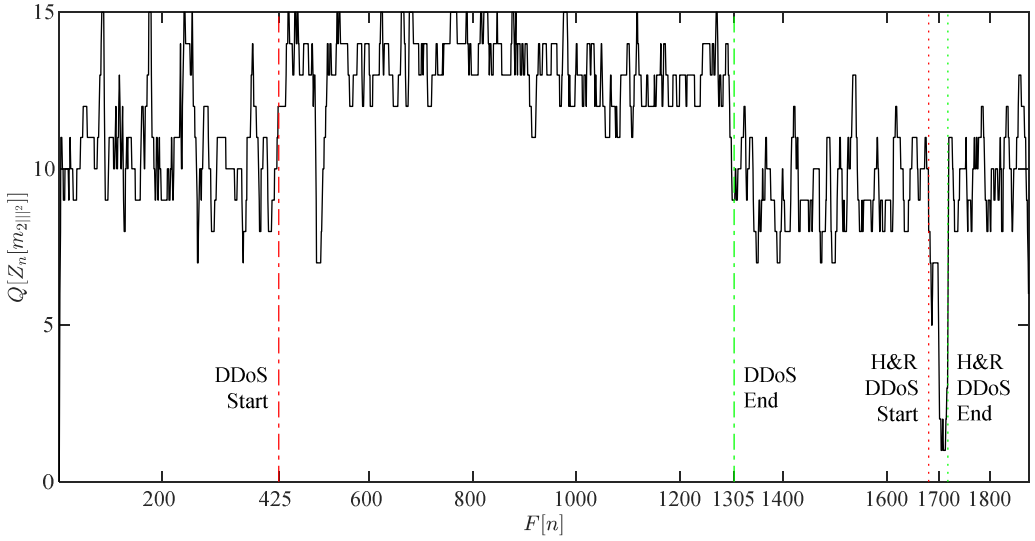


Fig. H.51. ZCR Z_n applied to the variance multiscalor 2nd component $(m_{2|||^2})$ quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

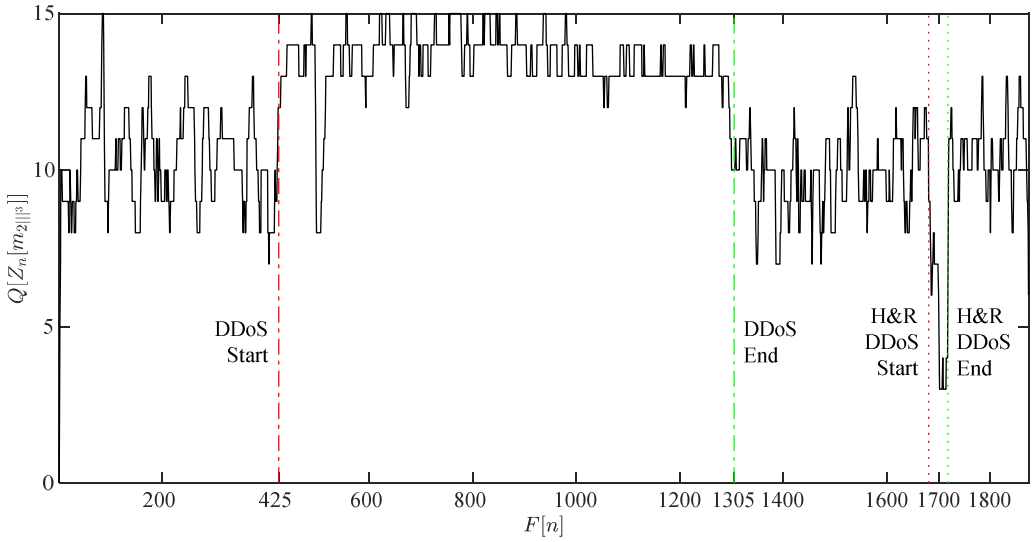


Fig. H.52. ZCR Z_n applied to the variance multiscalor 3rd component ($m_{2|||^3}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

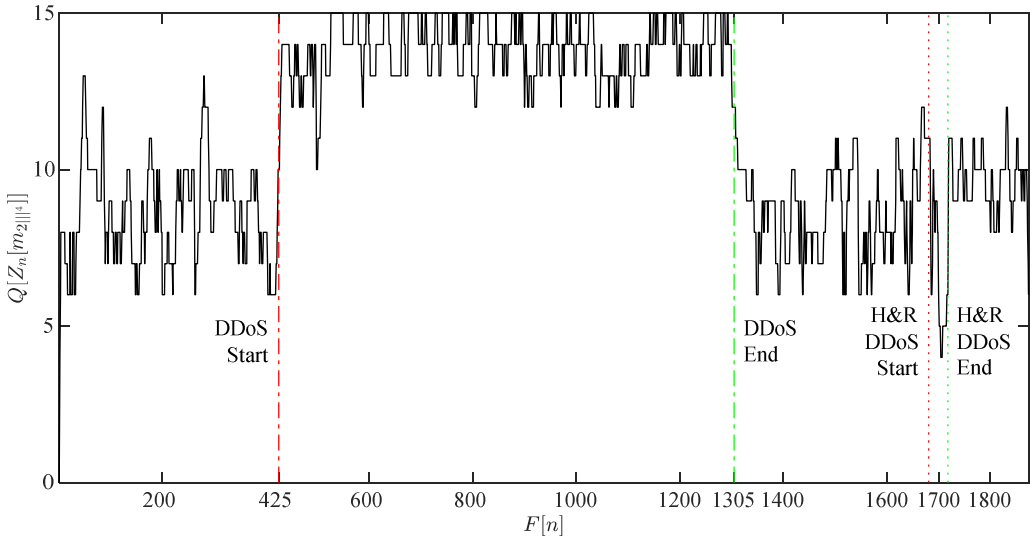


Fig. H.53. ZCR Z_n applied to the variance multiscalor 4th component ($m_{2|||^4}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

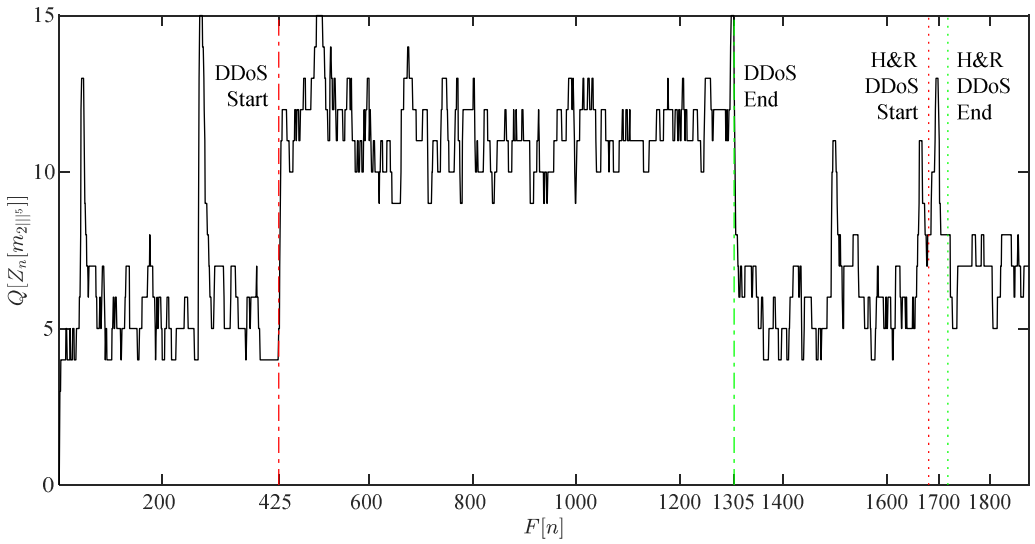


Fig. H.54. ZCR Z_n applied to the variance multiscalor 5th component ($m_{2\|\|^5}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

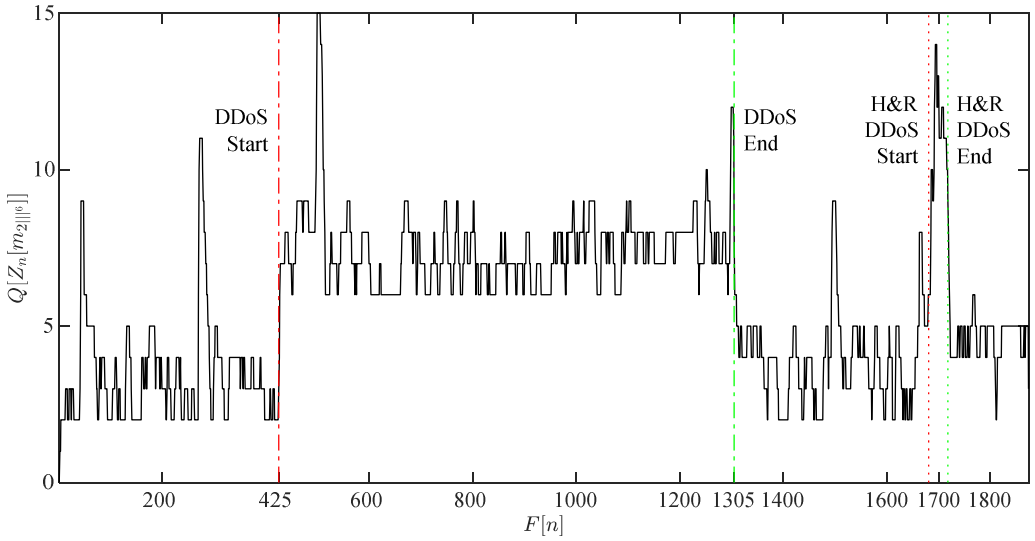


Fig. H.55. ZCR Z_n applied to the variance multiscalor 6th component ($m_{2\|\|^6}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

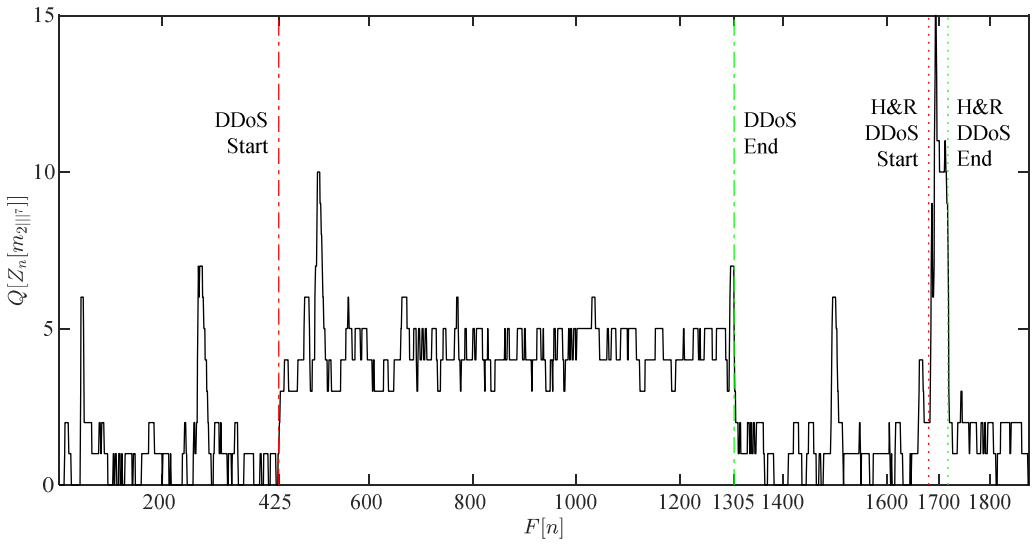


Fig. H.56. ZCR Z_n applied to the variance multiscalor 7th component ($m_{2|||^7}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

H.3 Shannon's Entropy

H.3.1 Shannon's Entropy Applied to Variance Multiscalor Components

Applying Shannon's entropy to each variance multiscalor component ($H[m_{2|||^n}]$) does not yield the best outcomes when compared with the previous methodologies, cumulative sum and ZCR. One might attribute this to the small number of samples in the processing frame, 256. However, experiments with bigger processing frames, 1,024 and 4,096, were carried and the quality of the features based on Shannon's entropy did not increase significantly. Continuing with the secondary processing frame having a size of 256 in order to represent the worst case data processing scenario and at the same time achieving the fast anomaly detection case, one can see in Figs. H.57 to H.63 representing the forth, $m_{2|||^4}$, to seventh, $m_{2|||^7}$, variance multiscalor components that for the DDoS attacks, the DNS amplification based attack has neither clear beginning nor end whereas the H&R attack seems to be represented by an inverse peak in Figs. M.57 to M.60 similar to previous cases within the ZCR. The DC value, in this case Shannon's entropy, of the waveforms increases from the fourth, $m_{2|||^4}$, to seventh, $m_{2|||^7}$, components.

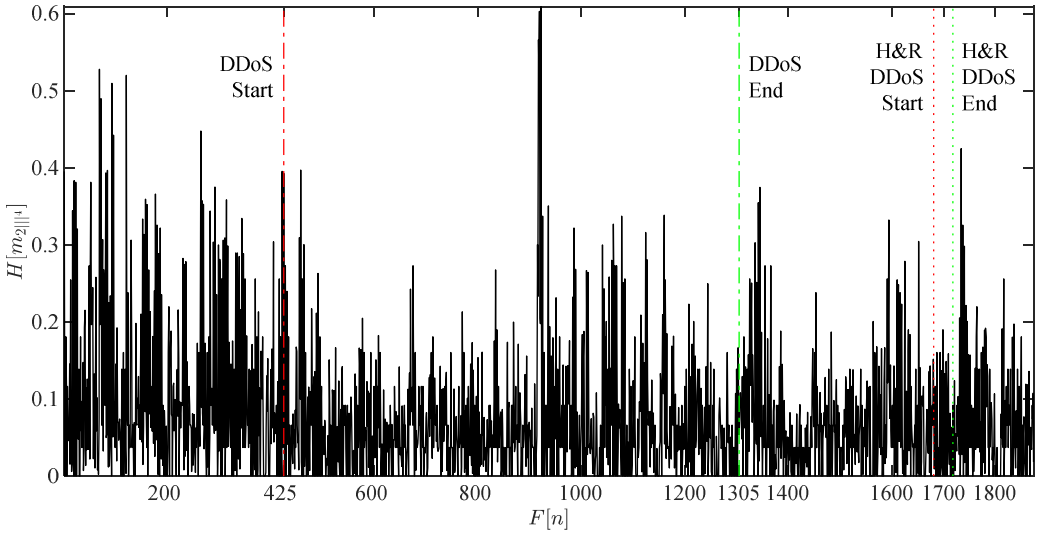


Fig. H.57. Shannon's entropy *H* applied to the variance multiscalor 4th component ($m_{2\|\|^4}$). A processing *frame* of 256 samples is used.

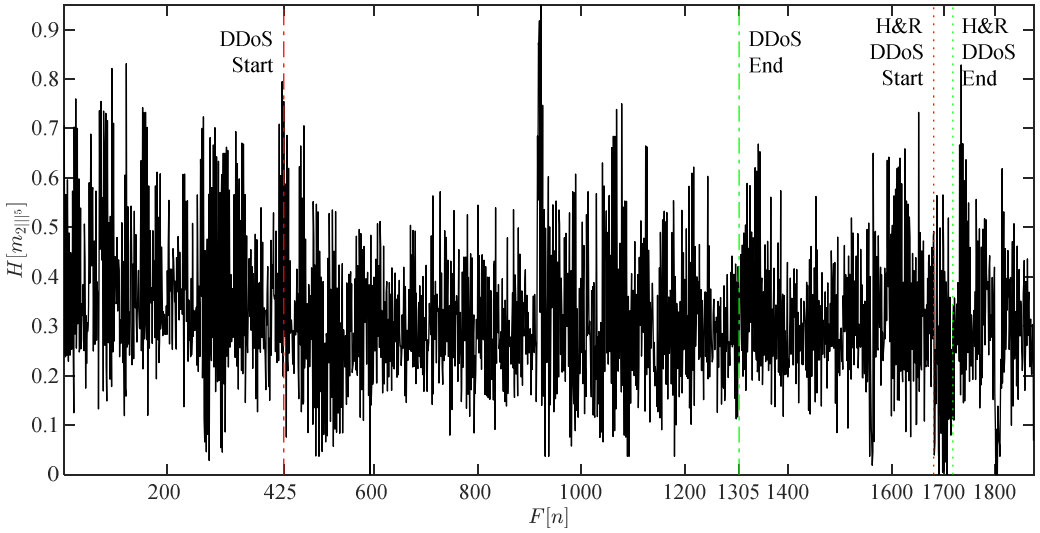


Fig. H.58. Shannon's entropy *H* applied to the variance multiscalor 5th component ($m_{2\|\|^5}$). A processing *frame* of 256 samples is used.

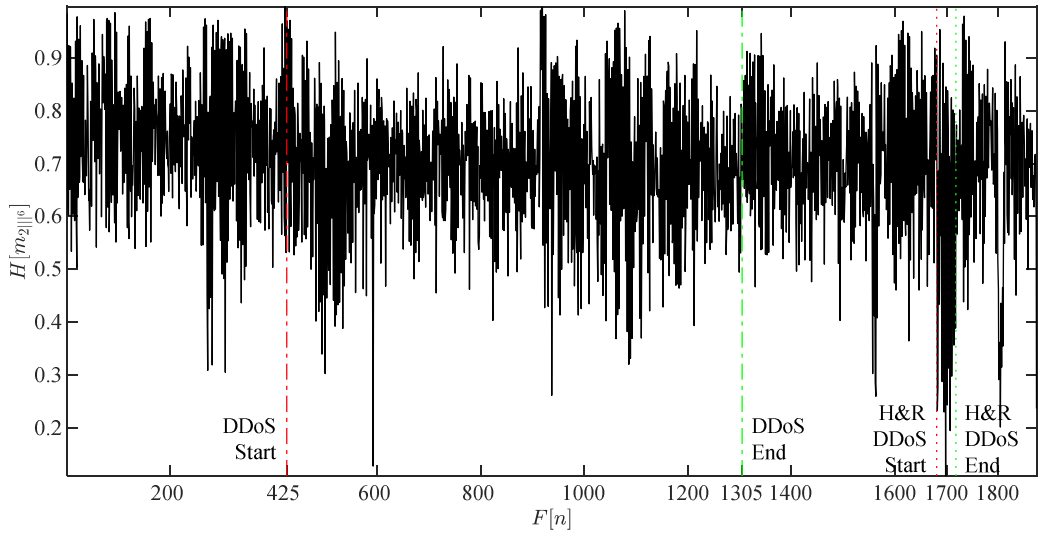


Fig. H.59. Shannon's entropy *H* applied to the variance multiscalor 6th component $(m_{2|||^6})$. A processing *frame* of 256 samples is used.

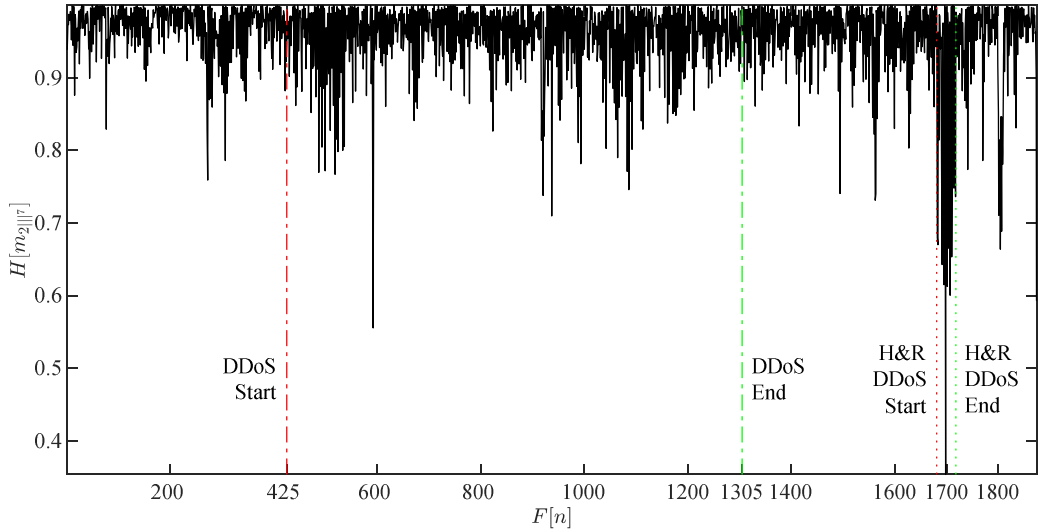


Fig. H.60. Shannon's entropy *H* applied to the variance multiscalor 7th component $(m_{2|||^7})$. A processing *frame* of 256 samples is used.

H.3.2 Shannon's Entropy Applied to Variance Multiscalor Components After Donoho's Denoising

Figures H.61 to H.64 depict Donoho's denoising applied to the outcome of Shannon's entropy after processing variance multiscalor components. It is seen that noise has been reduced

from the fourth, $m_{2|||^4}$, to seventh, $m_{2|||^7}$, components. From both DDoS attacks, the DNS amplification and H&R, seem to start emerging, with faint beginning and end and as a deeper peak respectively, in Figs. H.61 to H.64. The visual quality in the results presented in Figs. M.62 to M.64 appears to be more significant and increasing for the H&R attack.

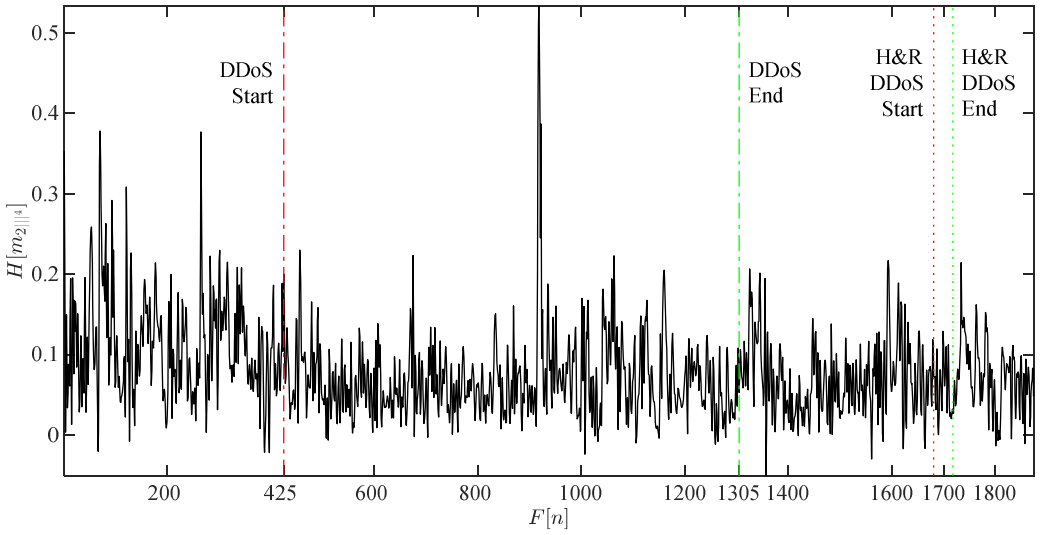


Fig. H.61. Shannon's entropy *H* applied to the variance multiscalor 4th component ($m_{2|||^4}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

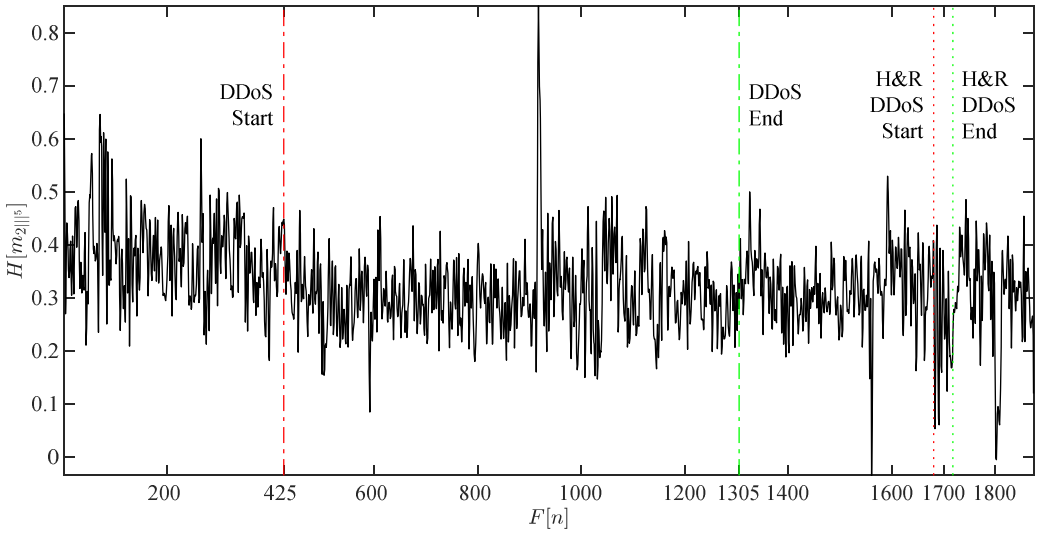


Fig. H.62. Shannon's entropy *H* applied to the variance multiscalor 5th component ($m_{2|||^5}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

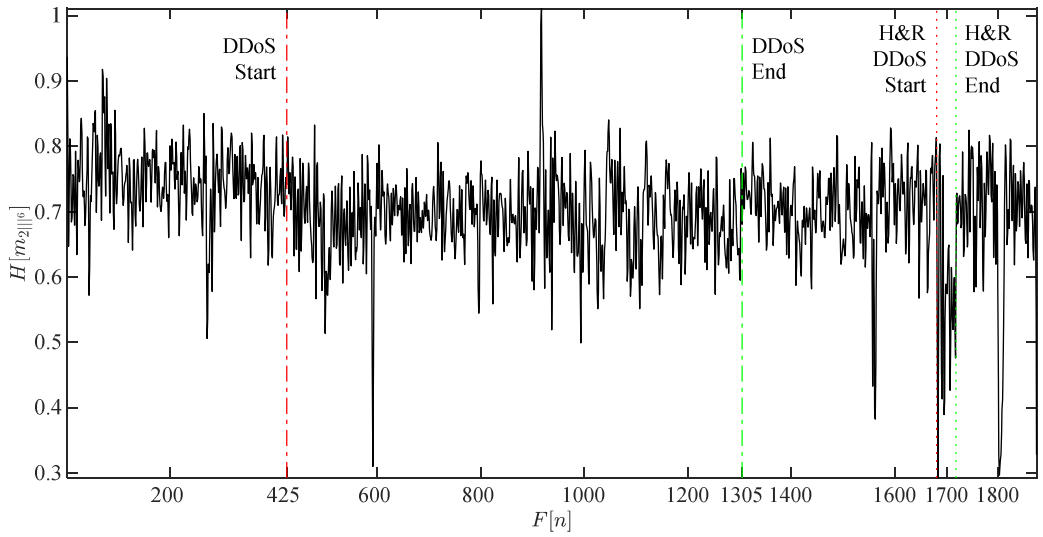


Fig. H.63. Shannon's entropy *H* applied to the variance multiscalor 6th component ($m_{2\|\|^6}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

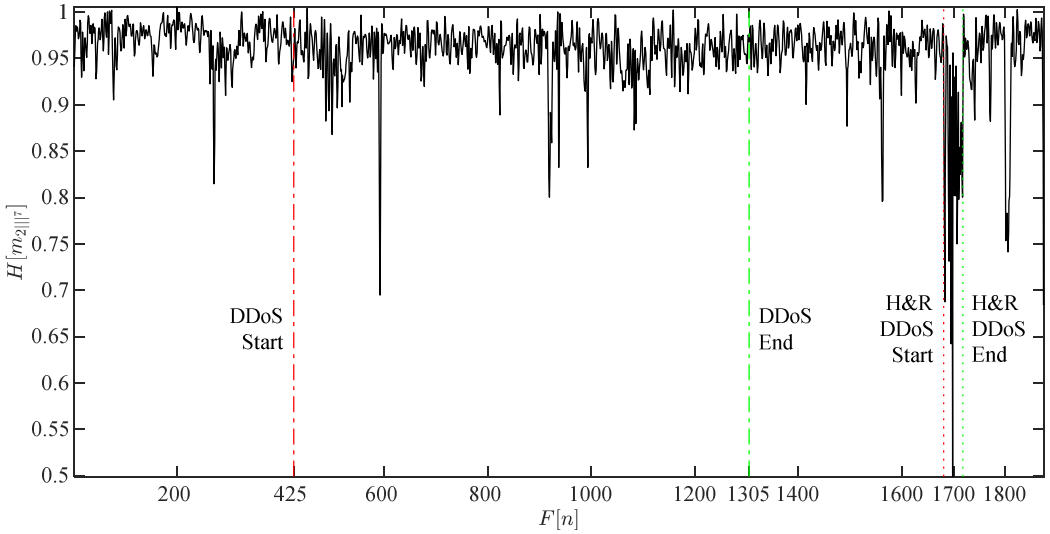


Fig. H.64. Shannon's entropy *H* applied to the variance multiscalor 7th component ($m_{2\|\|^7}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

H.3.3 Shannon's Entropy Applied to Variance Multiscalor Components Non-Linearly Filtered After Donoho's Denoising

The succeeding nonlinear filtering stage seen in Figs. H.65 to H.68 show the outcome of this processing stage for the fourth, $m_{2|||^4}$, to the seventh, $m_{2|||^7}$, components respectively. A further reduction in noise is seen for the cases shown, but the waveforms still appear spiky. The DNS amplification DDoS attack seems to have more defined beginning and end in Figs. M.66 and M.67, but not as sharp as in the previous methodologies used, cumulative sum and ZCR. The H&R DDoS attack is depicted as an inverted peak in Figs. M.66 to M.68. The visual quality in the results presented favour the detection of the H&R attack.

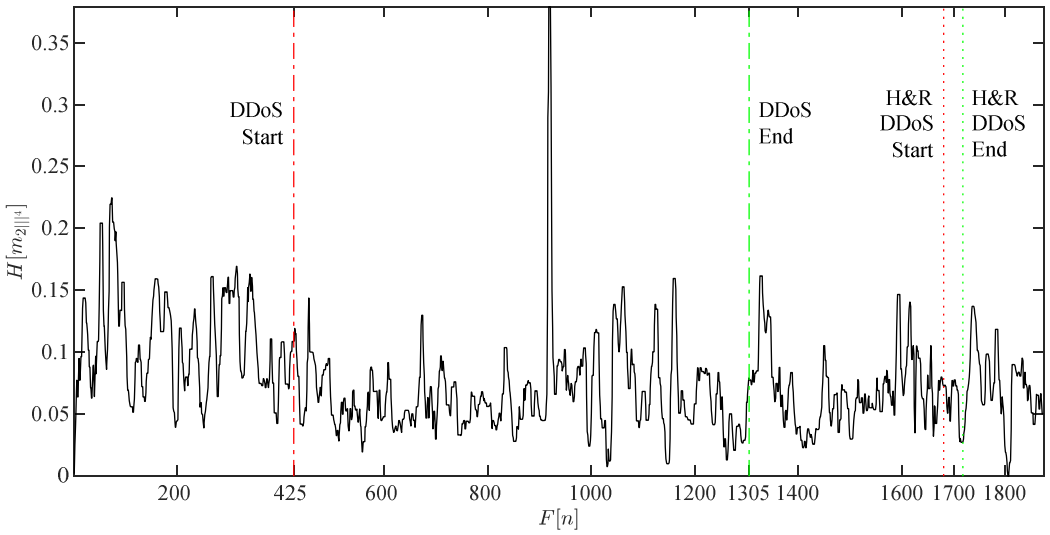


Fig. H.65. Shannon's entropy *H* applied to the variance multiscalor 4th component ($m_{2|||^4}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

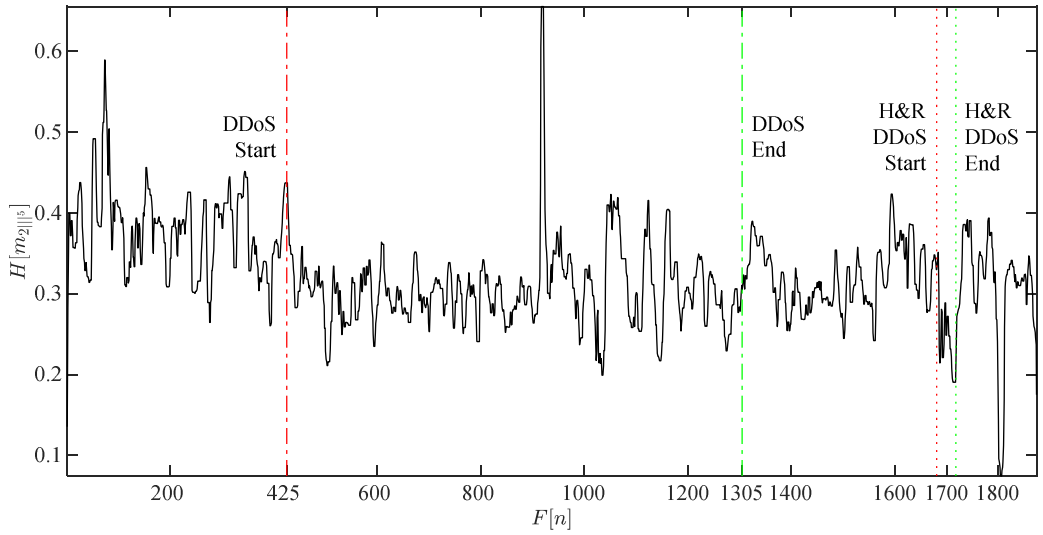


Fig. H.66. Shannon's entropy *H* applied to the variance multiscalor 5th component ($m_{2||^5}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

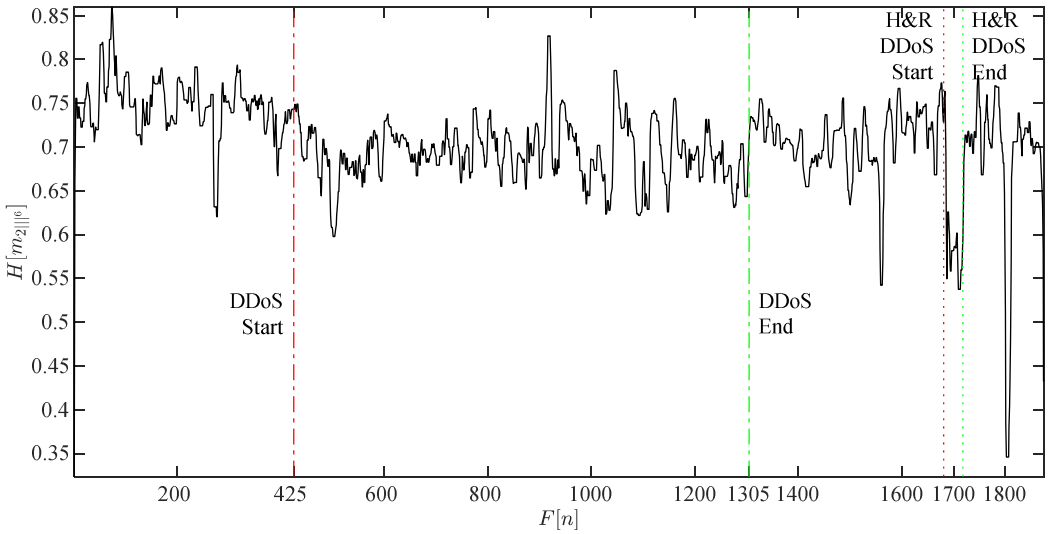


Fig. H.67. Shannon's entropy *H* applied to the variance multiscalor 6th component ($m_{2||^6}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

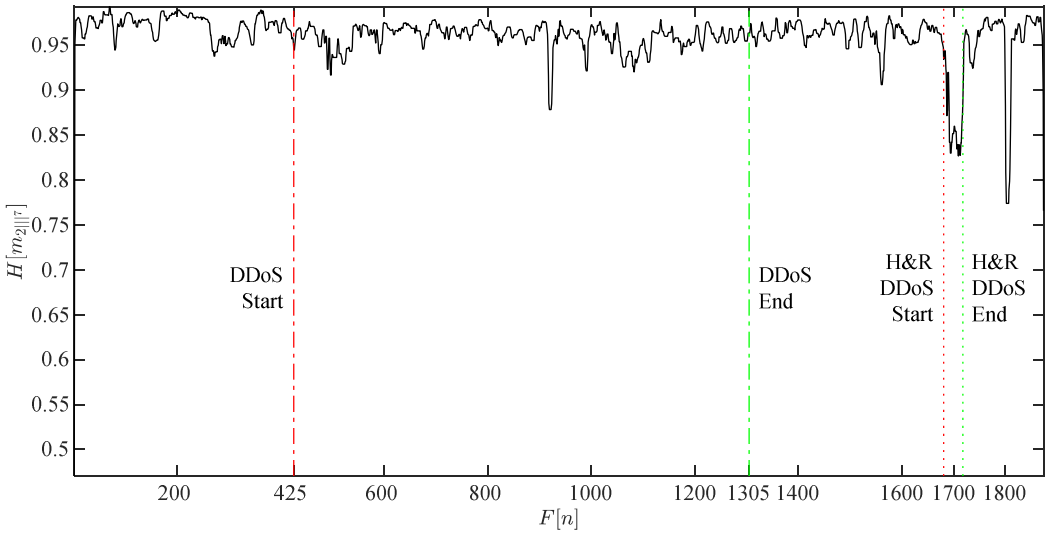


Fig. H.68. Shannon's entropy *H* applied to the variance multiscalor 7th component ($m_{2|||^7}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

H.3.4 Shannon's Entropy Applied to Variance Multiscalor Components Quantization of Non-Linear Filtering After Donoho's Denoising

Culminating the preparation of the features extracted through Shannon's entropy, a Lloyd's quantization phase is utilized for converting the outcomes of the previous nonlinear filtering stage into defined amplitude level codes. Figures H.69 to H.72 show the result of the quantization stage for the fourth, $m_{2|||^4}$, to the seventh, $m_{2|||^7}$, components respectively. The DNS amplification DDoS attack is confirmed having a more defined beginning and end in Figs. H.70 and H.71, but not as well defined as in the quantization stages from the previous methodologies used, cumulative sum and ZCR. The H&R DDoS attack is confirmed as an inverted peak in Figs. H.70 to H.72. The visual quality in the quantized waveforms favours best the detection of the H&R DDoS attack in Fig. H.72.

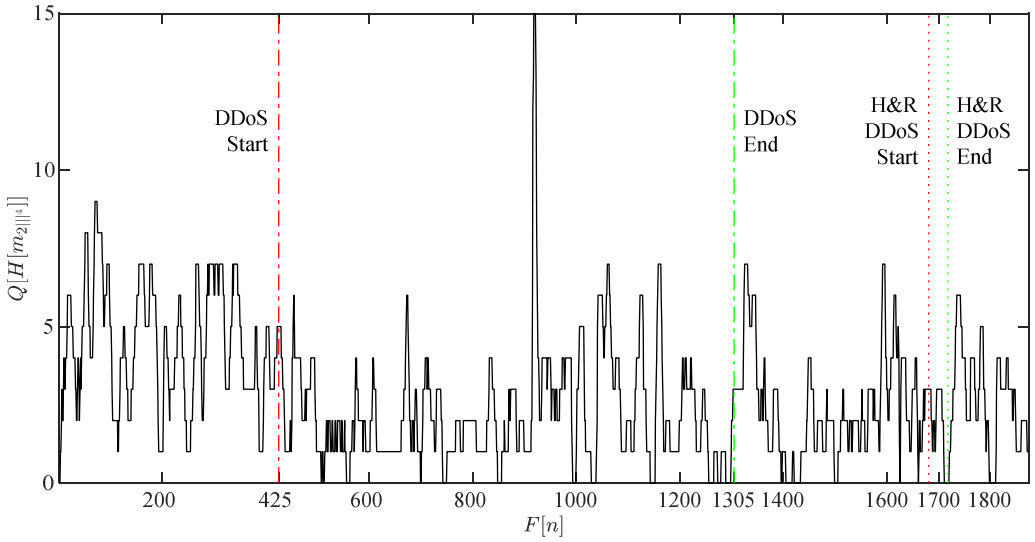


Fig. H.69. Shannon's entropy *H* applied to the variance multiscalor 4th component ($m_{2\|\|^4}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

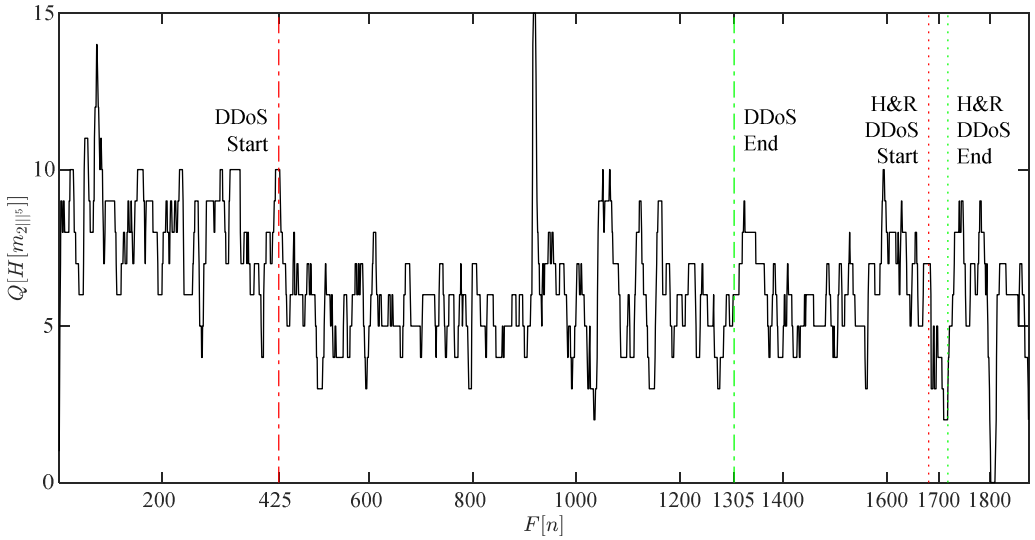


Fig. H.70. Shannon's entropy *H* applied to the variance multiscalor 5th component ($m_{2\|\|^5}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

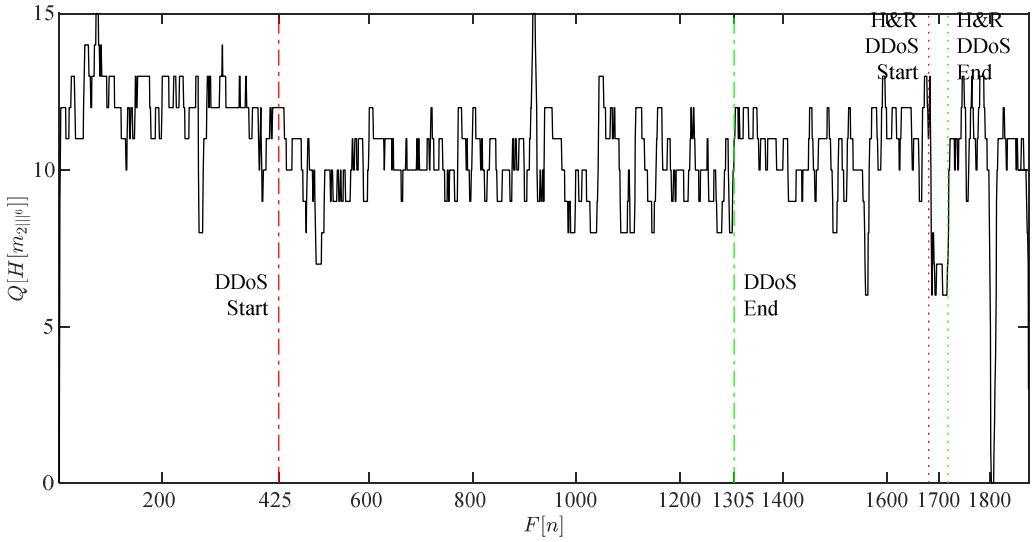


Fig. H.71. Shannon's entropy *H* applied to the variance multiscalor 6th component $(m_{2\|\|^6})$ quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

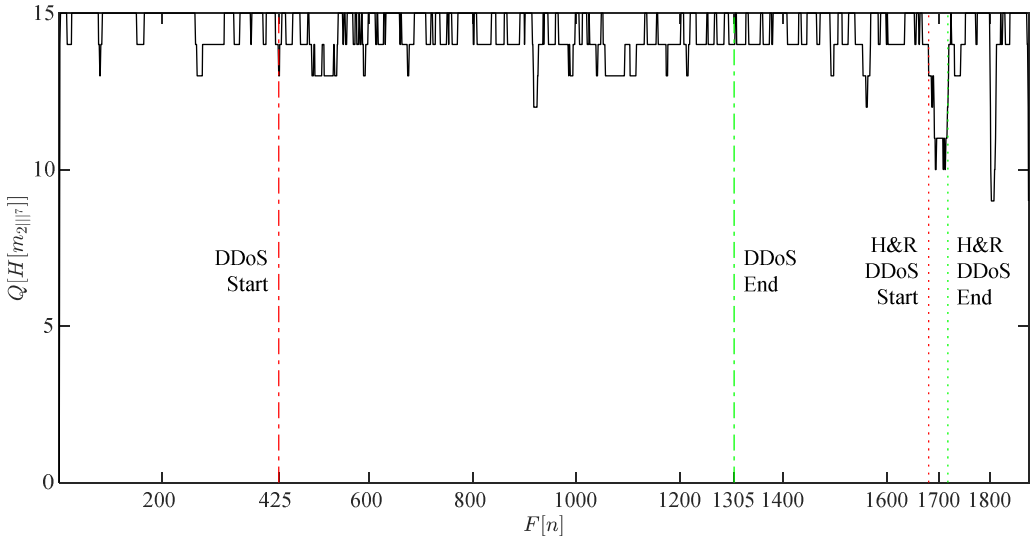


Fig. H.72. Shannon's entropy *H* applied to the variance multiscalor 7th component $(m_{2\|\|^7})$ quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

APPENDIX I

RESULTS OF SELECTED SECONDARY OPERATORS APPLIED TO SKEWNESS MULTISCALOR

I.1 Cumulative Sum

I.1.1 Cumulative Sum Applied to Skewness Multiscalor Components

The cumulative sum applied to the skewness multiscalor components ($S[m_{3\|\|^n}]$) provides results of different visual perception quality for the skewness multiscalor components (from first, $m_{3\|\|^1}$, to seventh, $m_{3\|\|^7}$) as seen in Figs. I.1 to I.7. The DNS amplification DDoS attack appears to have better quality for the first, $m_{3\|\|^1}$ shown in Fig. I.1, and fourth, $m_{3\|\|^4}$ shown in Fig. I.4, skewness multiscalor components, while a lesser quality for the second, $m_{3\|\|^2}$ shown in Fig. I.2, third, $m_{3\|\|^3}$ shown in Fig. I.3, fifth, $m_{3\|\|^5}$ shown in Fig. I.5, and not distinguishable contributions for the rest of the components. For the H&R DDoS attack case, this exhibits better quality in the sixth, $m_{3\|\|^6}$ shown in Fig. N.6, and seventh, $m_{3\|\|^7}$ shown in Fig. N.7. The shape of both DDoS attacks for the cumulative sum run on all skewness multiscalor components is preserved in results of varying quality degrees. These results are not as uniform as the ones obtained with the cumulative sum applied to the variance multiscalor components.

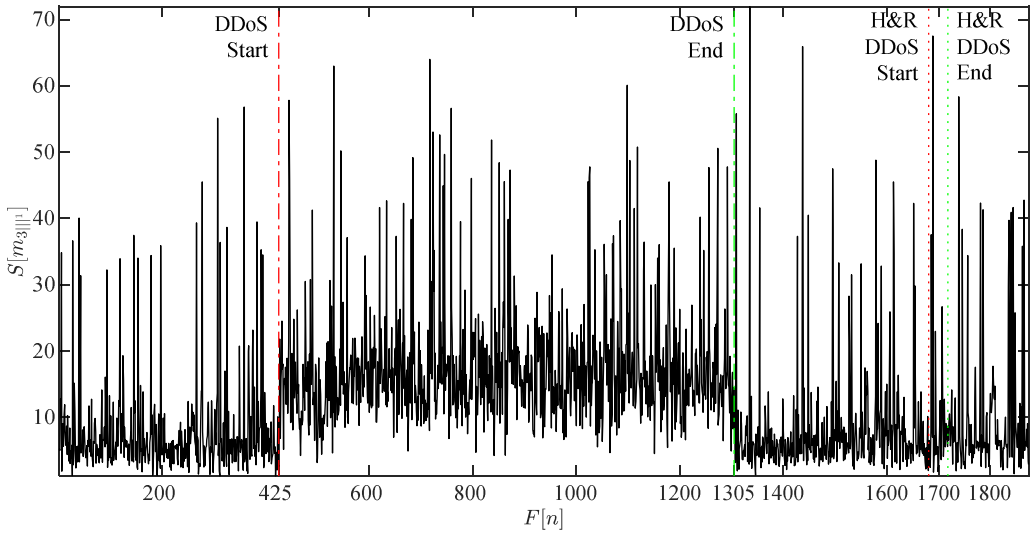


Fig. I.1. Cumulative sum *S* applied to the skewness multiscalor 1st component ($m_{3||^1}$). A processing *frame* of 256 samples is used.

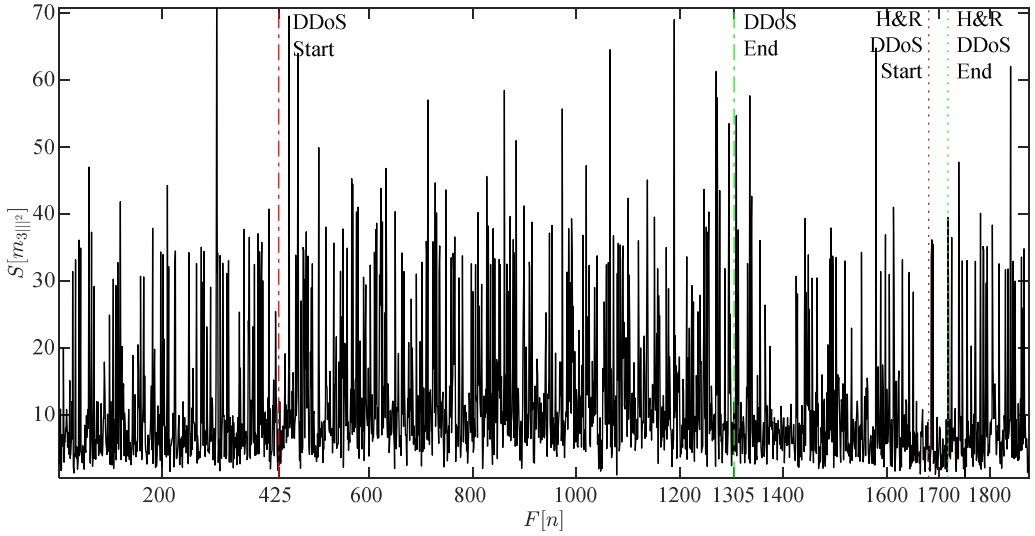


Fig. I.2. Cumulative sum *S* applied to the skewness multiscalor 2nd component ($m_{3||^2}$). A processing *frame* of 256 samples is used.

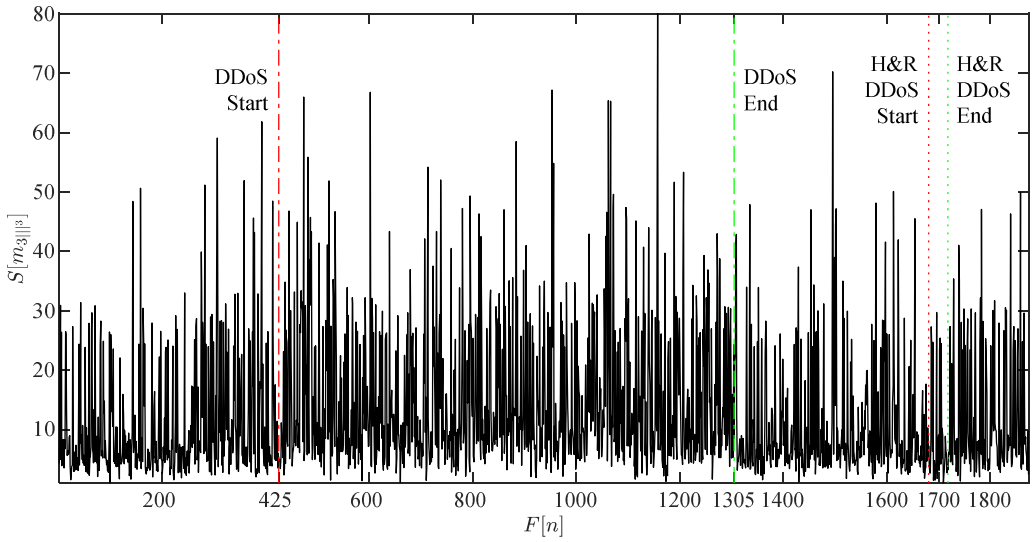


Fig. I.3. Cumulative sum *S* applied to the skewness multiscalor 3rd component ($m_{3\|\|^3}$). A processing *frame* of 256 samples is used.

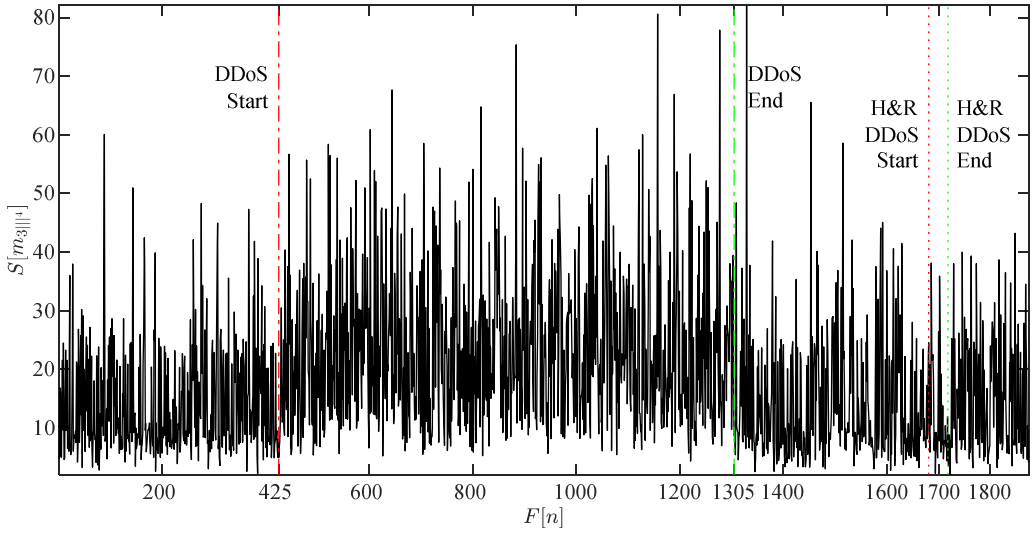


Fig. I.4. Cumulative sum *S* applied to the skewness multiscalor 4th component ($m_{3\|\|^4}$). A processing *frame* of 256 samples is used.

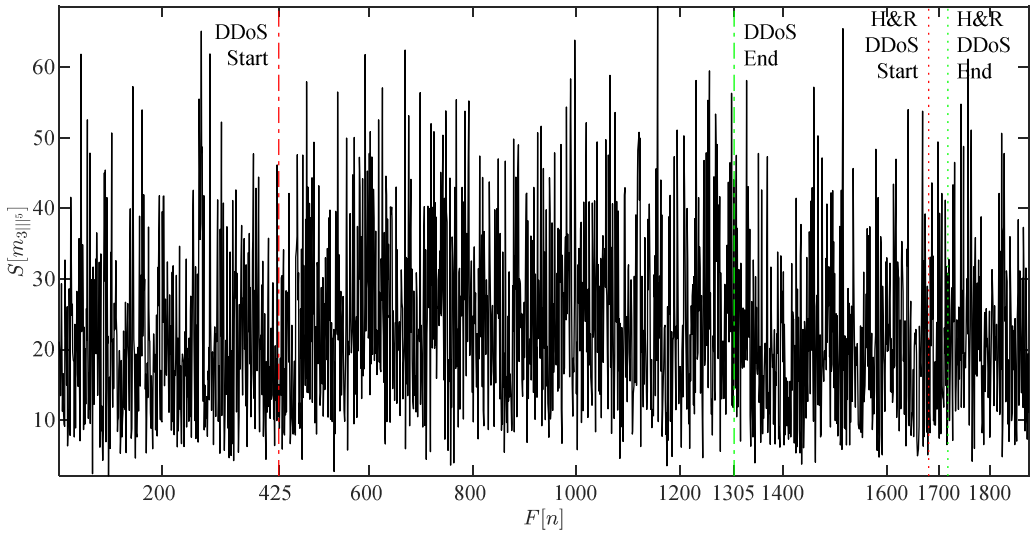


Fig. I.5. Cumulative sum *S* applied to the skewness multiscalor 5th component $(m_{3||^5})$. A processing *frame* of 256 samples is used.

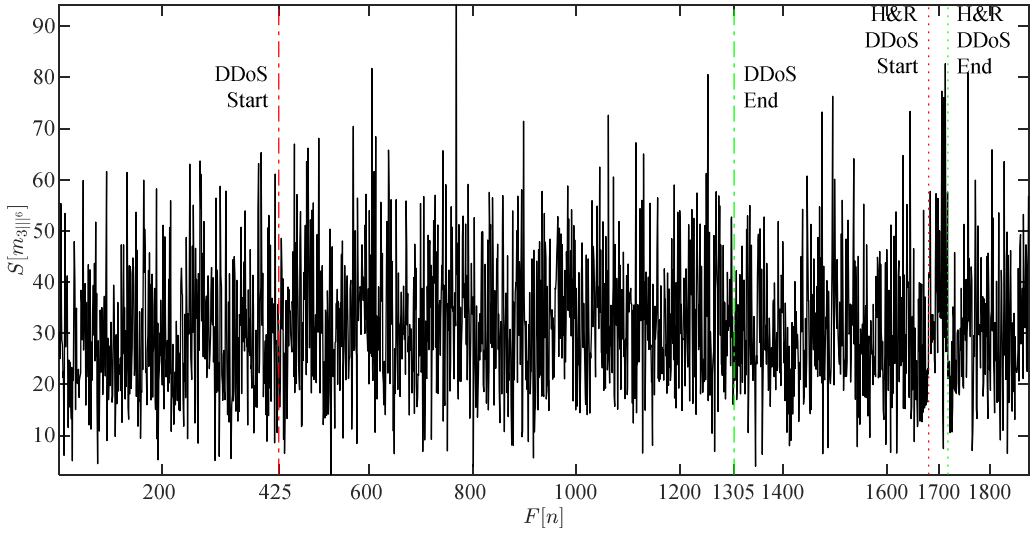


Fig. I.6. Cumulative sum *S* applied to the skewness multiscalor 6th component $(m_{3||^6})$. A processing *frame* of 256 samples is used.

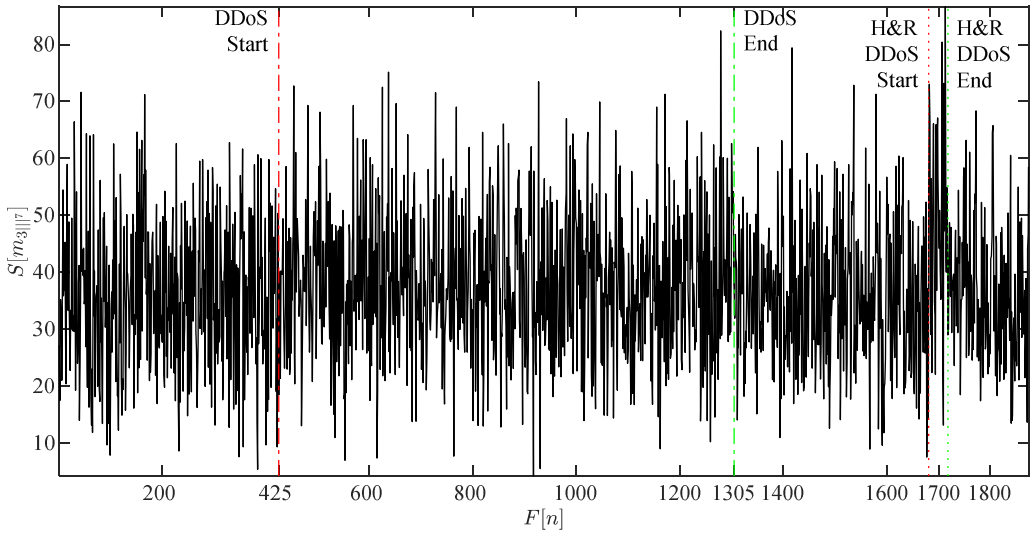


Fig. I.7. Cumulative sum *S* applied to the skewness multiscalor 7th component ($m_{3|||^7}$). A processing *frame* of 256 samples is used.

I.1.2 Cumulative Sum Applied to Skewness Multiscalor Components After Donoho's Denoising

Running Donoho's denoising to the outcomes of cumulative sum applied to the skewness multiscalor components ($S[m_{3|||^n}]$) provides results of different visual perception quality for the skewness multiscalor components (from first, $m_{3|||^1}$, to seventh, $m_{3|||^7}$) as seen in Figs. I.8 to I.14. After applying Donoho's denoising, all skewness multiscalor components are smoothed, but the DNS amplification DDoS attack appearing with better quality for the first, $m_{3|||^1}$, shown in Fig. I.8, and fourth, $m_{3|||^4}$, shown in Fig. I.11, components, while a lesser quality for the second, $m_{3|||^2}$ shown in Fig. I.9, third, $m_{3|||^3}$ shown in Fig. I.10, and fifth, $m_{3|||^5}$ shown in Fig. I.12, and not distinguishable contributions for the rest of the components, still holds. For the now smoother H&R DDoS attack case, this exhibits better quality only in the seventh, $m_{3|||^7}$, component shown in Fig. I.14 and it has apparently faded in the sixth, $m_{3|||^6}$, component shown in Fig. I.13.

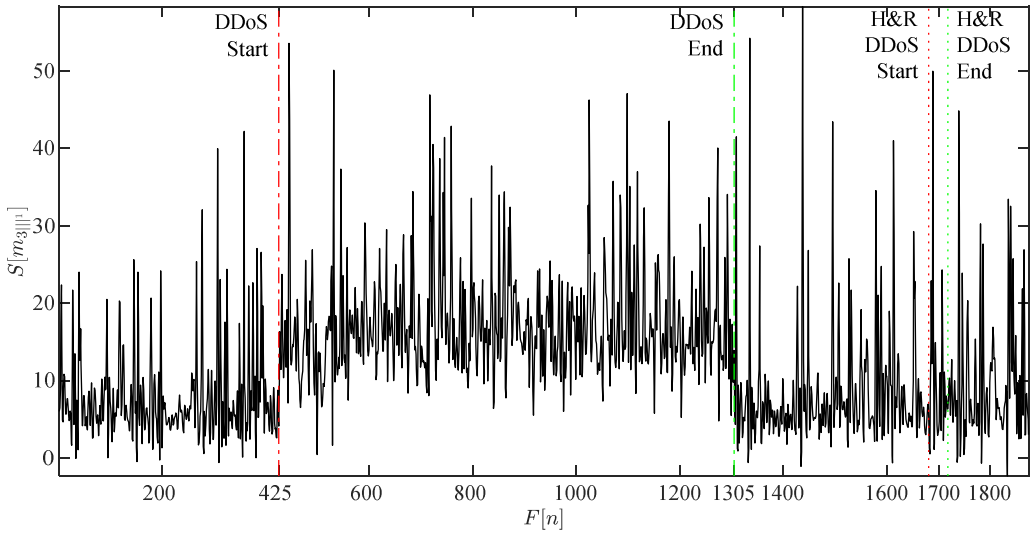


Fig. I.8. Cumulative sum *S* applied to the skewness multiscalor 1st component ($m_{3\|\|^1}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

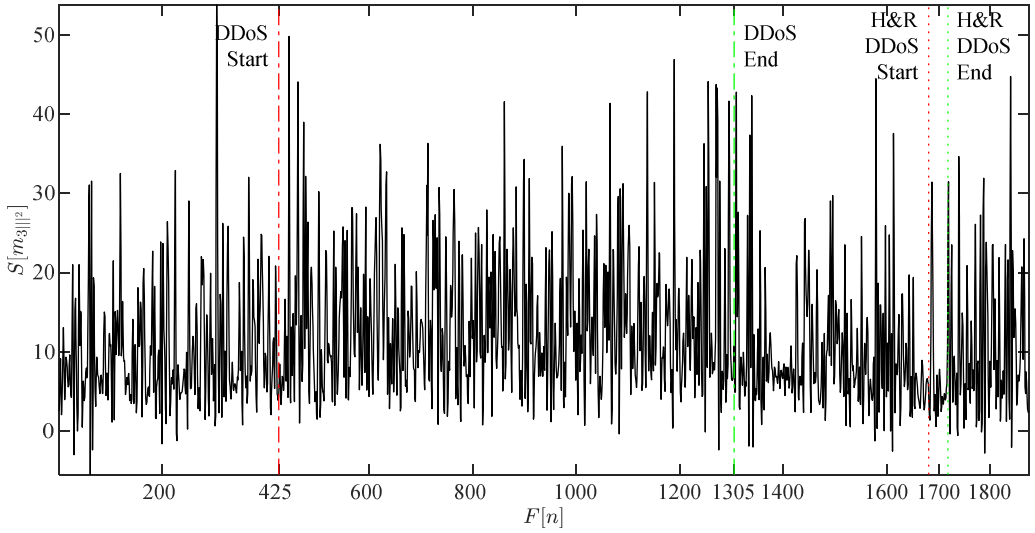


Fig. I.9. Cumulative sum *S* applied to the skewness multiscalor 2nd component ($m_{3\|\|^2}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

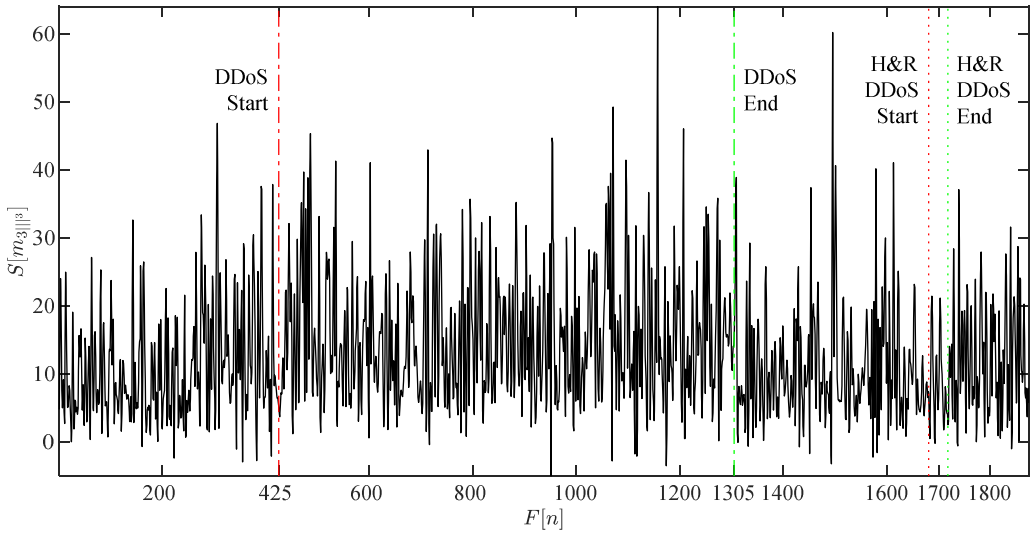


Fig. I.10. Cumulative sum *S* applied to the skewness multiscalor 3rd component ($m_{3|||^3}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

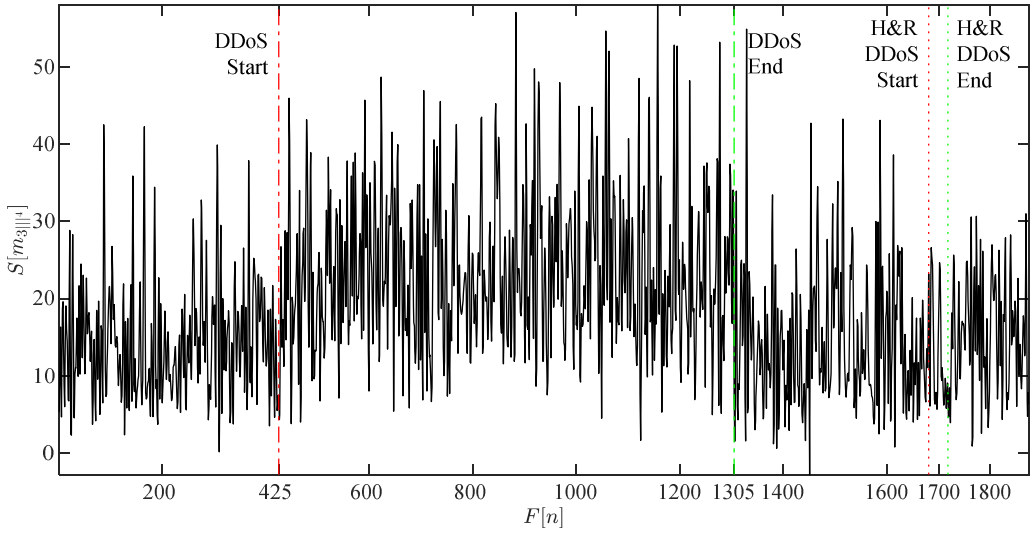


Fig. I.11. Cumulative sum *S* applied to the skewness multiscalor 4th component ($m_{3|||^4}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

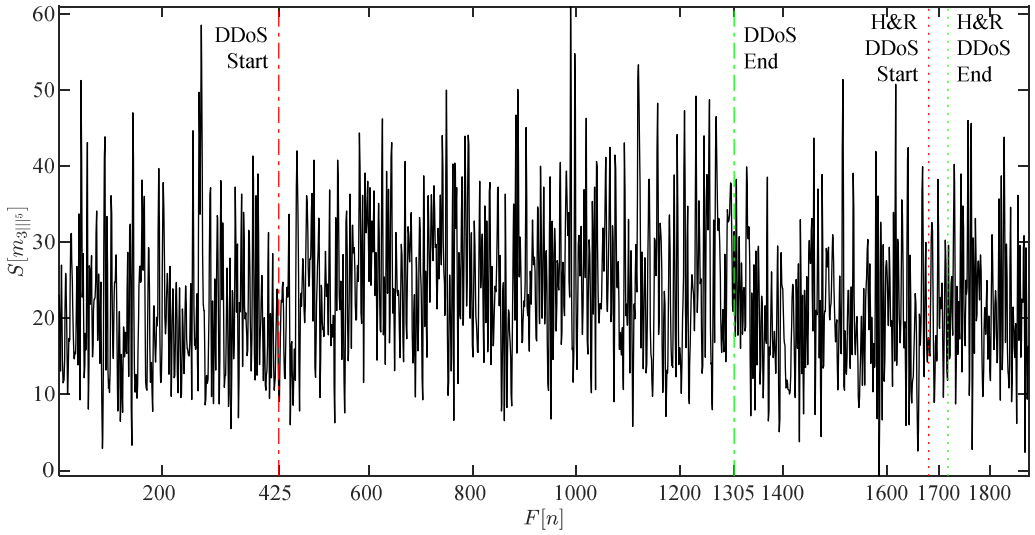


Fig. I.12. Cumulative sum *S* applied to the skewness multiscalor 5th component ($m_{3||^5}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

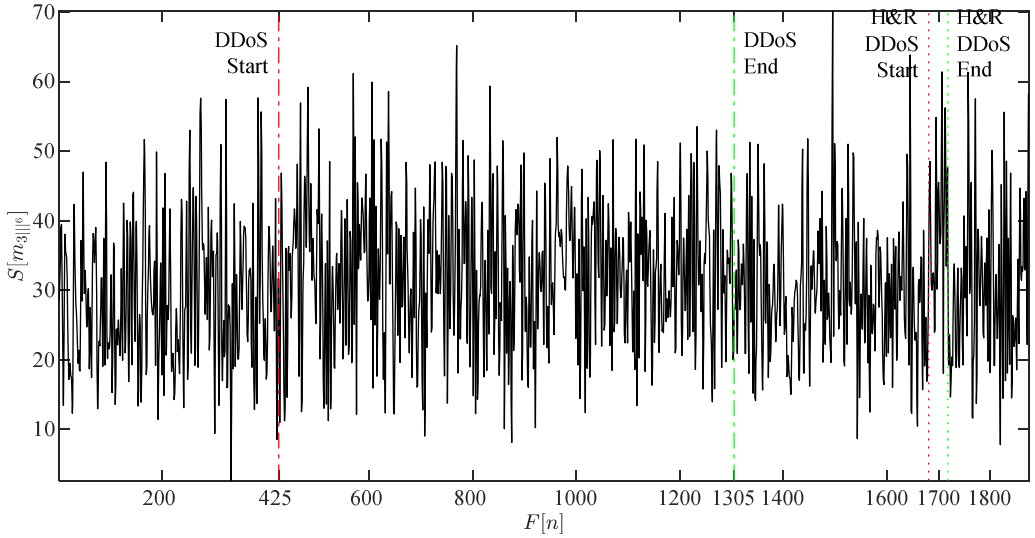


Fig. I.13. Cumulative sum *S* applied to the skewness multiscalor 6th component ($m_{3||^6}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

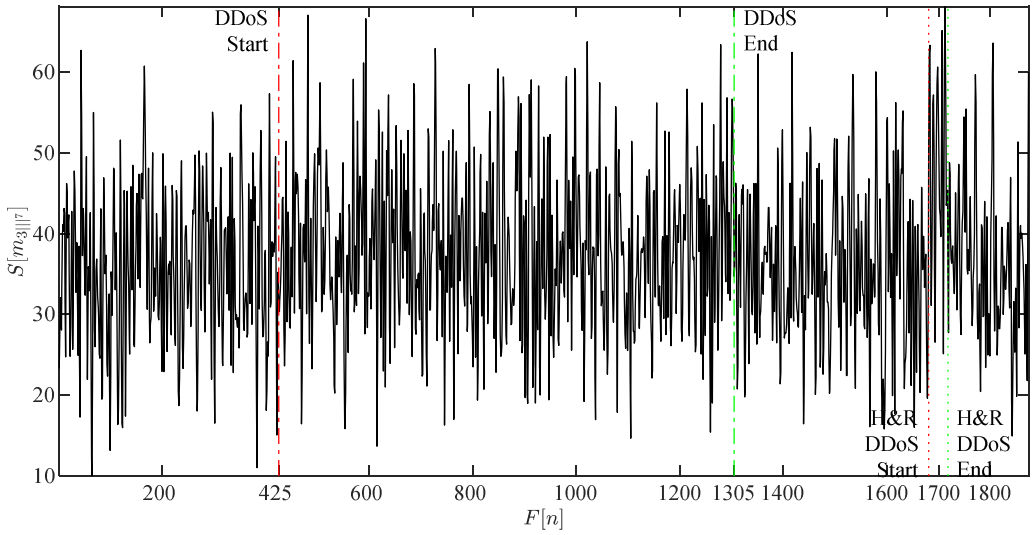


Fig. I.14. Cumulative sum *S* applied to the skewness multiscalor 7th component ($m_{3|||^7}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

I.1.3 Cumulative Sum Applied to Skewness Multiscalor Components Non-Linearly Filtered After Donoho's Denoising

Continuing with the subsequent nonlinear filtering stage is applied to the skewness multiscalor components after Donoho's denoising, smoother features are yielded as shown in Figs. I.15 to I.21 the skewness multiscalor components corresponding to the first, $m_{3|||^1}$, to seventh, $m_{3|||^7}$. After the nonlinear data processing, the DNS amplification DDoS attack is visibly having better visual quality (in the context of the skewness multiscalors because the variance multiscalors are by far better) in the first, $m_{3|||^1}$, shown in Fig. I.15, and fourth, $m_{3|||^4}$, shown in Fig. I.18, components, lesser quality for the second, $m_{3|||^2}$ shown in Fig. I.16, third, $m_{3|||^3}$ shown in Fig. I.17, and fifth, $m_{3|||^5}$ shown in Fig. I.18, and not distinguishable contributions for the rest of the components. The H&R DDoS attack exhibits better quality in the sixth, $m_{3|||^6}$, and in the seventh, $m_{3|||^7}$, components shown in Figs. I.20 and I.21 respectively.

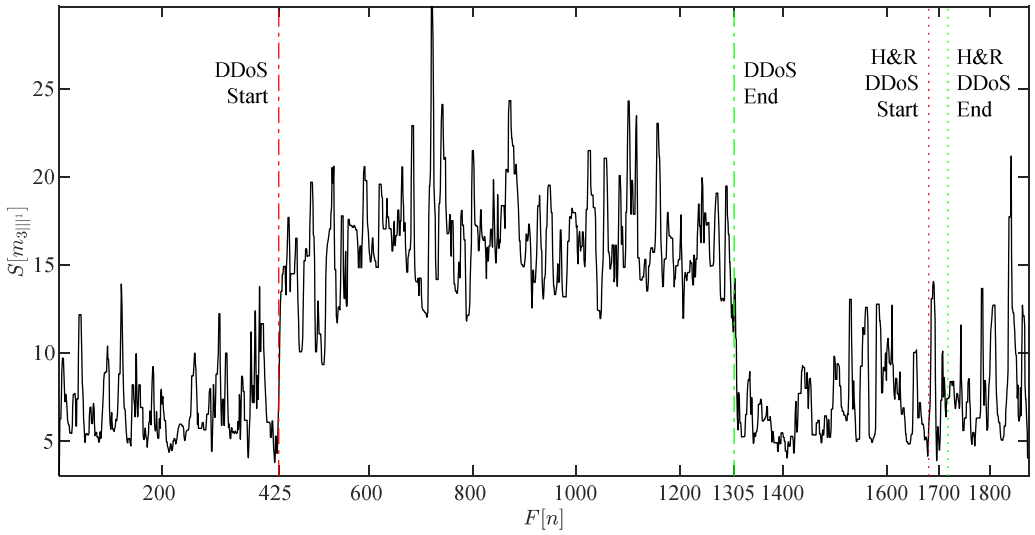


Fig. I.15. Cumulative sum *S* applied to the skewness multiscalor 1st component ($m_{3\|\|^1}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

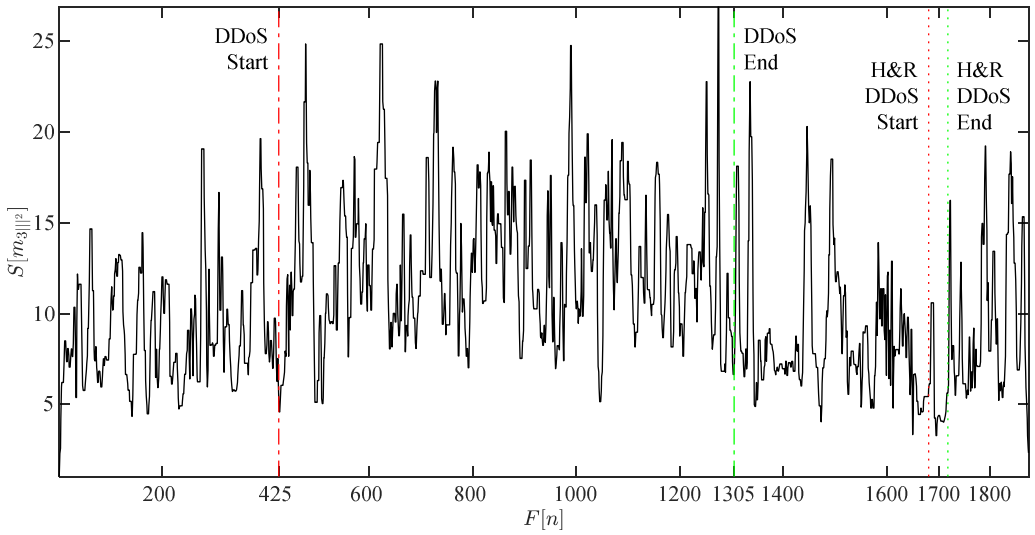


Fig. I.16. Cumulative sum *S* applied to the skewness multiscalor 2nd component ($m_{3\|\|^2}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

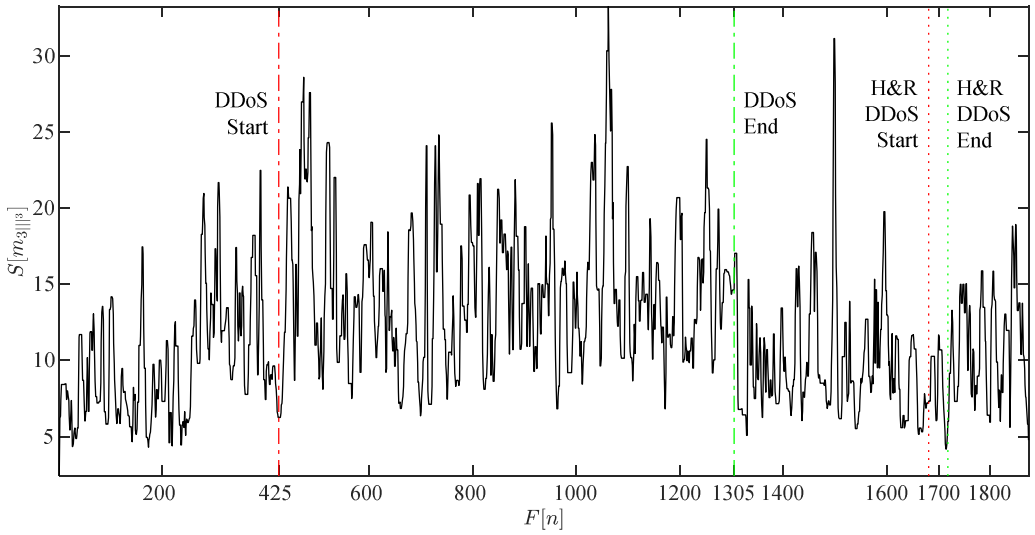


Fig. I.17. Cumulative sum *S* applied to the skewness multiscalor 3rd component ($m_{3\|\|^3}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

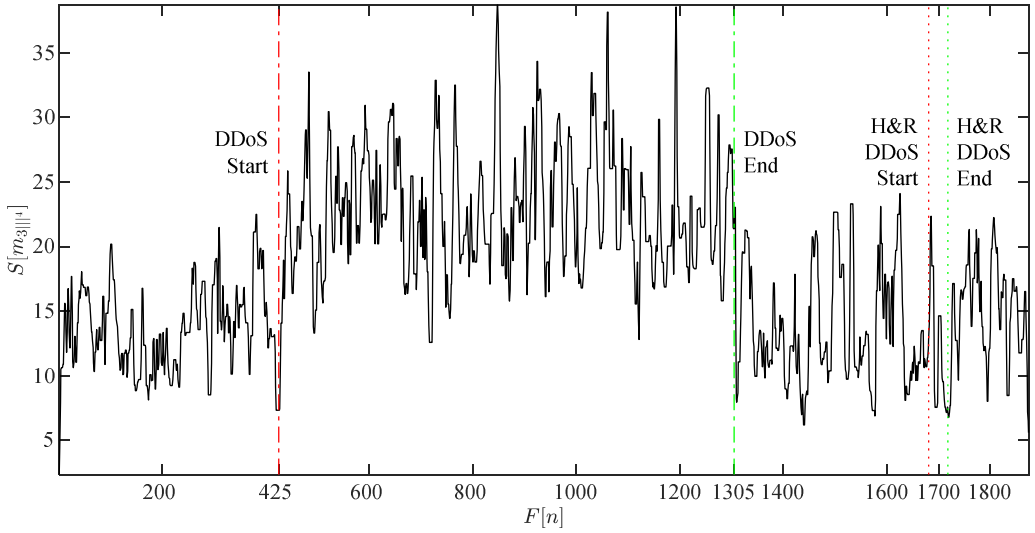


Fig. I.18. Cumulative sum *S* applied to the skewness multiscalor 4th component ($m_{3\|\|^4}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

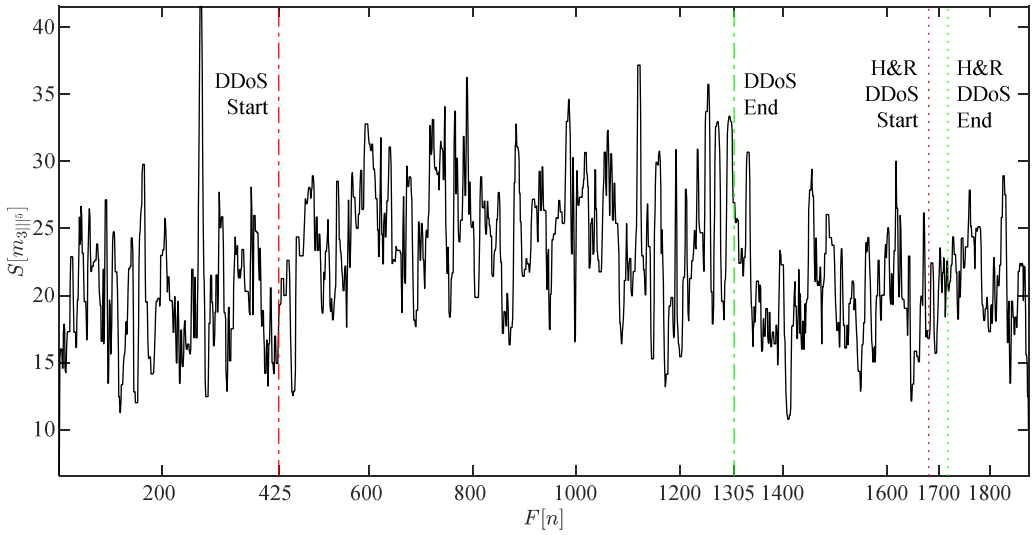


Fig. I.19. Cumulative sum *S* applied to the skewness multiscalor 5th component ($m_{3||^5}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

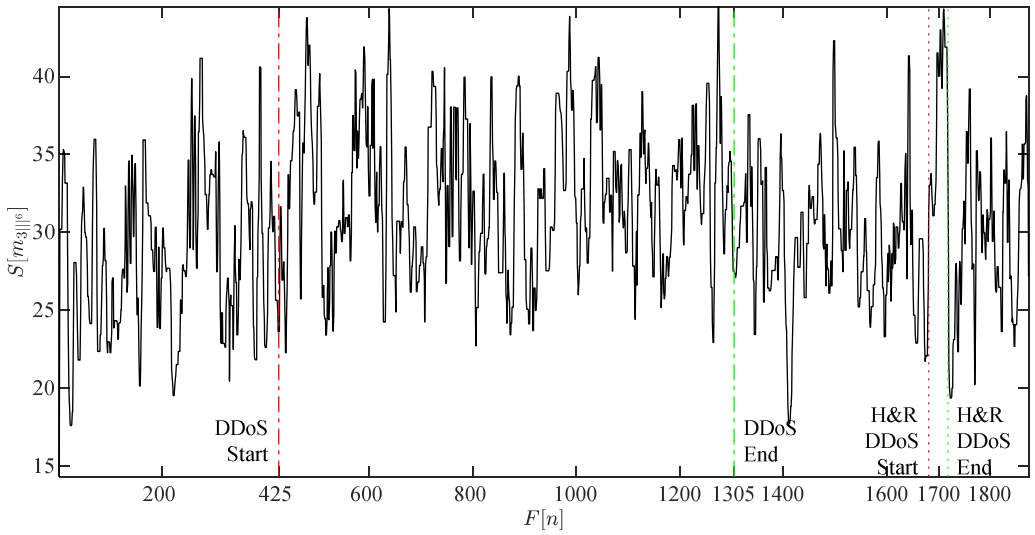


Fig. I.20. Cumulative sum *S* applied to the skewness multiscalor 6th component ($m_{3||^6}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

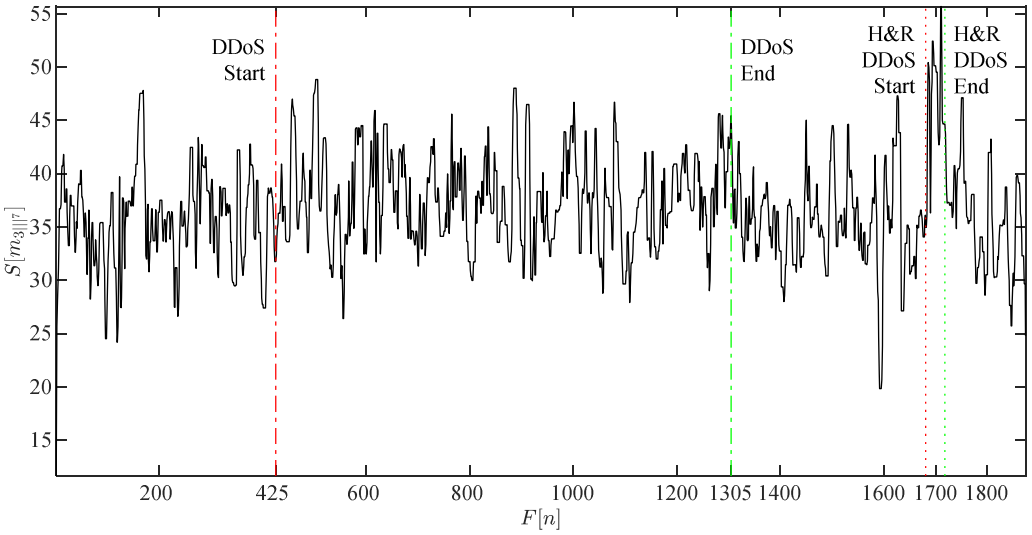


Fig. I.21. Cumulative sum *S* applied to the skewness multiscalor 7th component ($m_{3|||^7}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

I.1.4 Cumulative Sum Applied to Skewness Multiscalor Components Quantization of Non-Linear Filtering After Donoho's Denoising

As previously, the extracted features through cumulative sum applied to the skewness multiscalors are readied for ART1 via Lloyd's quantization to transform them into defined amplitude level codes that can be translated into binary representation. The figures I.22 to I.28 show the skewness multiscalor components corresponding to the first, $m_{3|||^1}$, to seventh, $m_{3|||^7}$ Lloyd's quantization after the cumulative sum has been denoised with Donoho's methodology and nonlinear filtering. After the nonlinear data processing, the DNS amplification DDoS attack is confirmed to have better visual quality (in the context of the skewness multiscalors because the variance multiscalors appear visibly better) in the first, $m_{3|||^1}$, shown in Fig. I.22, and fourth, $m_{3|||^4}$, shown in Fig. I.25, components, lesser quality for the second, $m_{3|||^2}$ shown in Fig. I.23, third, $m_{3|||^3}$ shown in Fig. I.24, and fifth, $m_{3|||^5}$ shown in Fig. I.26, and not distinguishable contributions for the rest of the components. Similarly for the H&R DDoS attack, it is confirmed that it has better quality in the sixth, $m_{3|||^6}$, and in the seventh, $m_{3|||^7}$, components shown in Figs. N.27 and N.28 respectively.

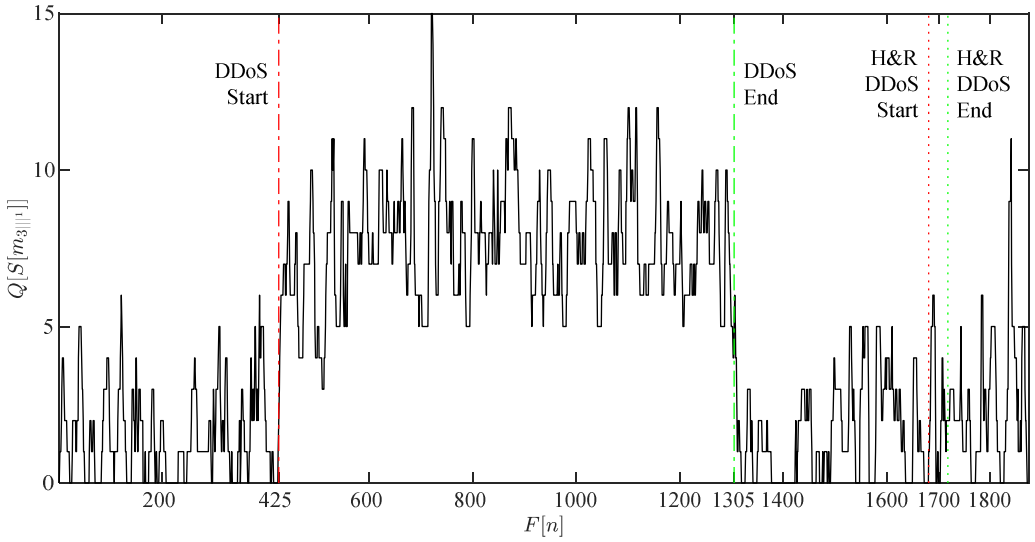


Fig. I.22. Cumulative sum *S* applied to the skewness multiscalor 1st component ($m_{3\|\|^1}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

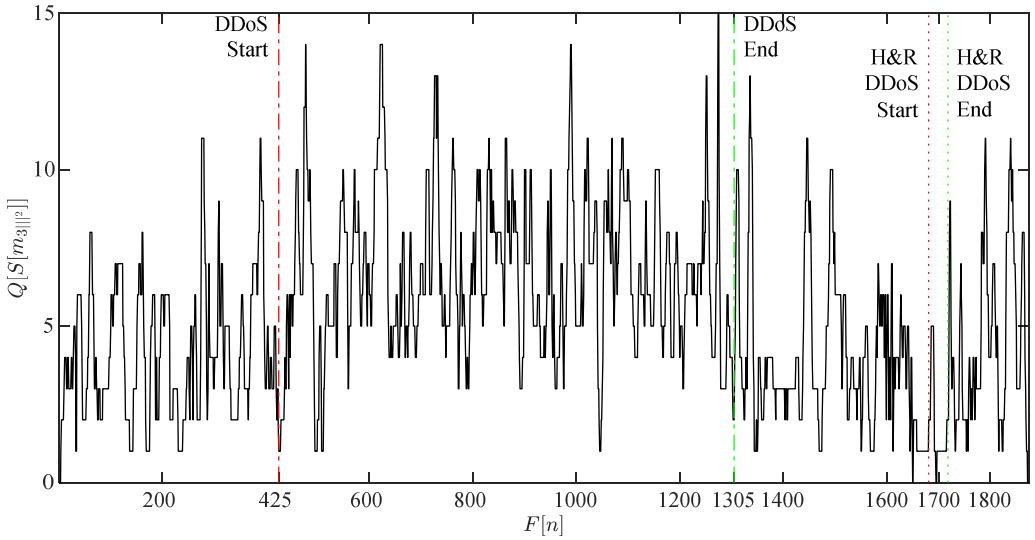


Fig. I.23. Cumulative sum *S* applied to the skewness multiscalor 2nd component ($m_{3\|\|^2}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

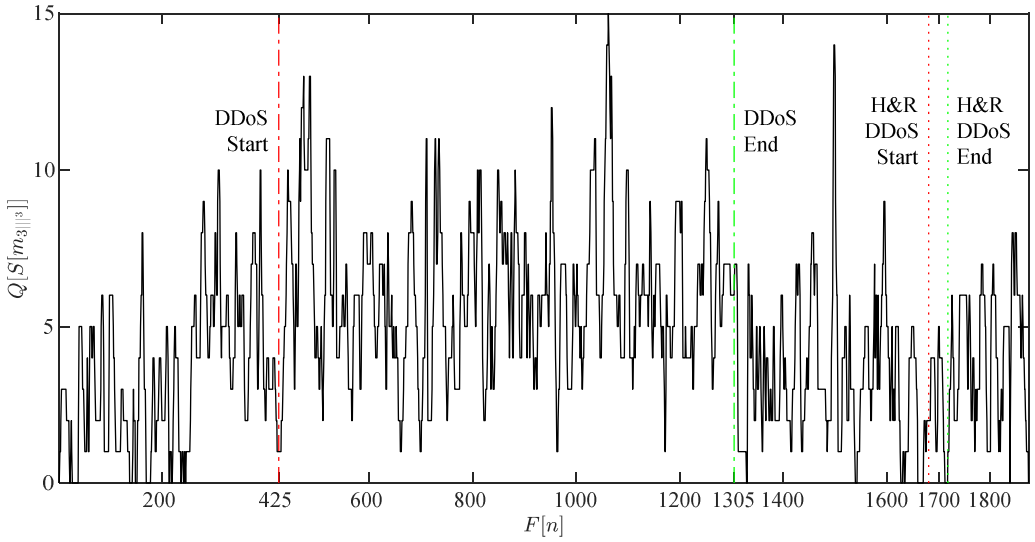


Fig. I.24. Cumulative sum *S* applied to the skewness multiscalor 3rd component ($m_{3\|\|^3}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

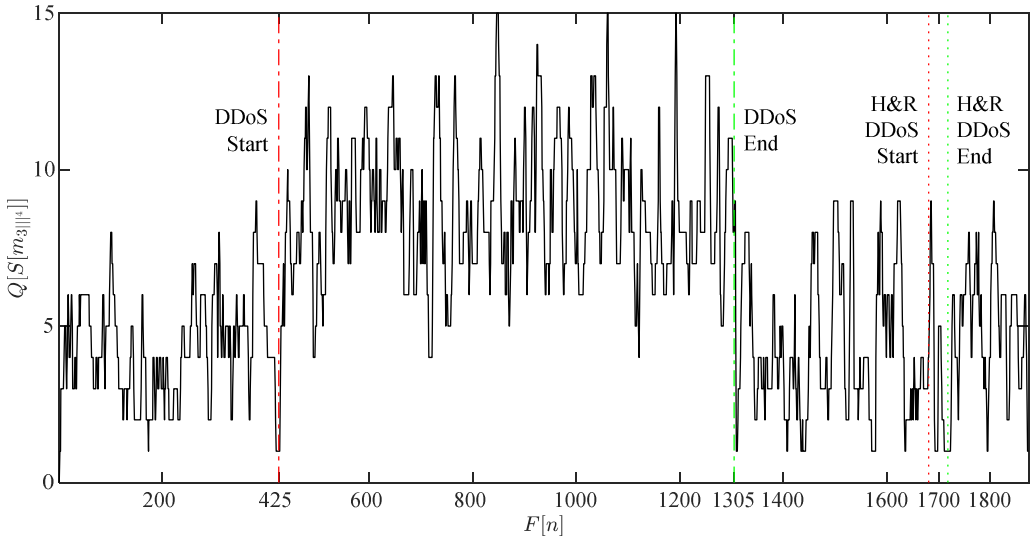


Fig. I.25. Cumulative sum *S* applied to the skewness multiscalor 4th component ($m_{3\|\|^4}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

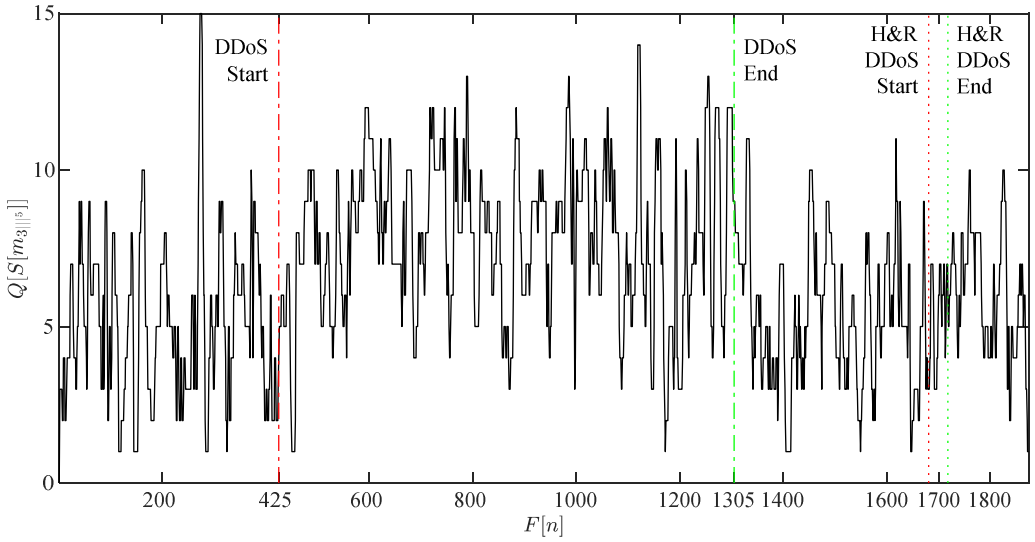


Fig. I.26. Cumulative sum *S* applied to the skewness multiscalor 5th component ($m_{3||^5}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

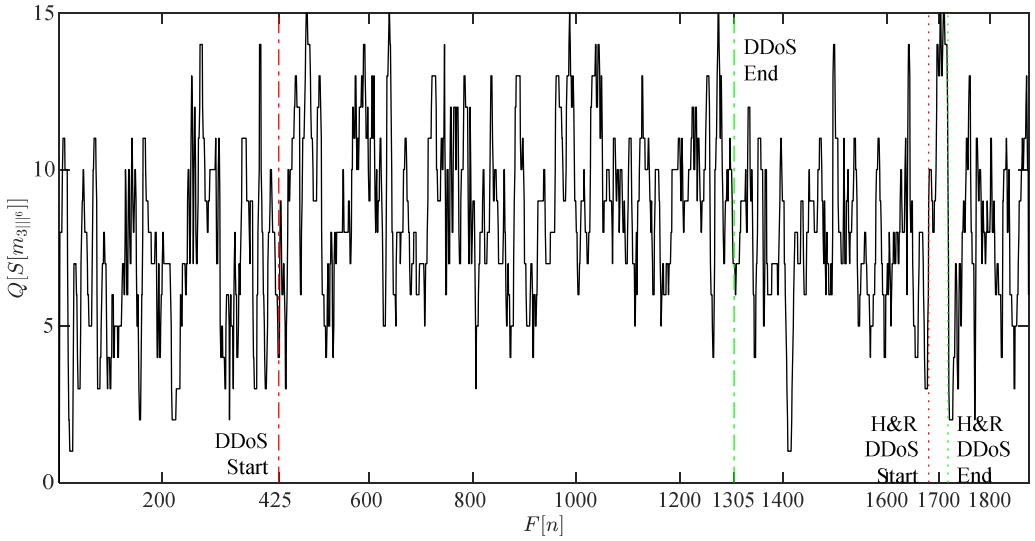


Fig. I.27. Cumulative sum *S* applied to the skewness multiscalor 6th component ($m_{3||^6}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

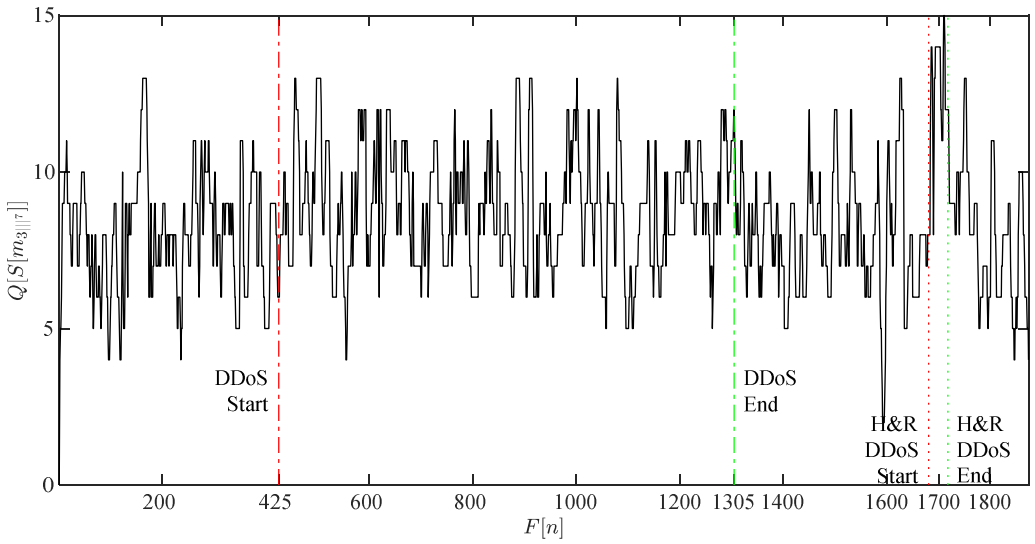


Fig. I.28. Cumulative sum *S* applied to the skewness multiscalor 7th component ($m_{3|||^7}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

I.2 Zero-Crossing Rate

I.2.1 Zero-Crossing Rate Applied to Skewness Multiscalor Components

When ZCR is applied to the skewness multiscalor components ($Z_n[m_{3|||^n}]$), results of different visual perception quality (from first, $m_{3|||^1}$, to seventh, $m_{3|||^7}$), as seen in Figs. I.29 to I.35, are provided. Particularly, the DNS amplification DDoS attack appears to have better quality for the first, $m_{3|||^1}$ shown in Fig. I.29, and a lesser quality for the fourth, $m_{3|||^4}$ shown in Fig. I.32, and not distinguishable contributions for the rest of the components. Related to the H&R DDoS attack case, this exhibits better quality from the fifth, $m_{3|||^5}$ shown in Fig. I.33, to the seventh, $m_{3|||^7}$ shown in Fig. I.35. The shape of both DDoS attacks for the ZCR applied on all skewness multiscalor components appears in varying quality degrees. The results obtained from ZCR applied to the skewness multiscalor components are not as good as the ones obtained from the variance multiscalor components.

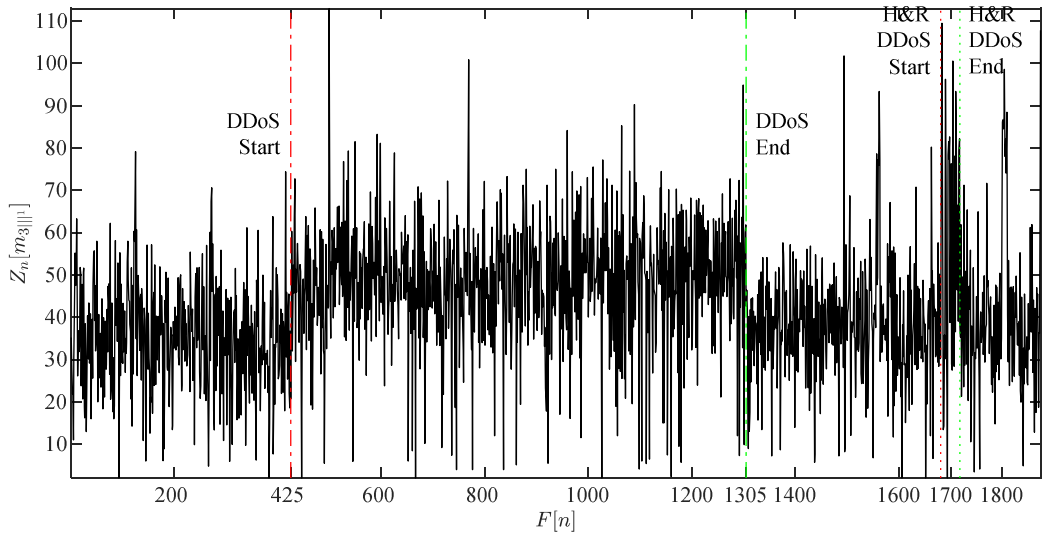


Fig. I.29. ZCR Z_n applied to the skewness multiscalor 1st component $(m_{3\|\|^1})$. A processing *frame* of 256 samples is used.

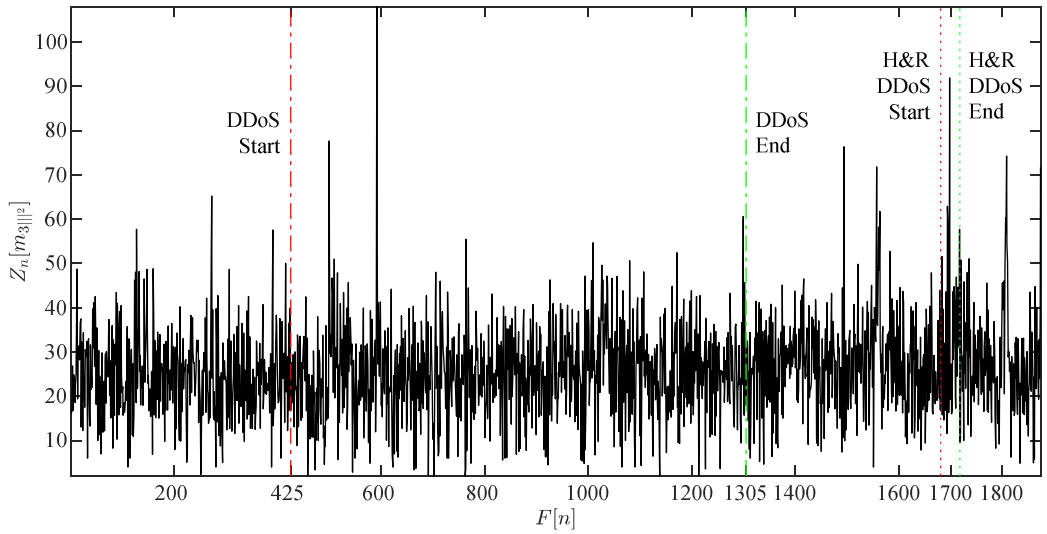


Fig. I.30. ZCR Z_n applied to the skewness multiscalor 2nd component $(m_{3\|\|^2})$. A processing *frame* of 256 samples is used.

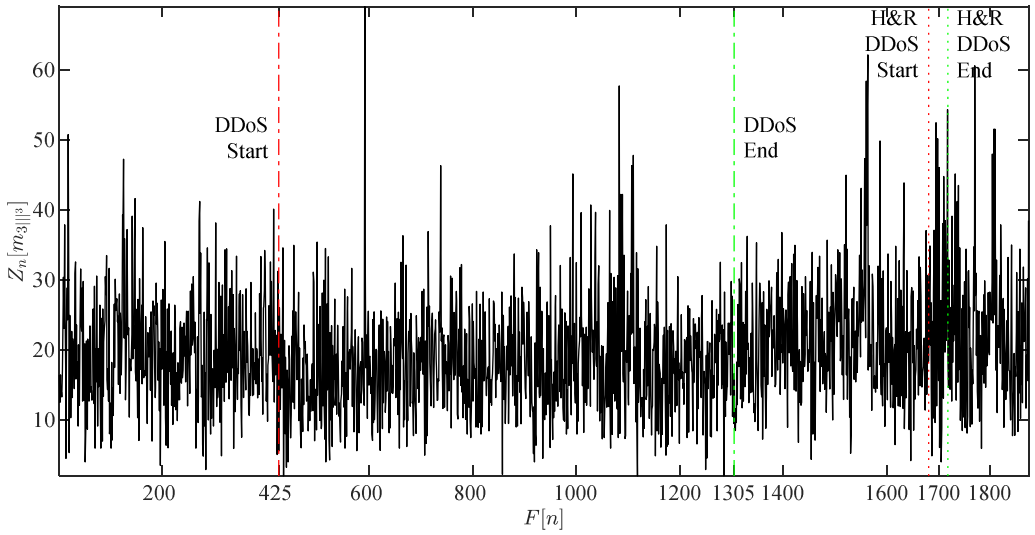


Fig. I.31. ZCR Z_n applied to the skewness multiscalor 3rd component $(m_{3\|\|^3})$. A processing *frame* of 256 samples is used.

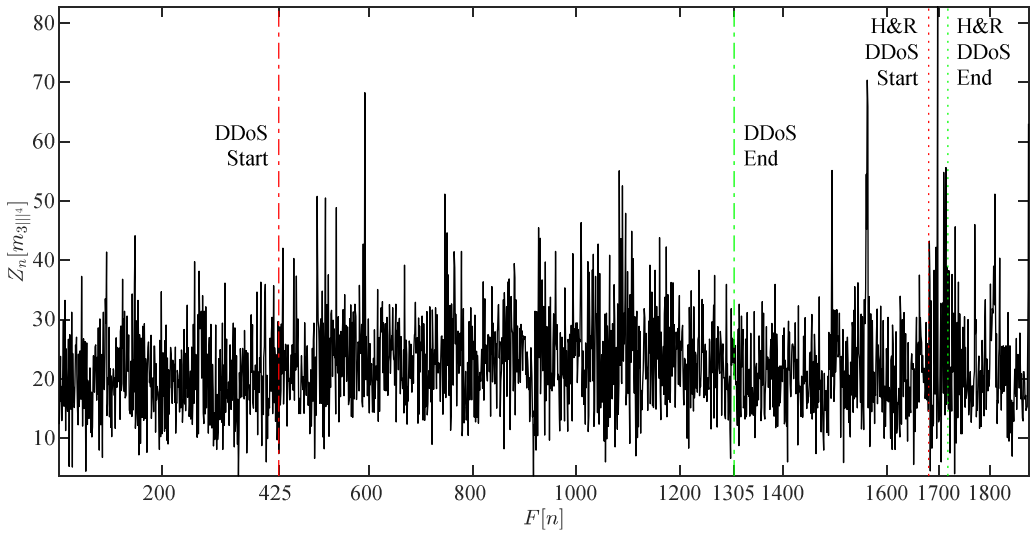


Fig. I.32. ZCR Z_n applied to the skewness multiscalor 4th component $(m_{3\|\|^4})$. A processing *frame* of 256 samples is used.

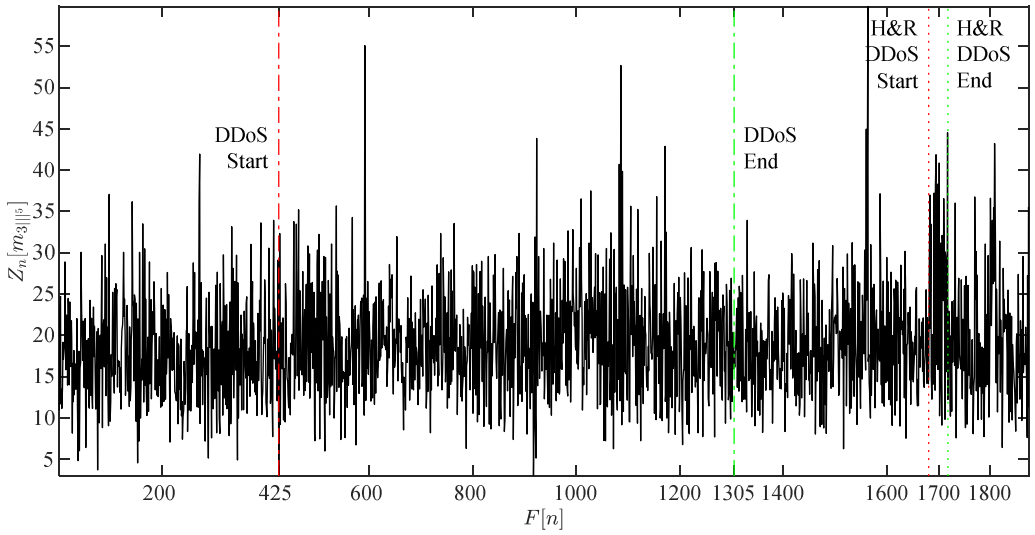


Fig. I.33. ZCR Z_n applied to the skewness multiscalor 5th component $(m_{3||^5})$. A processing *frame* of 256 samples is used.

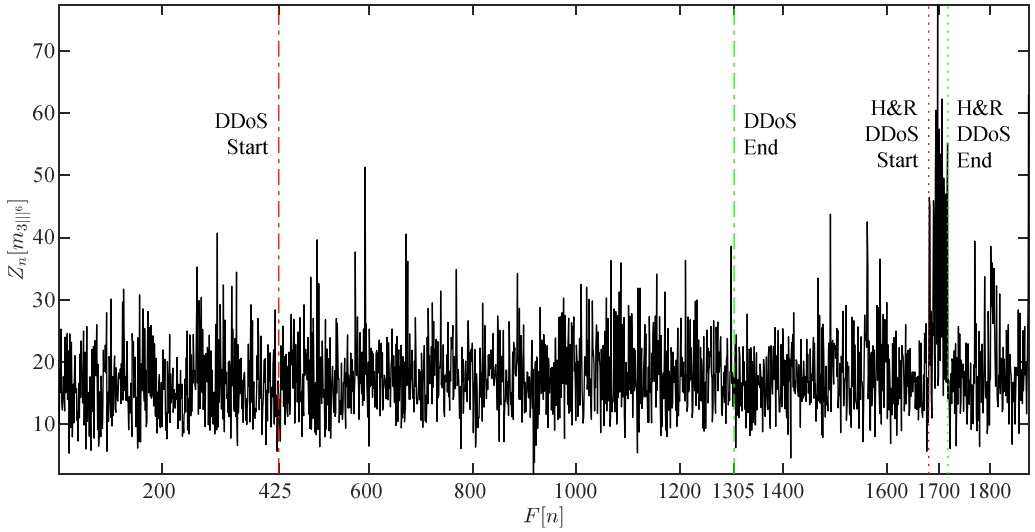


Fig. I.34. ZCR Z_n applied to the skewness multiscalor 6th component $(m_{3||^6})$. A processing *frame* of 256 samples is used.

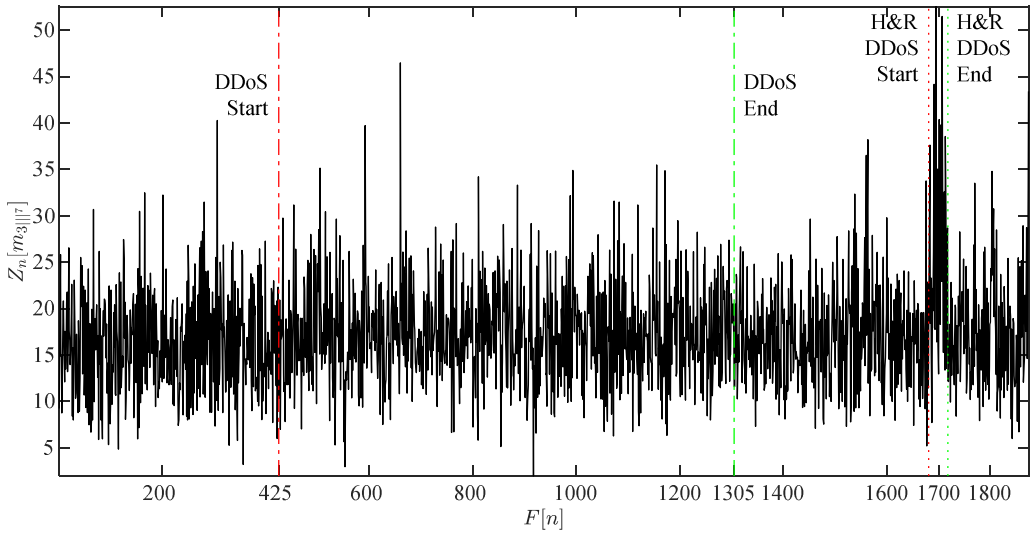


Fig. I.35. ZCR Z_n applied to the skewness multiscalor 7th component $(m_{3|||^7})$. A processing *frame* of 256 samples is used.

I.2.2 Zero-Crossing Rate Applied to Skewness Multiscalor Components After Donoho's Denoising

For the Donoho's denoising of the ZCR applied to the skewness multiscalor components $(Z_n[m_{3|||^n}])$, the results of different visual perception quality (from first, $m_{3|||^1}$, to seventh, $m_{3|||^7}$), as seen in Figs. N.36 to N.42, are smoothed. Particularly, the DNS amplification DDoS attack appears to have better quality for the first, $m_{3|||^1}$ shown in Fig. I.36, and a lesser quality for the fourth, $m_{3|||^4}$ shown in Fig. I.39, and not clear contributions for the rest of the components. Related to the H&R DDoS attack case, this exhibits better quality from the fifth, $m_{3|||^5}$ shown in Fig. I.40, to the seventh, $m_{3|||^7}$ shown in Fig. I.42. The shape of both DDoS attacks, when utilizing Donoho's denoising in results of the ZCR for all skewness multiscalor components, starts to appear smooth.

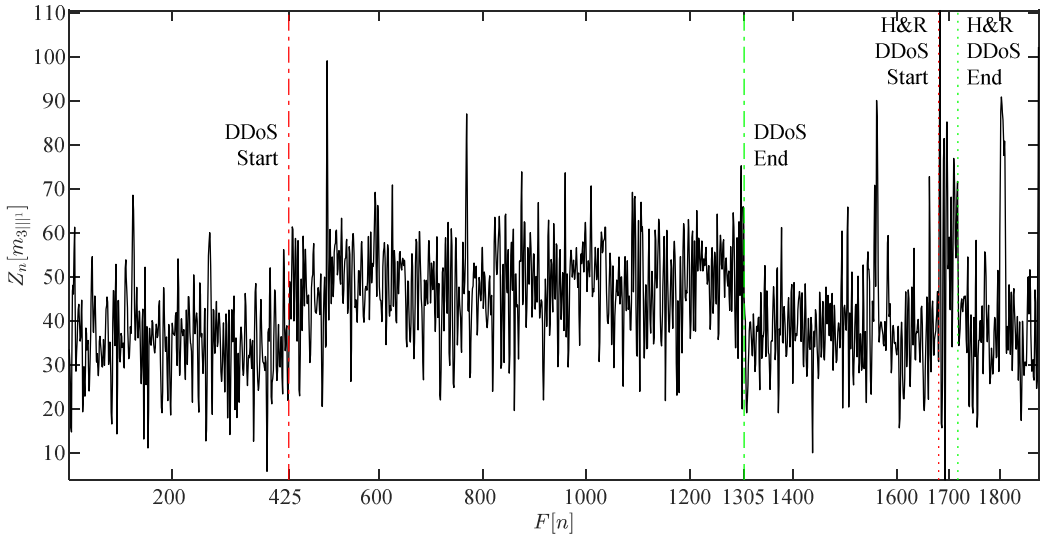


Fig. I.36. ZCR Z_n applied to the skewness multiscalor 1st component $(m_{3|||^1})$ after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

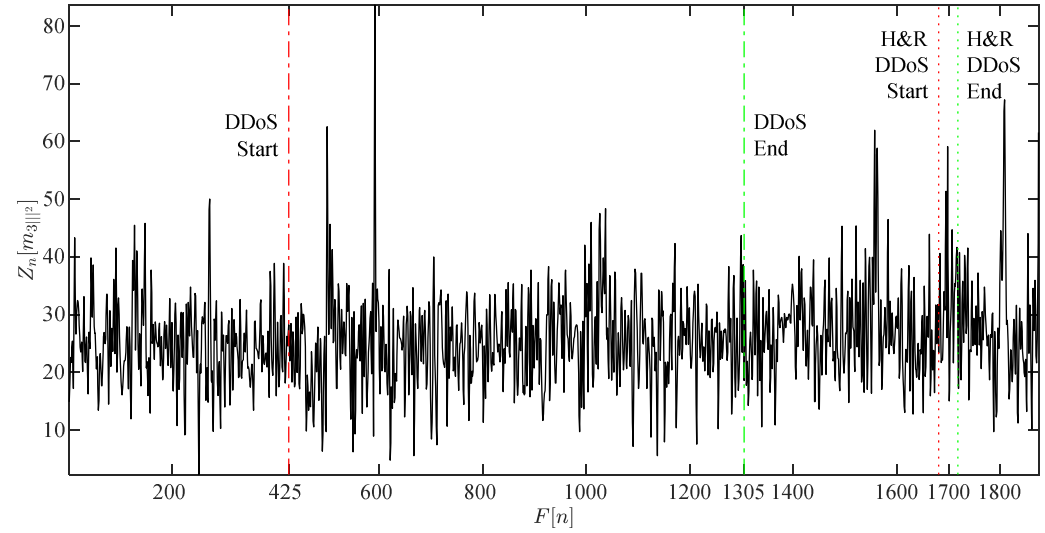


Fig. I.37. ZCR Z_n applied to the skewness multiscalor 2nd component $(m_{3|||^2})$ after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

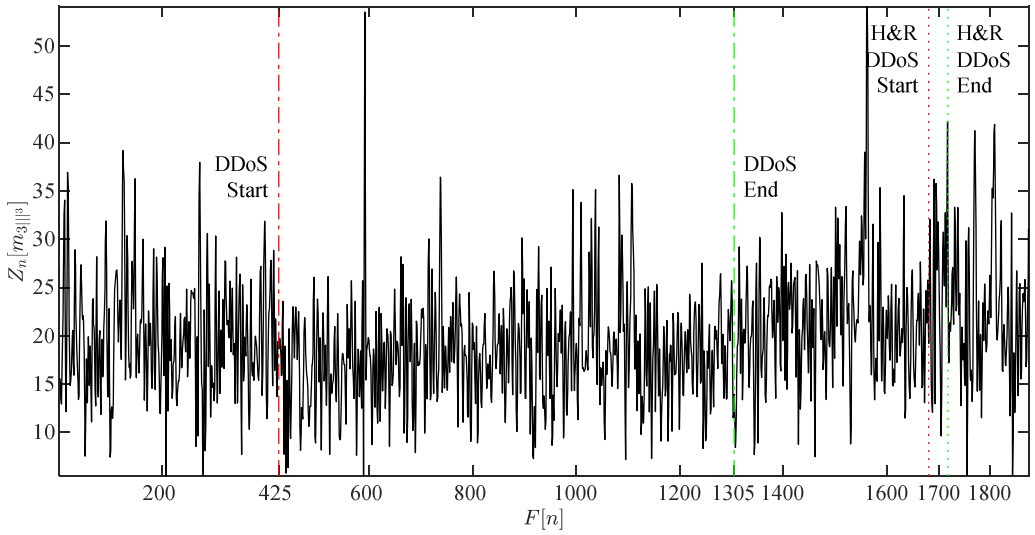


Fig. I.38. ZCR Z_n applied to the skewness multiscalor 3rd component $(m_{3\|\|^3})$ after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

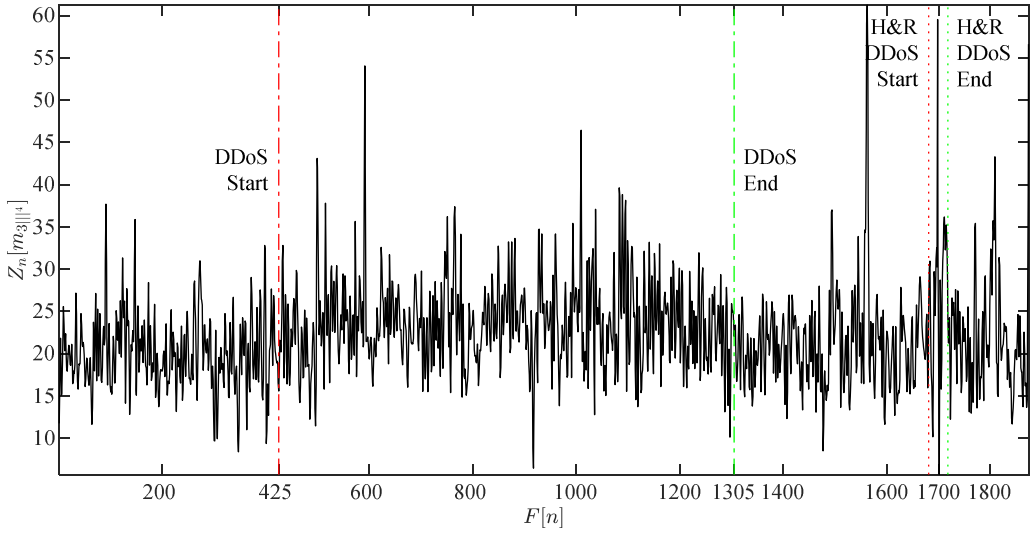


Fig. I.39. ZCR Z_n applied to the skewness multiscalor 4th component $(m_{3\|\|^4})$ after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

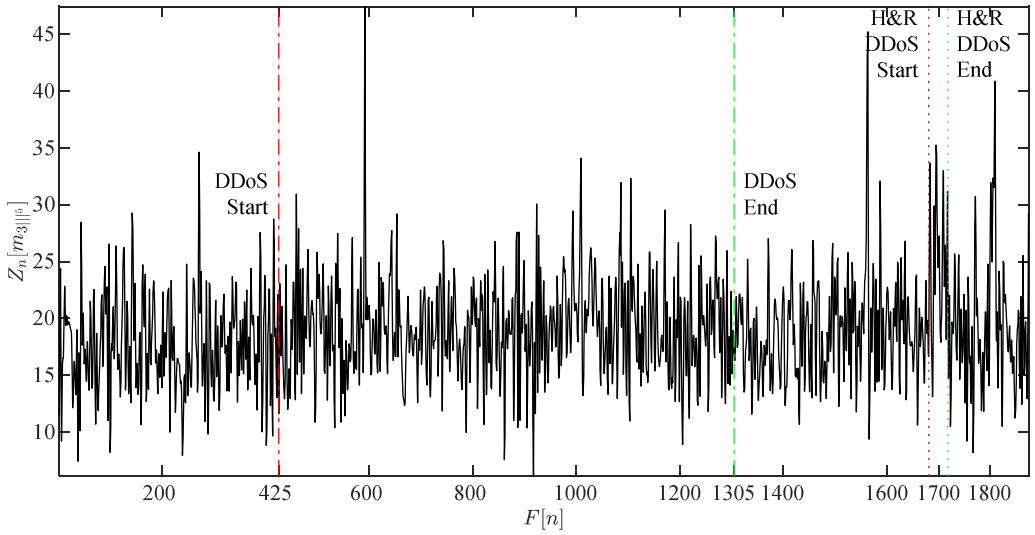


Fig. I.40. ZCR Z_n applied to the skewness multiscalor 5th component $(m_{3\|\|^5})$ after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

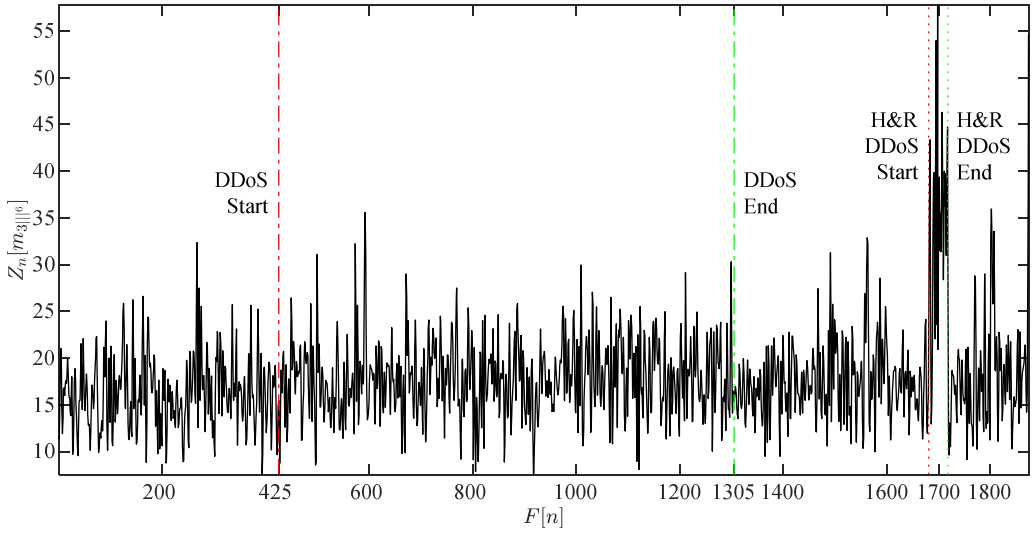


Fig. I.41. ZCR Z_n applied to the skewness multiscalor 6th component $(m_{3\|\|^6})$ after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

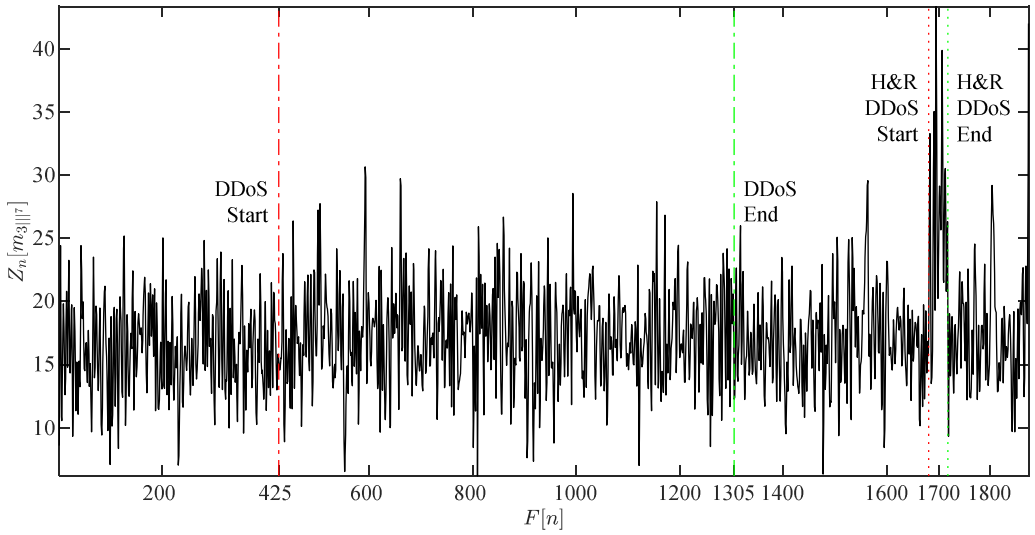


Fig. I.42. ZCR Z_n applied to the skewness multiscalor 7th component ($m_{3|||^7}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

I.2.3 Zero-Crossing Rate Applied to Skewness Multiscalor Components Non-Linearly Filtered After Donoho's Denoising

When nonlinearly filtering the previous results of Donoho's denoising from the ZCR applied to the skewness multiscalor components ($Z_n[m_{3|||^n}]$), the outcomes of different visual perception quality (from first, $m_{3|||^1}$, to seventh, $m_{3|||^7}$), as seen in Figs. I.43 to I.49, achieve their highest reduction of spiky behaviour. Specifically, the DNS amplification DDoS attack emerges as having better quality for the first, $m_{3|||^1}$ shown in Fig. I.43, and a lesser quality for the fourth, $m_{3|||^4}$ shown in Fig. I.46, and not clear contributions for the rest of the components. Related to the H&R DDoS attack case, this exhibits better quality from the fifth, $m_{3|||^5}$ shown in Fig. I.47, to the seventh, $m_{3|||^7}$ shown in Fig. I.49. The shapes of both DDoS attacks, after nonlinearly filtering Donoho's denoising results of the ZCR for the skewness multiscalor components, appear discernible and the specifics for particular cases have been singled out.

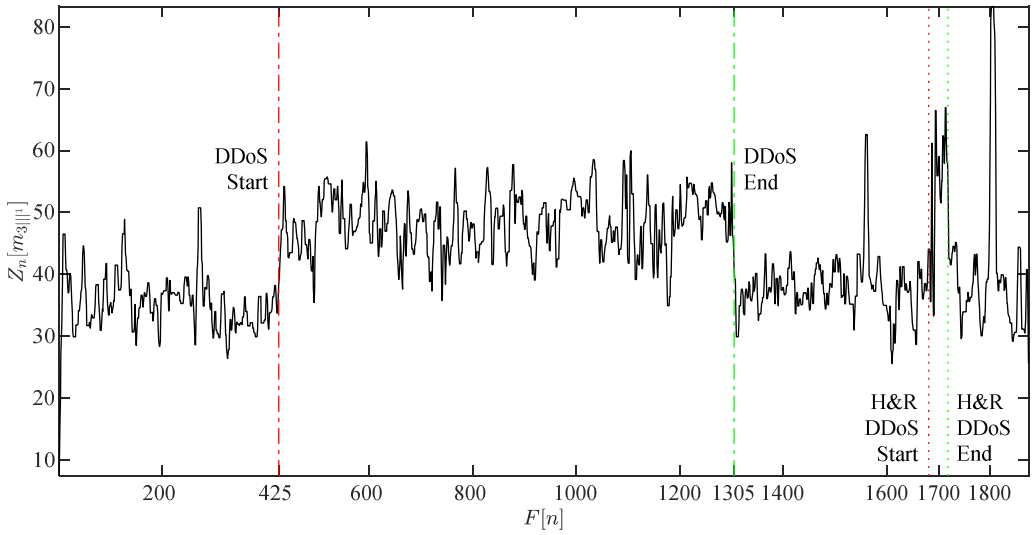


Fig. I.43. ZCR Z_n applied to the skewness multiscalor 1st component ($m_{3\|\|^1}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

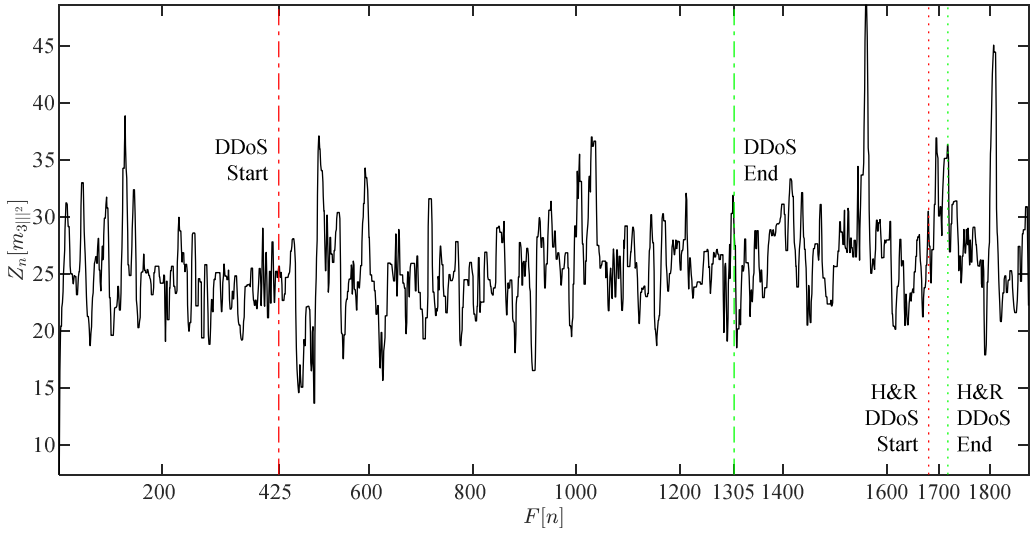


Fig. I.44. ZCR Z_n applied to the skewness multiscalor 2nd component ($m_{3\|\|^2}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

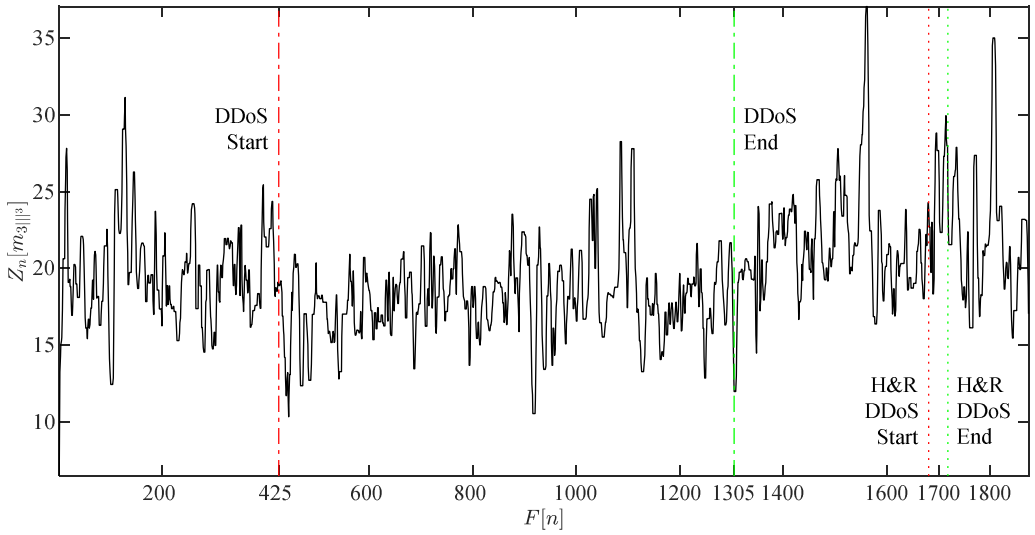


Fig. I.45. ZCR Z_n applied to the skewness multiscalor 3rd component $(m_{3|||^3})$ median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

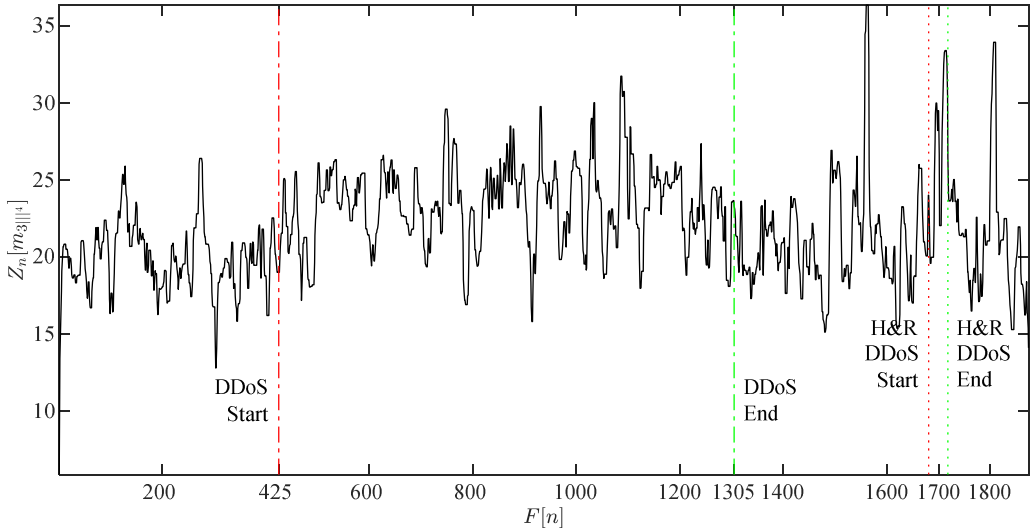


Fig. I.46. ZCR Z_n applied to the skewness multiscalor 4th component $(m_{3|||^4})$ median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

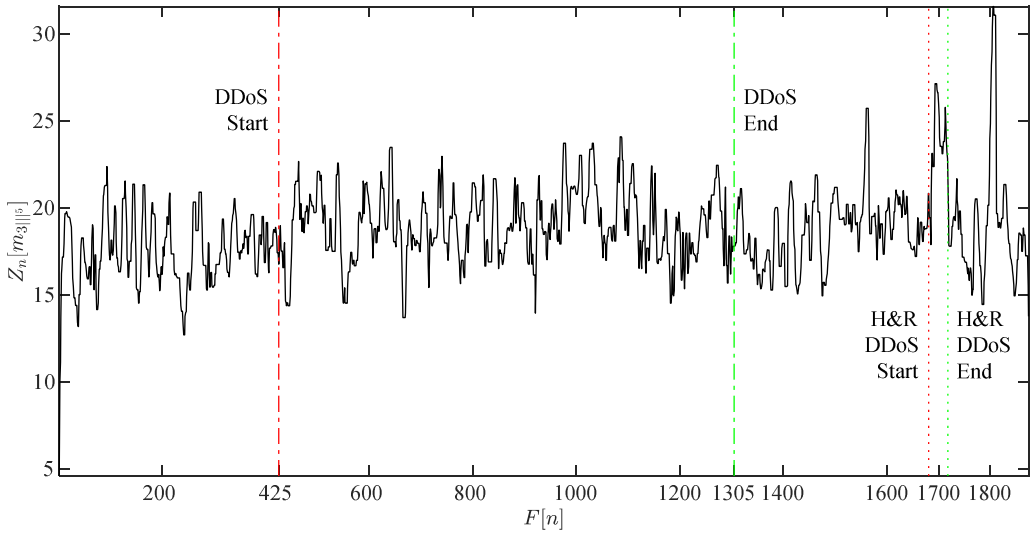


Fig. I.47. ZCR Z_n applied to the skewness multiscalor 5th component $(m_{3\|\|^5})$ median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

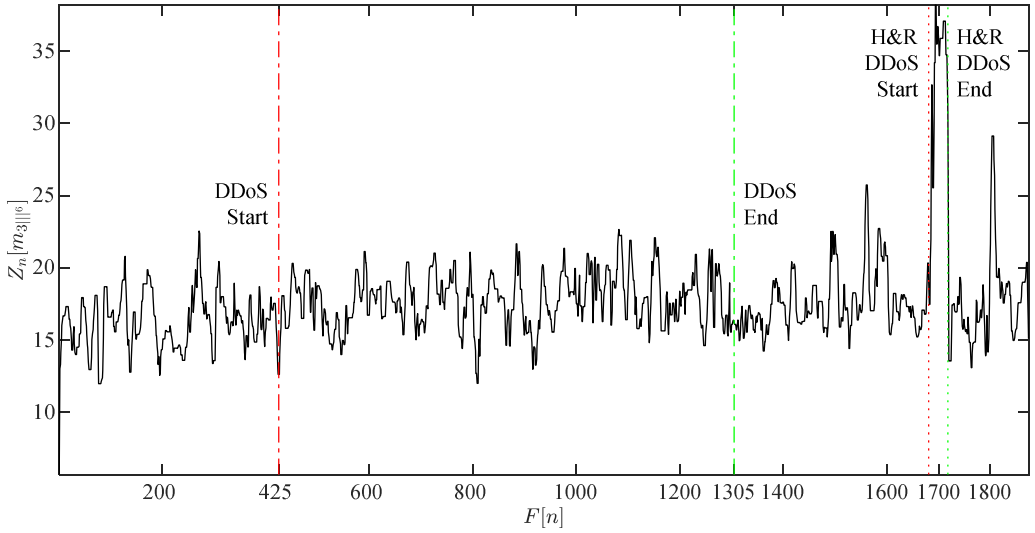


Fig. I.48. ZCR Z_n applied to the skewness multiscalor 6th component $(m_{3\|\|^6})$ median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

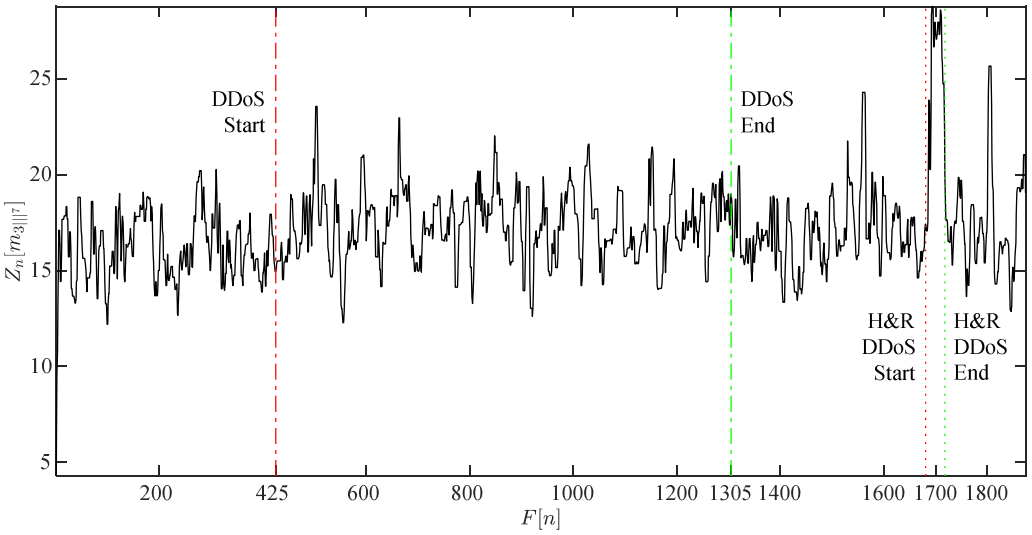


Fig. I.49. ZCR Z_n applied to the skewness multiscalor 7th component $(m_{3|||^7})$ median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

I.2.4 Zero-Crossing Rate Applied to Skewness Multiscalor Components Quantization of Non-Linear Filtering After Donoho's Denoising

Now, quantizing the nonlinear filtering the previous results of Donoho's denoising from the ZCR applied to the skewness multiscalor components $(Z_n[m_{3|||^n}])$, the outcomes of different visual perception quality (from first, $m_{3|||^1}$, to seventh, $m_{3|||^7}$), as seen in Figs. I.50 to I.56, are prepared for further processing by ART. Specifically, the DNS amplification DDoS attack is confirmed to have better quality for the first, $m_{3|||^1}$ shown in Fig. I.50, and a lesser quality for the forth, $m_{3|||^4}$ shown in Fig. I.53, and the fifth (now visibly distinguishable, but not very strong), $m_{3|||^5}$ shown in Fig. I.54, and not clear contributions for the rest of the components. Related to the H&R DDoS attack case, through quantization it can be confirmed that it has better quality for the first, $m_{3|||^1}$ shown in Fig. I.50, the sixth, $m_{3|||^6}$ shown in Fig. I.55, and the seventh, $m_{3|||^7}$ shown in Fig. I.56. The shapes of both DDoS attacks, after quantizing the nonlinear filtering of Donoho's denoising results from the ZCR of the skewness multiscalor components, appears in their clearest form and the specifics for particular cases have been singled out.

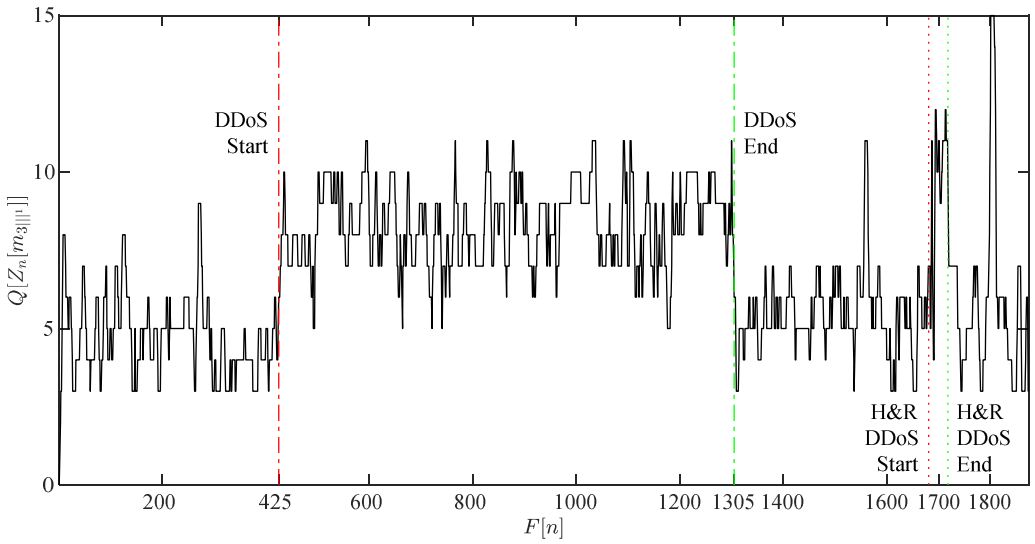


Fig. I.50. ZCR Z_n applied to the skewness multiscalor 1st component $(m_{3||^1})$ quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

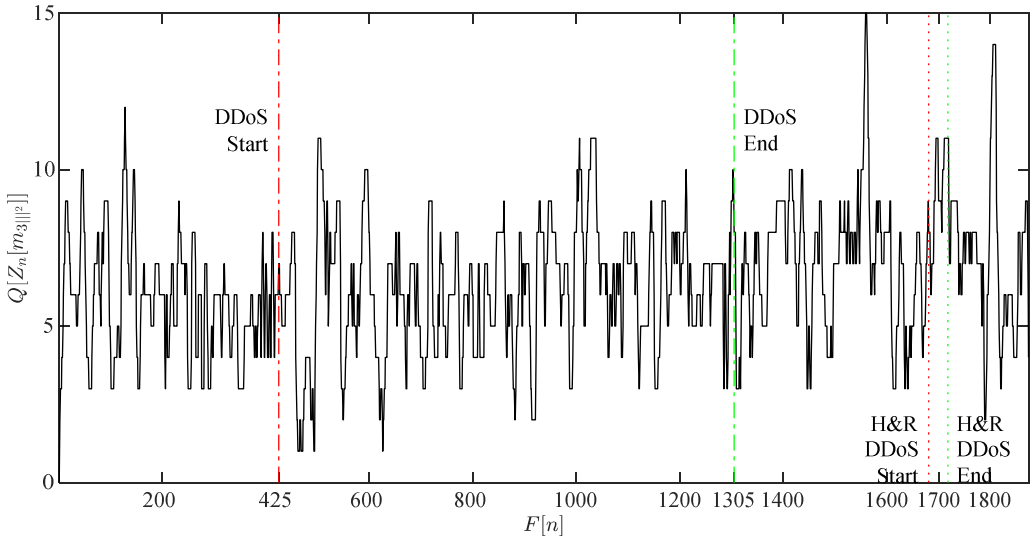


Fig. I.51. ZCR Z_n applied to the skewness multiscalor 2nd component $(m_{3||^2})$ quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

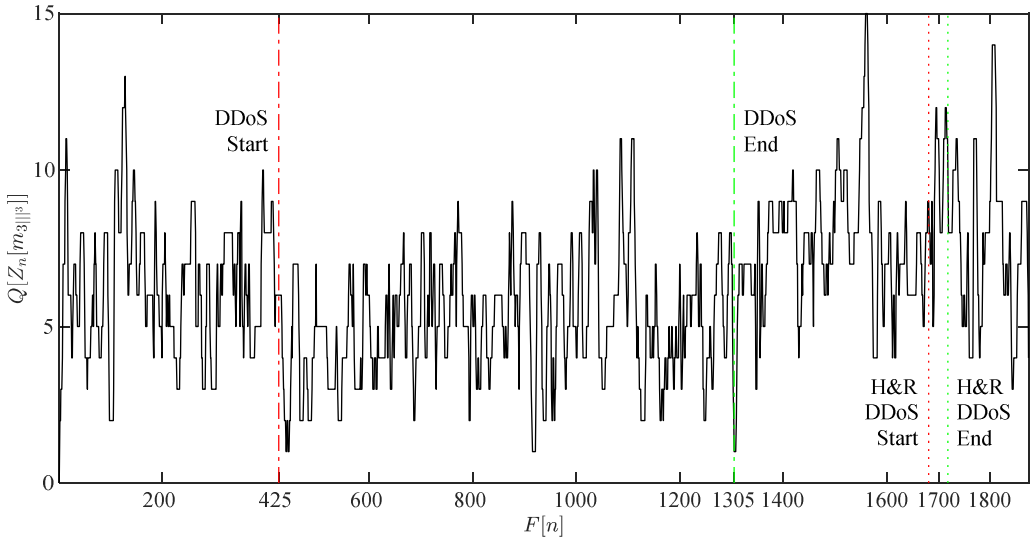


Fig. I.52. ZCR Z_n applied to the skewness multiscalor 3rd component ($m_{3\|\|^3}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

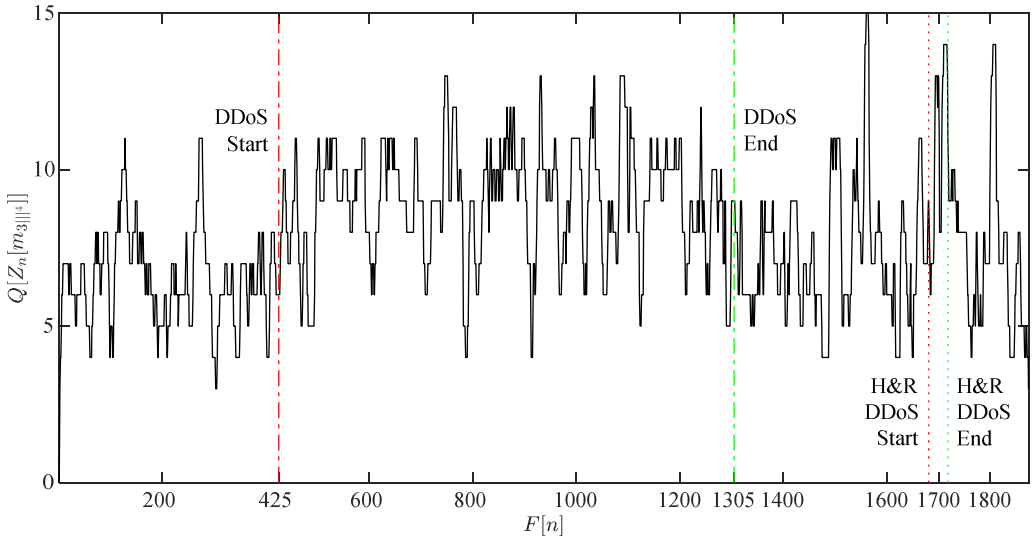


Fig. I.53. ZCR Z_n applied to the skewness multiscalor 4th component ($m_{3\|\|^4}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

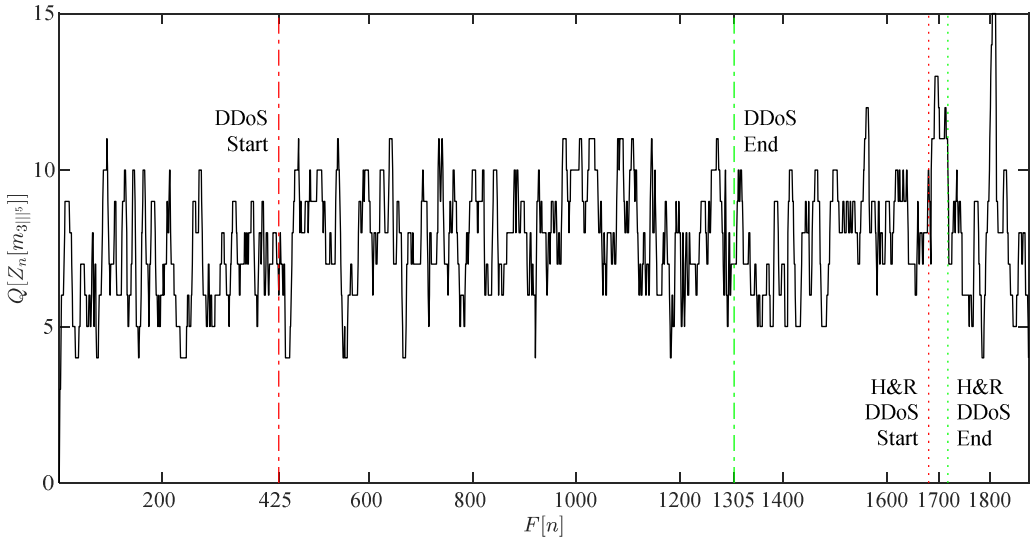


Fig. I.54. ZCR Z_n applied to the skewness multiscalor 5th component ($m_{3||^5}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

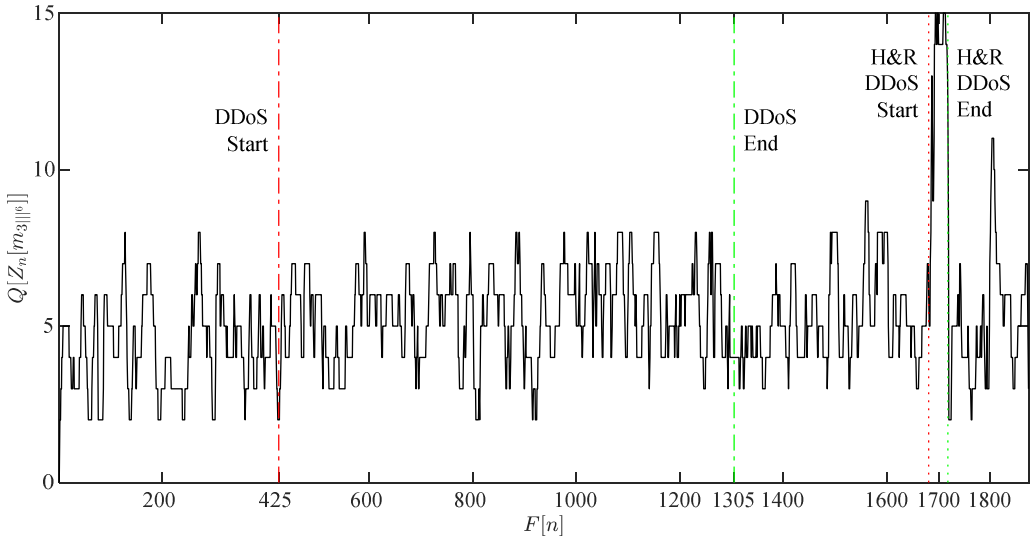


Fig. I.55. ZCR Z_n applied to the skewness multiscalor 6th component ($m_{3||^6}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

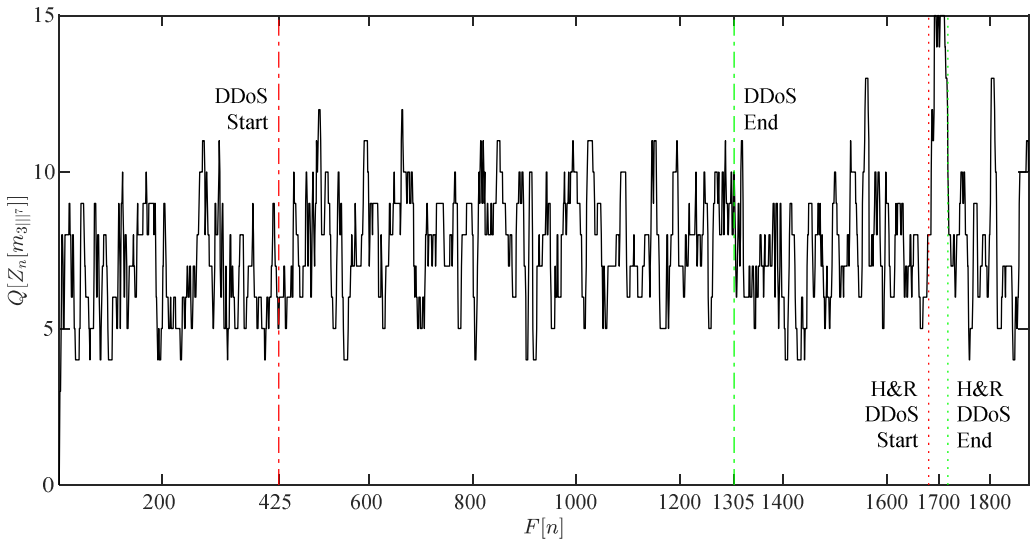


Fig. I.56. ZCR Z_n applied to the skewness multiscalor 7th component $(m_{3|||^7})$ quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

I.3 Shannon's Entropy

I.3.1 Shannon's Entropy Applied to Skewness Multiscalor Components

Applying Shannon's entropy to the skewness multiscalor component $(H[m_{3|||^n}])$, as seen in Figs. I.57 to I.63 representing the first, $m_{3|||^1}$, to seventh, $m_{3|||^7}$. From analysing the skewness multiscalor components with Shannon's entropy, one can see that the dynamics of the DNS DDoS attack are noticed in Fig. I.57 (first skewness multiscalor, $m_{3|||^1}$), while the dynamics of the H&R DDoS attack are noticed in Figs. I.57 (first skewness multiscalor, $m_{3|||^1}$) and from Figs. I.60 to I.63 (from fourth to seventh skewness multiscalors, $m_{3|||^4}$ to $m_{3|||^7}$).

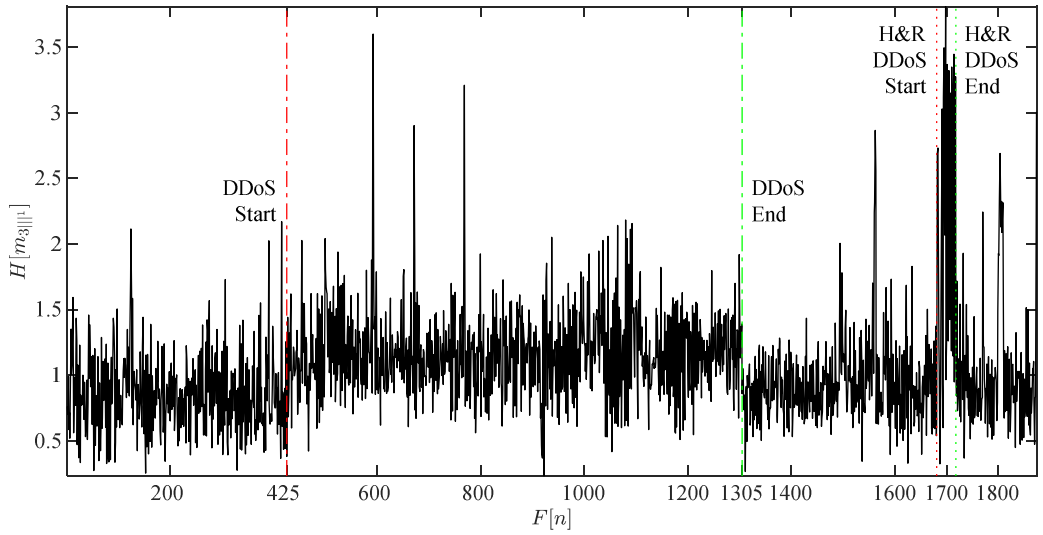


Fig. I.57. Shannon's entropy *H* applied to the skewness multiscalor 1st component ($m_{3\|\|^1}$). A processing *frame* of 256 samples is used.

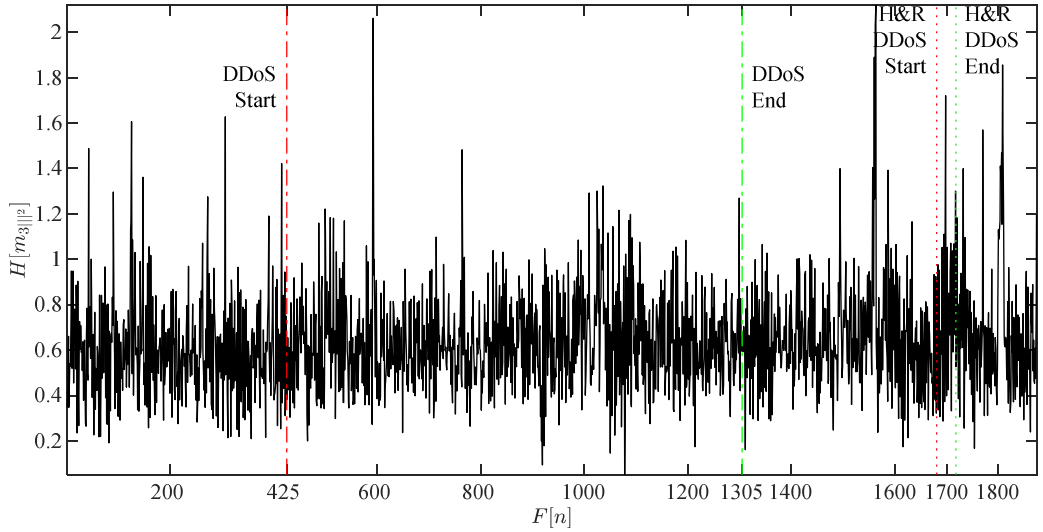


Fig. I.58. Shannon's entropy *H* applied to the skewness multiscalor 2nd component ($m_{3\|\|^2}$). A processing *frame* of 256 samples is used.

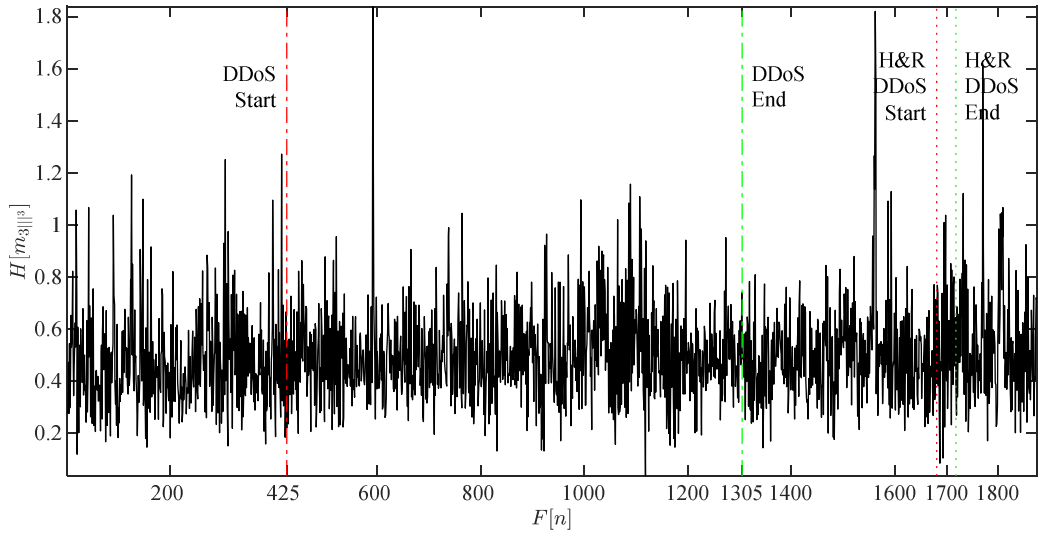


Fig. I.59. Shannon's entropy *H* applied to the skewness multiscalor 3rd component $(m_{3\|\|^3})$. A processing *frame* of 256 samples is used.

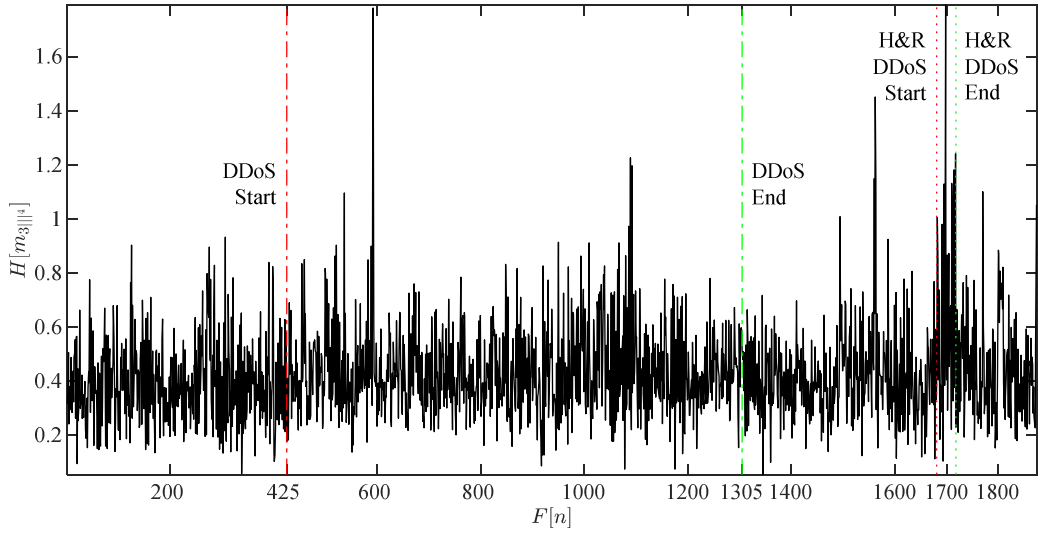


Fig. I.60. Shannon's entropy *H* applied to the skewness multiscalor 4th component $(m_{3\|\|^4})$. A processing *frame* of 256 samples is used.

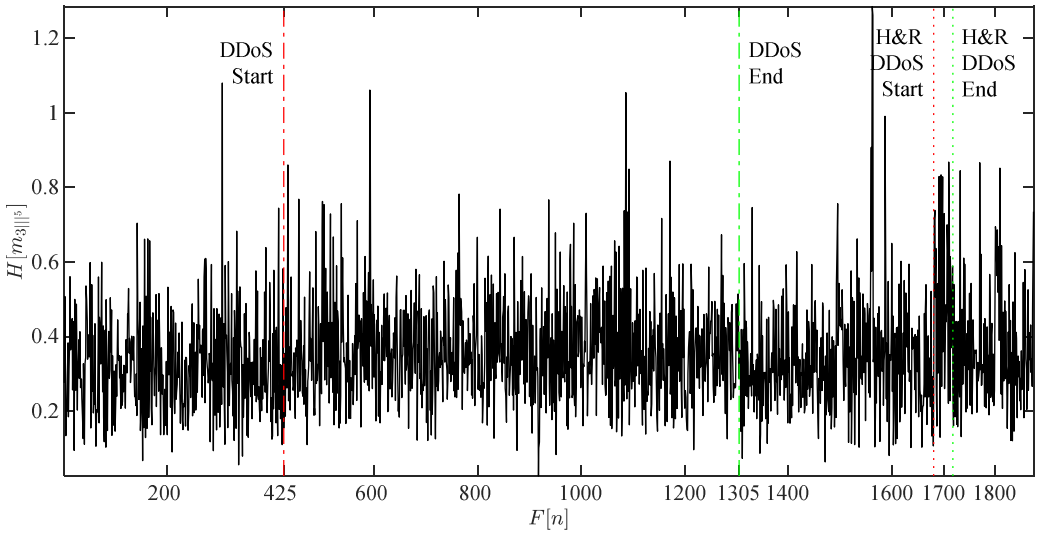


Fig. I.61. Shannon's entropy *H* applied to the skewness multiscalor 5th component $(m_{3||^5})$. A processing *frame* of 256 samples is used.

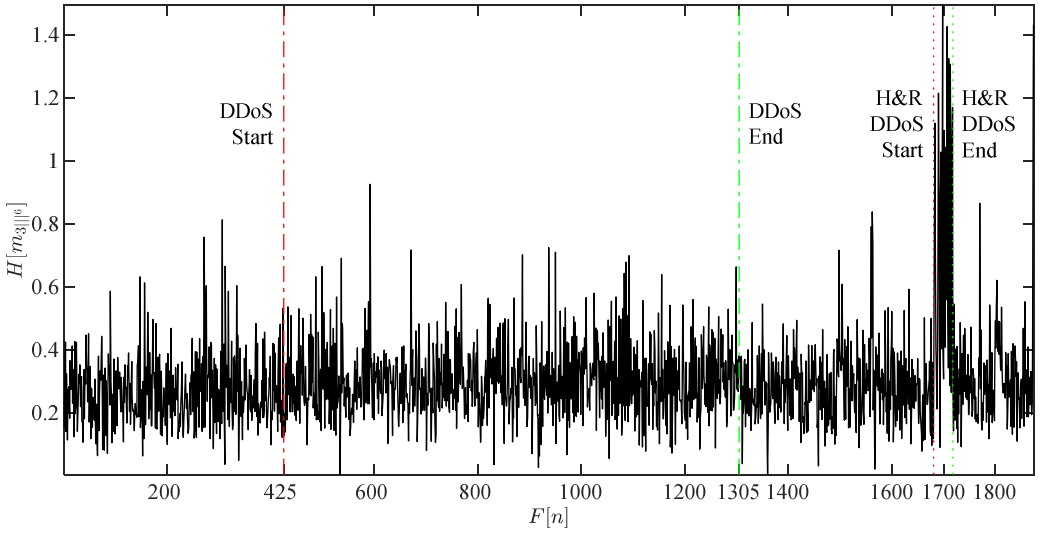


Fig. I.62. Shannon's entropy *H* applied to the skewness multiscalor 6th component $(m_{3||^6})$. A processing *frame* of 256 samples is used.

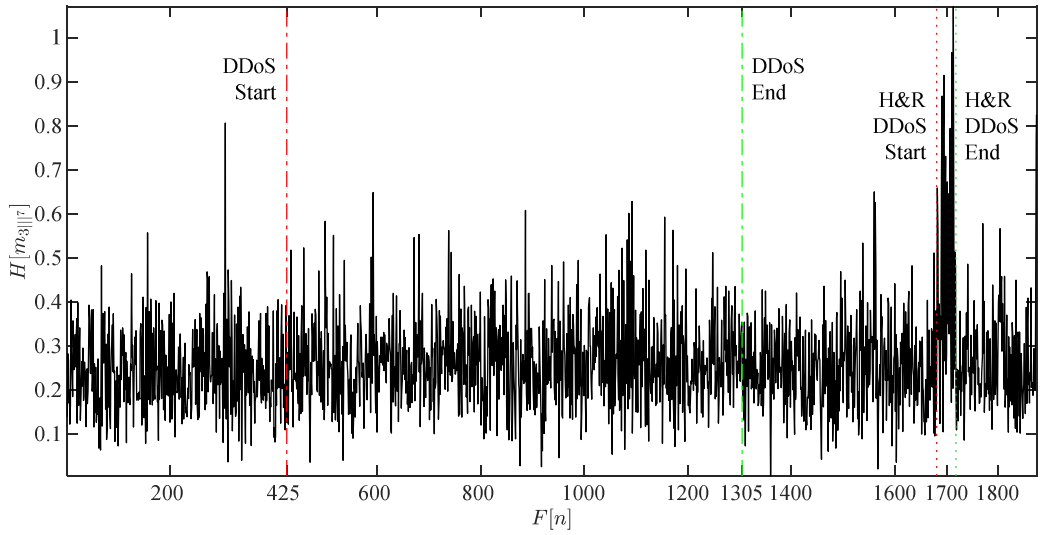


Fig. I.63. Shannon's entropy *H* applied to the skewness multiscalor 7th component ($m_{3|||^7}$). A processing *frame* of 256 samples is used.

I.3.2 Shannon's Entropy Applied to Skewness Multiscalor Components After Donoho's Denoising

Donoho's denoising smooths the results obtained from Shannon's entropy on the skewness multiscalor component ($H[m_{3|||^n}]$), as seen in Figs. I.64 to I.70 representing the first, $m_{3|||^1}$, to seventh, $m_{3|||^7}$. Donoho's denoised skewness multiscalor components analyzed with Shannon's entropy shows a more defined shape for the DNS DDoS attack is noticed in Fig. I.64 (first skewness multiscalor, $m_{3|||^1}$). Also, a clearer shape for the H&R DDoS attack is noticed in Figs. I.64 (first skewness multiscalor, $m_{3|||^1}$) and from Figs. I.67 to I.70 (from fourth to seventh skewness multiscalors, $m_{3|||^4}$ to $m_{3|||^7}$).

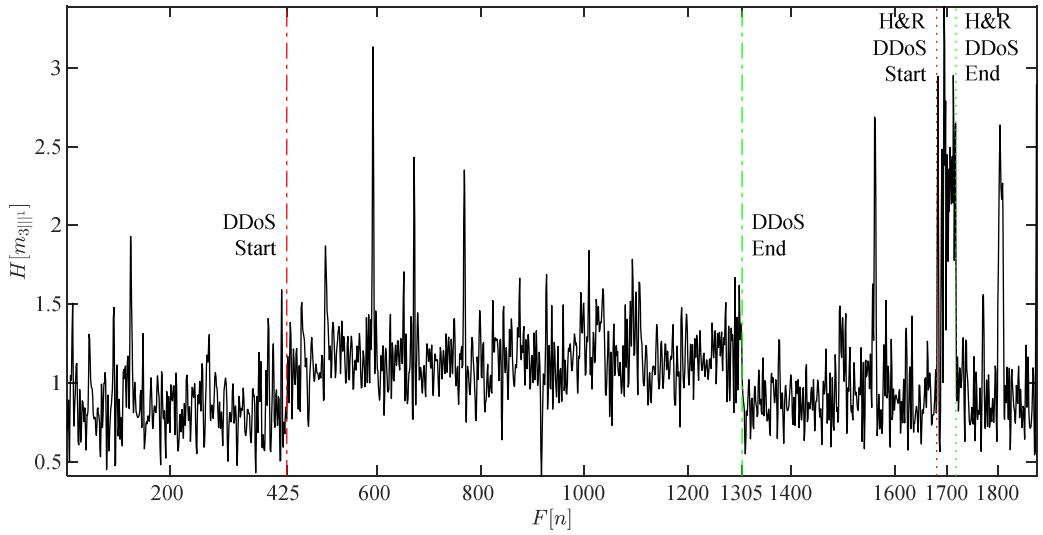


Fig. I.64. Shannon's entropy *H* applied to the skewness multiscalor 1st component ($m_{3\|1^1}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

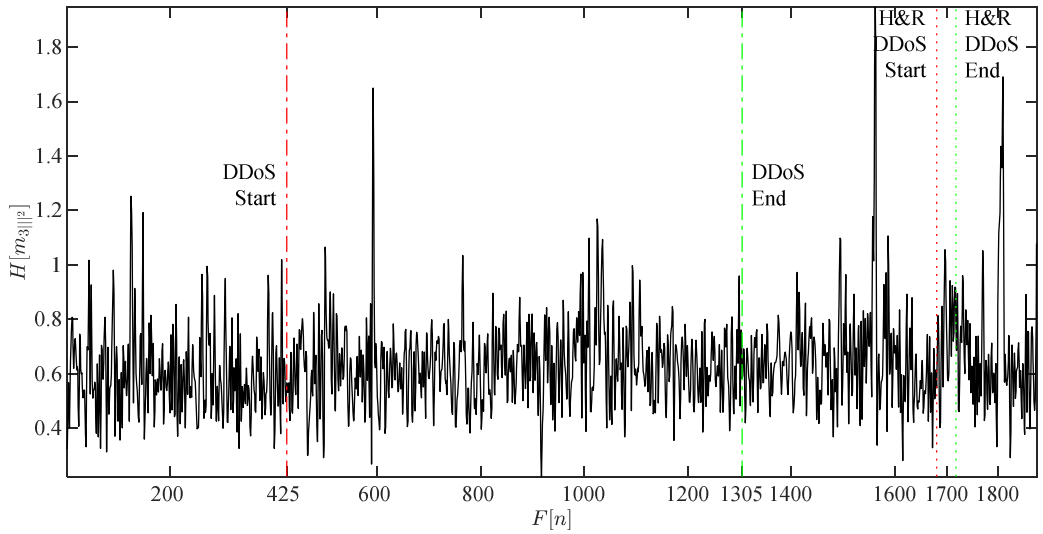


Fig. I.65. Shannon's entropy *H* applied to the skewness multiscalor 2nd component ($m_{3\|1^2}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

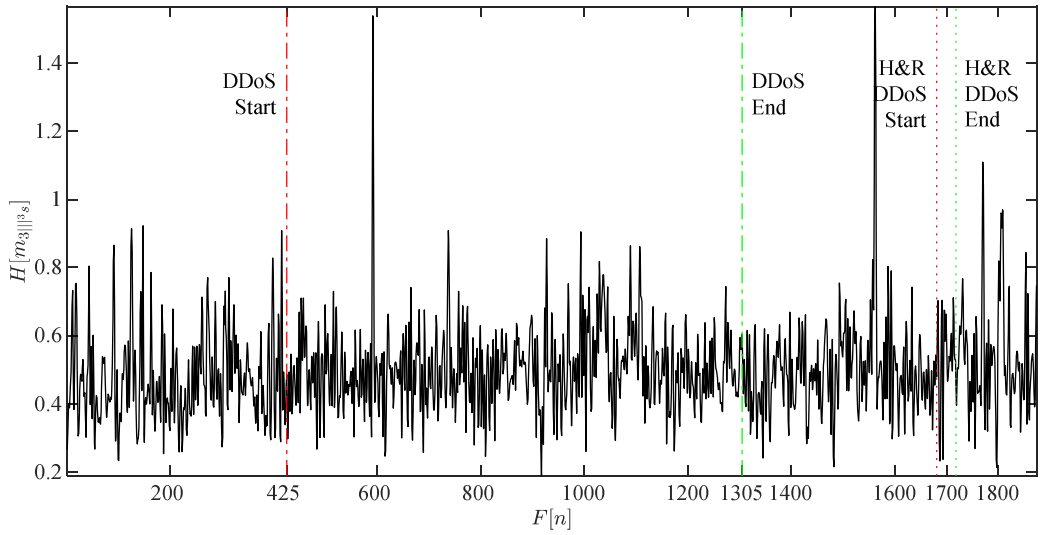


Fig. I.66. Shannon's entropy *H* applied to the skewness multiscalor 3rd component ($m_{3|||^3}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

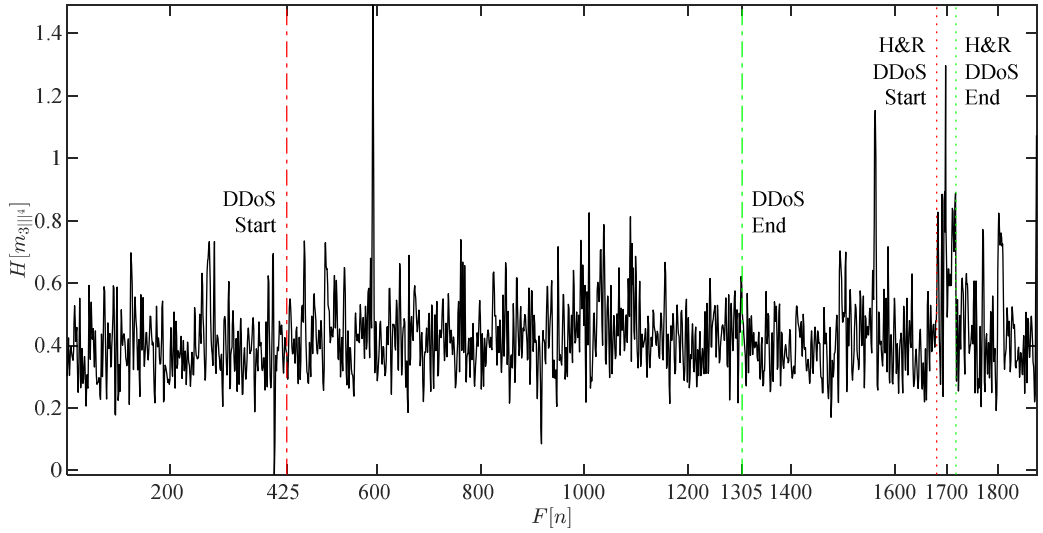


Fig. I.67. Shannon's entropy *H* applied to the skewness multiscalor 4th component ($m_{3|||^4}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

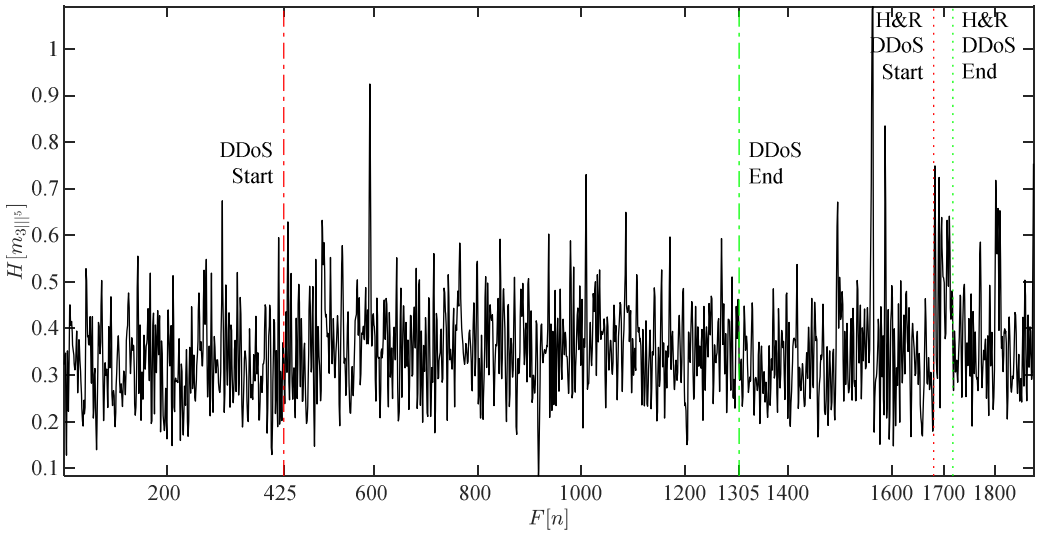


Fig. I.68. Shannon's entropy *H* applied to the skewness multiscalor 5th component ($m_{3\|\|^5}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

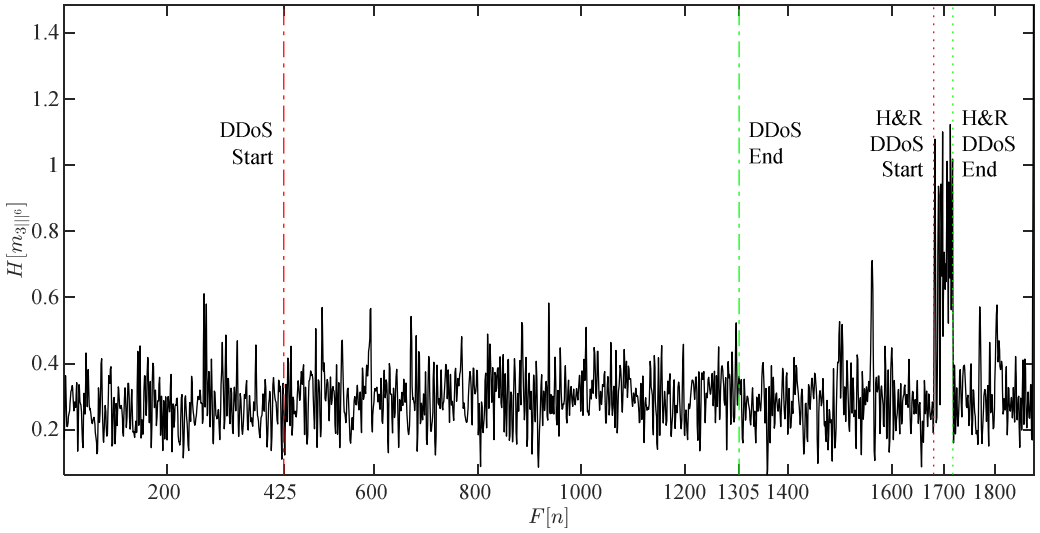


Fig. I.69. Shannon's entropy *H* applied to the skewness multiscalor 6th component ($m_{3\|\|^6}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

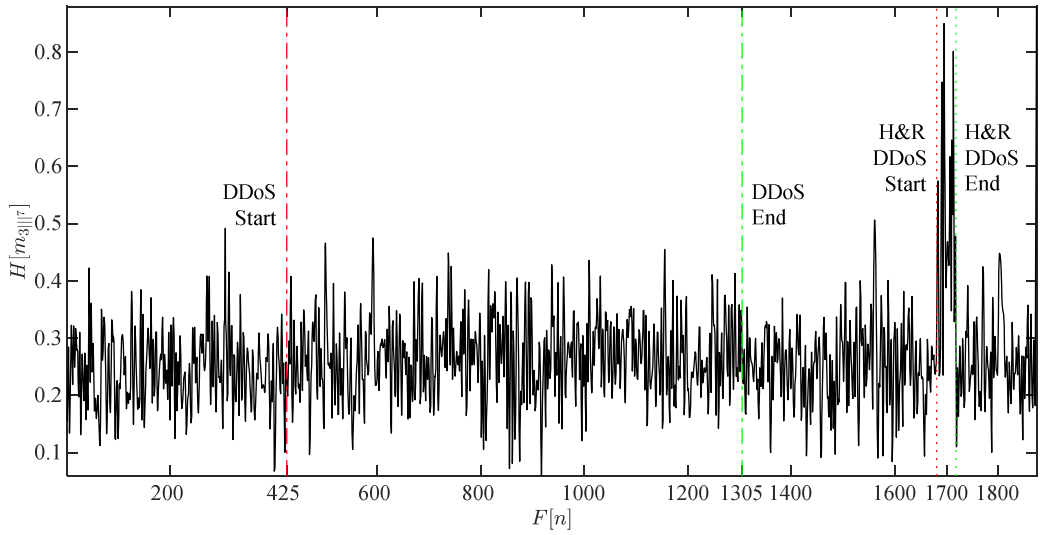


Fig. I.70. Shannon's entropy *H* applied to the skewness multiscalor 7th component ($m_{3|||^7}$) after Donoho's denoising. A Coiflet wavelet with scaling factor of five is used. The DDoS attack dynamics are clearly seen. Also, the hit and run DDoS attack is seen.

I.3.3 Shannon's Entropy Applied to Skewness Multiscalor Components Non-Linearly Filtered After Donoho's Denoising

Chaining nonlinear filtering to Donoho's denoising incorporates a second smoothing pass to the results obtained from Shannon's entropy on the skewness multiscalor component ($H[m_{3|||^n}]$), as seen in Figs. I.71 to I.77 representing the first, $m_{3|||^1}$, to seventh, $m_{3|||^7}$. Nonlinearly filtering the Donoho's denoised skewness multiscalor components analyzed with Shannon's entropy confirms a clearer shape for the DNS DDoS attack is noticed in Fig. I.71 (first skewness multiscalor, $m_{3|||^1}$). Also, a clearer shape for the H&R DDoS attack is confirmed in Figs. I.71 (first skewness multiscalor, $m_{3|||^1}$) and from Figs. I.74 to I.77 (from fourth to seventh skewness multiscalors, $m_{3|||^4}$ to $m_{3|||^7}$).

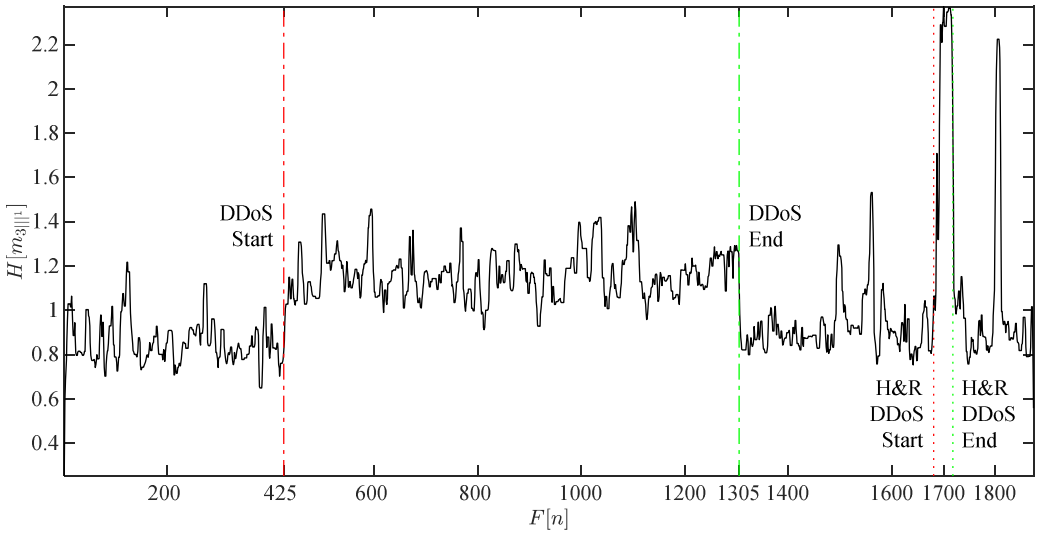


Fig. I.71. Shannon's entropy *H* applied to the skewness multiscalor 1st component ($m_{3\|\|^1}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

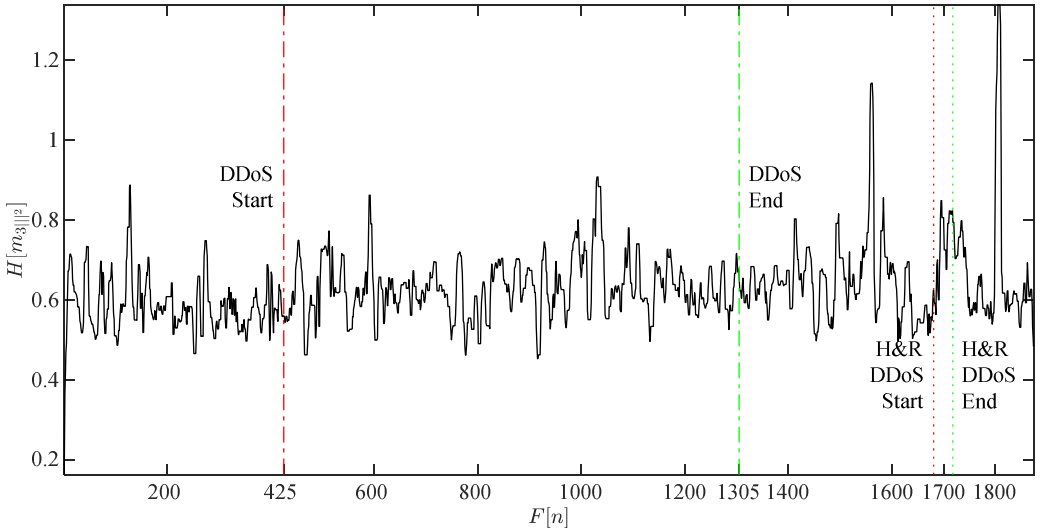


Fig. I.72. Shannon's entropy *H* applied to the skewness multiscalor 2nd component ($m_{3\|\|^2}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

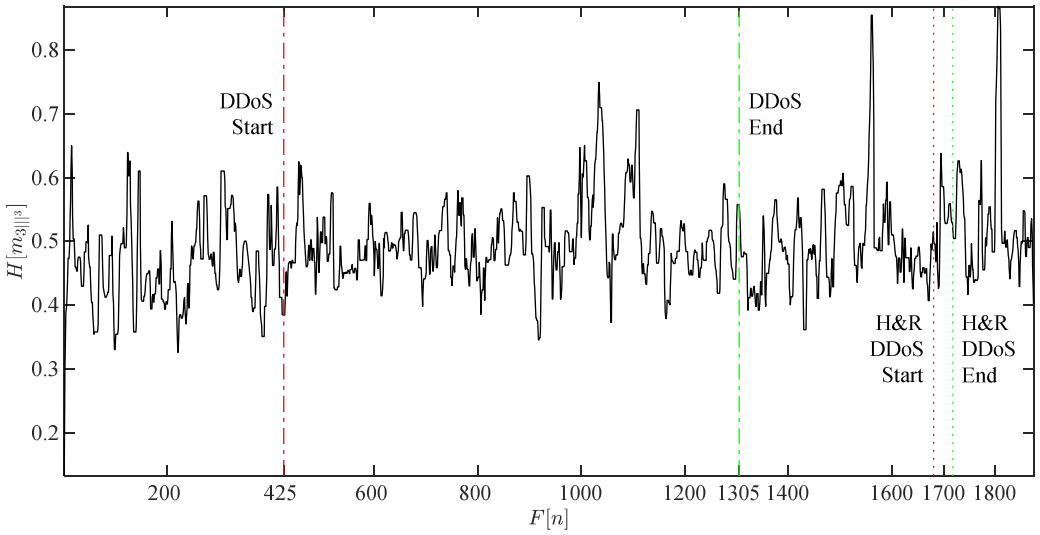


Fig. I.73. Shannon's entropy *H* applied to the skewness multiscalor 3rd component ($m_{3||^3}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

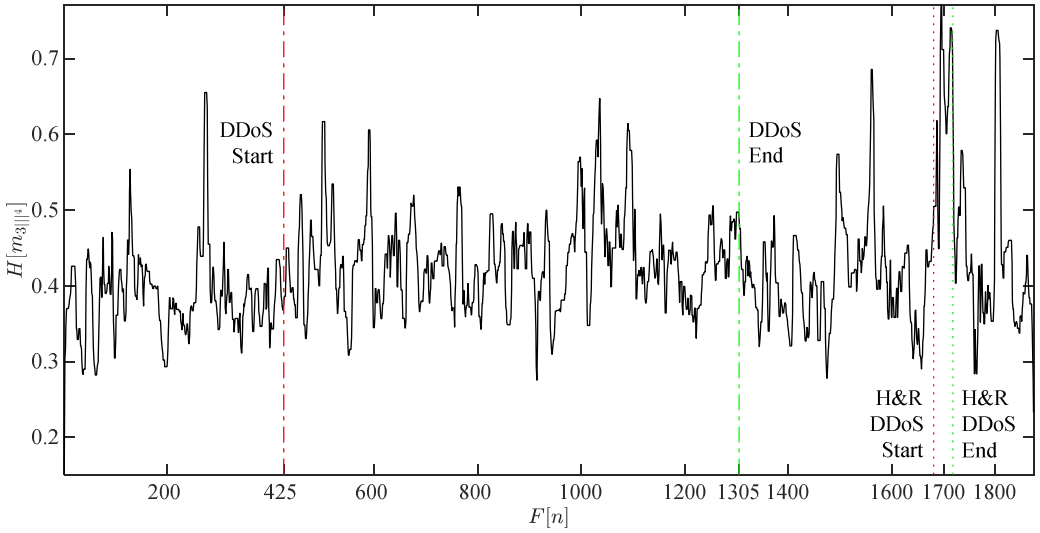


Fig. I.74. Shannon's entropy *H* applied to the skewness multiscalor 4th component ($m_{3||^4}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

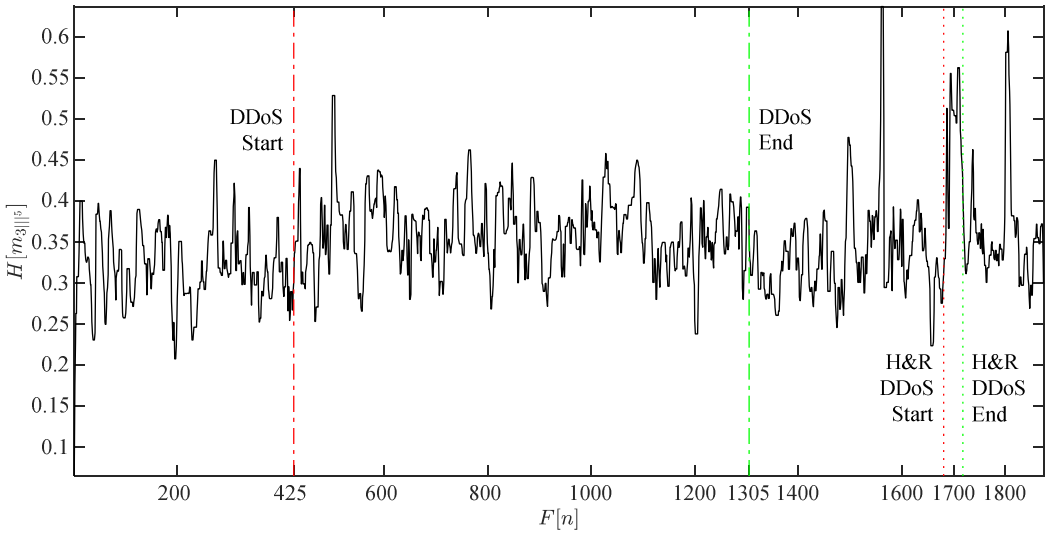


Fig. I.75. Shannon's entropy *H* applied to the skewness multiscalor 5th component ($m_{3\|\|^5}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

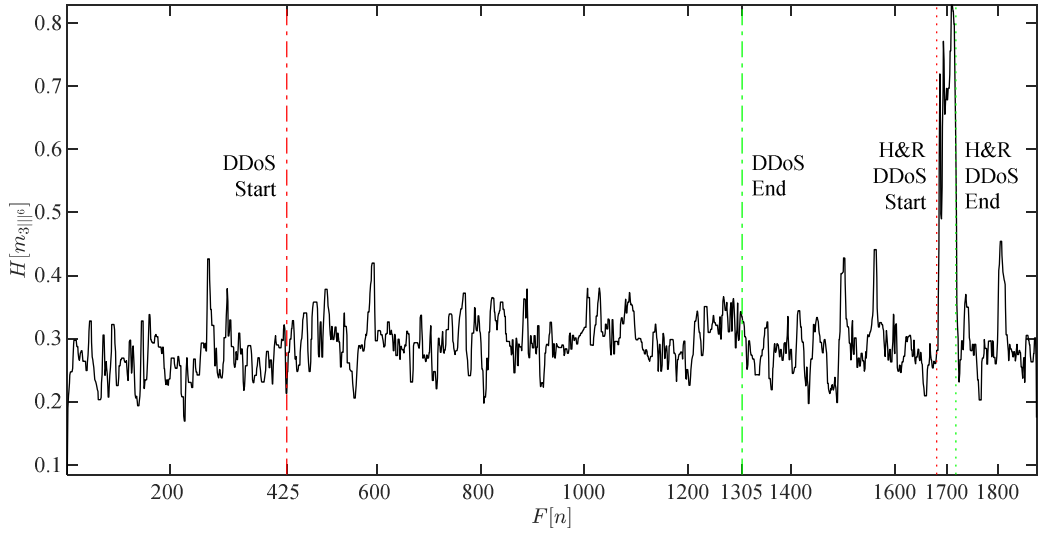


Fig. I.76. Shannon's entropy *H* applied to the skewness multiscalor 6th component ($m_{3\|\|^6}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

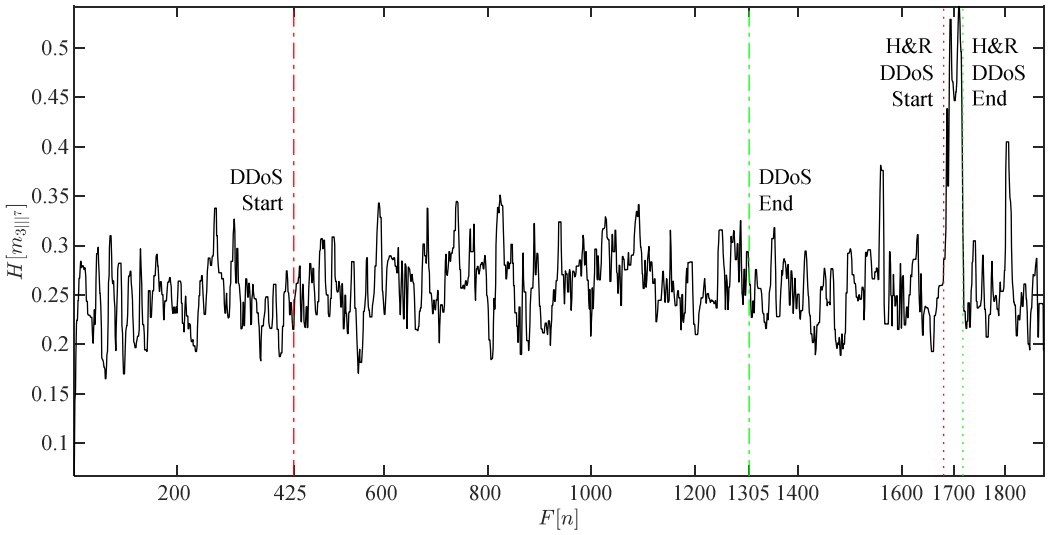


Fig. I.77. Shannon's entropy *H* applied to the skewness multiscalor 7th component ($m_{3\|\|^7}$) median filtering once denoised with Donoho's methodology. The DDoS attack dynamics are seen. Also, a hit and run DDoS attack is seen.

I.3.4 Shannon's Entropy Applied to Skewness Multiscalor Components Quantization of Non-Linear Filtering After Donoho's Denoising

Quantizing the nonlinear filtering of Donoho's denoising from the Shannon's entropy on the skewness multiscalor component ($H[m_{3\|\|^n}]$), as seen in Figs. I.78 to I.84 representing the first, $m_{3\|\|^1}$, to seventh, $m_{3\|\|^7}$, prepares the metrics for further processing by ART. The quantization of nonlinearly filtering the Donoho's denoised skewness multiscalor components analyzed with Shannon's entropy depicts the DNS DDoS attack in its pure form achievable by the data processing chain utilized in this research as seen in Fig. I.78 (first skewness multiscalor, $m_{3\|\|^1}$). Also, the shapes for the H&R DDoS attack are depicted in their purest form achievable here in Figs. I.78 (first skewness multiscalor, $m_{3\|\|^1}$) and from Figs. I.81 to I.84 (from fourth to seventh skewness multiscalors, $m_{3\|\|^4}$ to $m_{3\|\|^7}$).

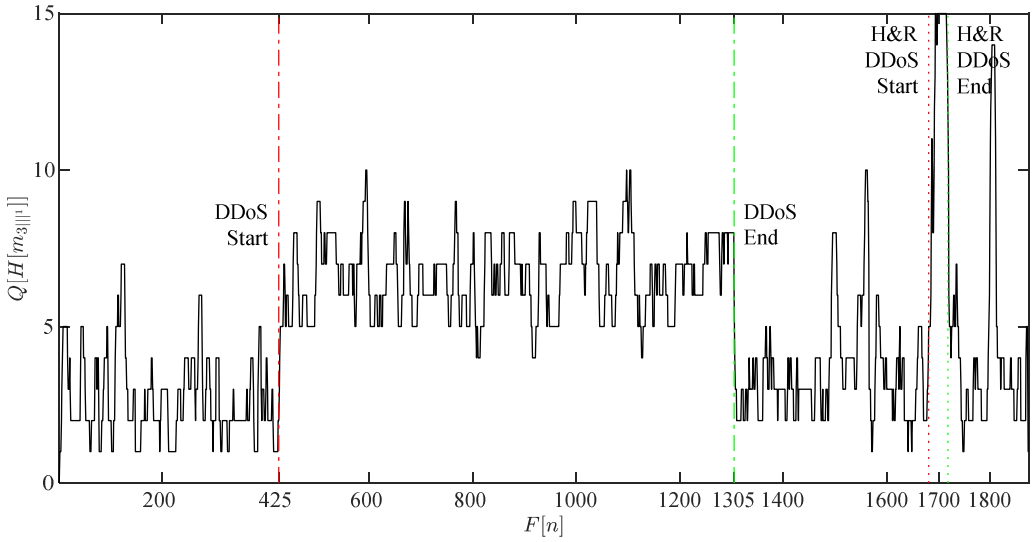


Fig. I.78. Shannon's entropy *H* applied to the skewness multiscalor 1st component ($m_{3\|\|^1}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

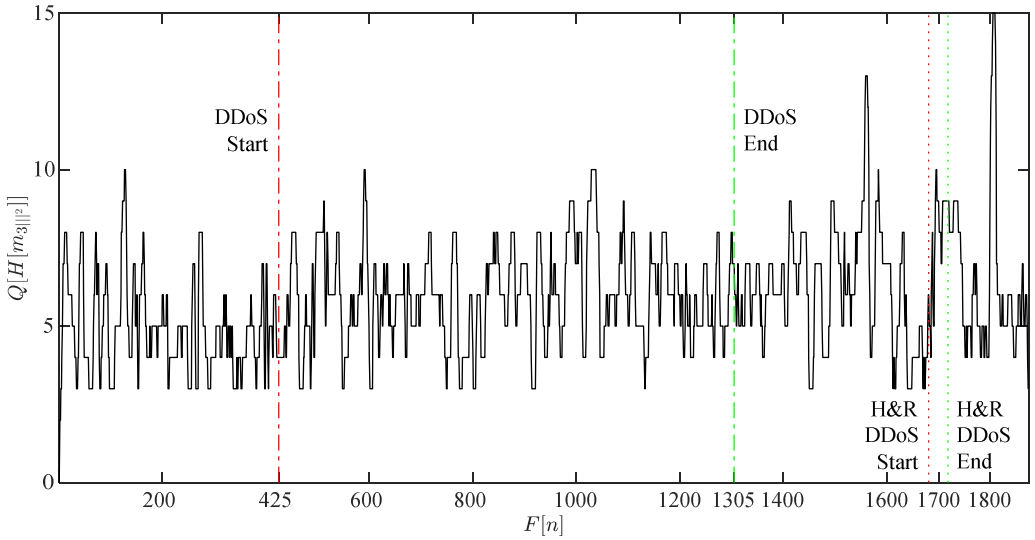


Fig. I.79. Shannon's entropy *H* applied to the skewness multiscalor 2nd component ($m_{3\|\|^2}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

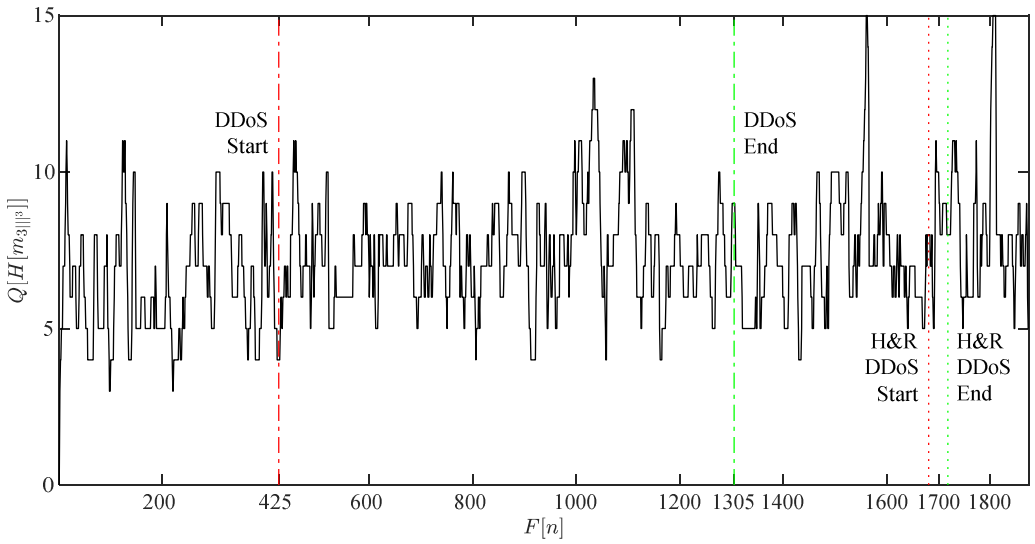


Fig. I.80. Shannon's entropy *H* applied to the skewness multiscalor 3rd component ($m_{3\|\|^3}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

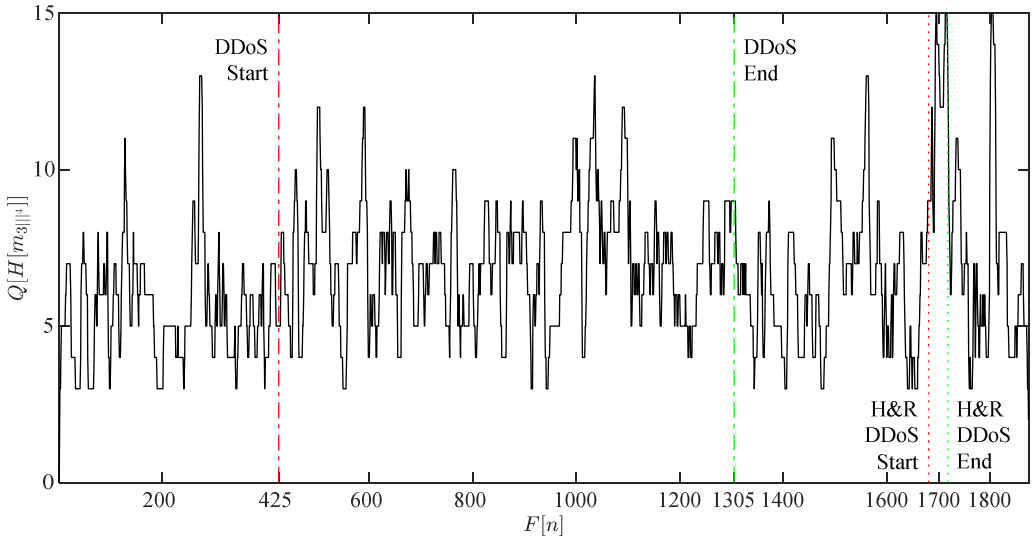


Fig. I.81. Shannon's entropy *H* applied to the skewness multiscalor 4th component ($m_{3\|\|^4}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

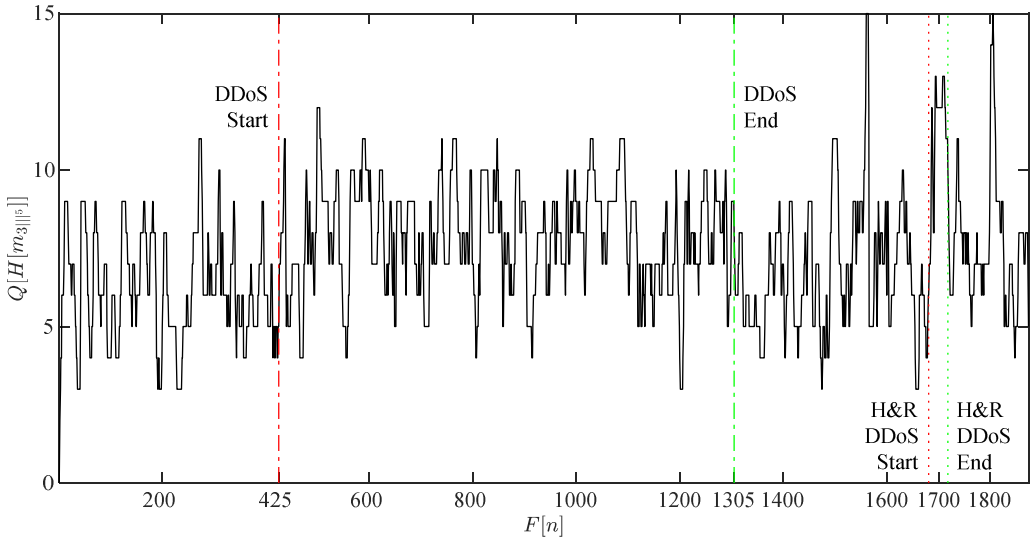


Fig. I.82. Shannon's entropy *H* applied to the skewness multiscalor 5th component $(m_{3\|\|^5})$ quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

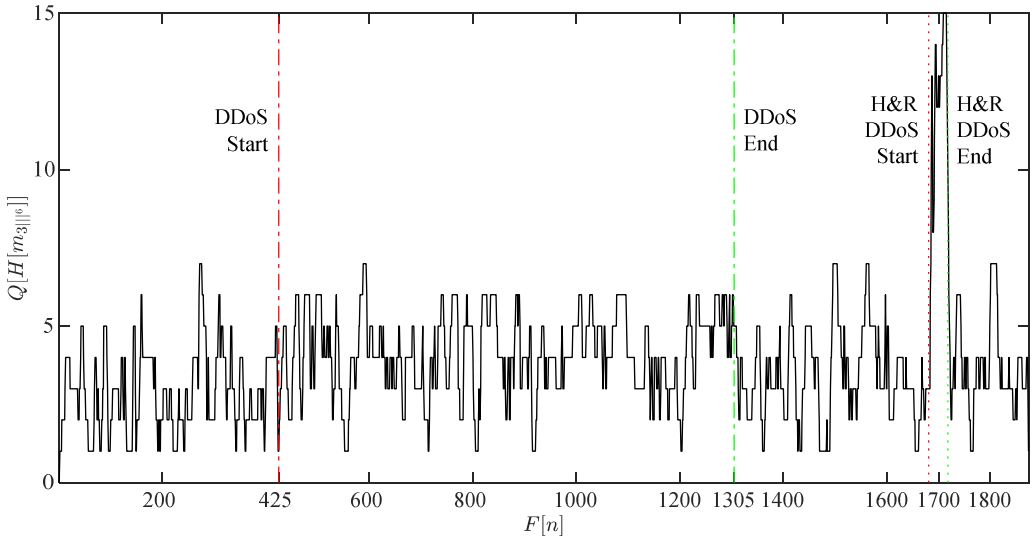


Fig. I.83. Shannon's entropy *H* applied to the skewness multiscalor 6th component $(m_{3\|\|^6})$ quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

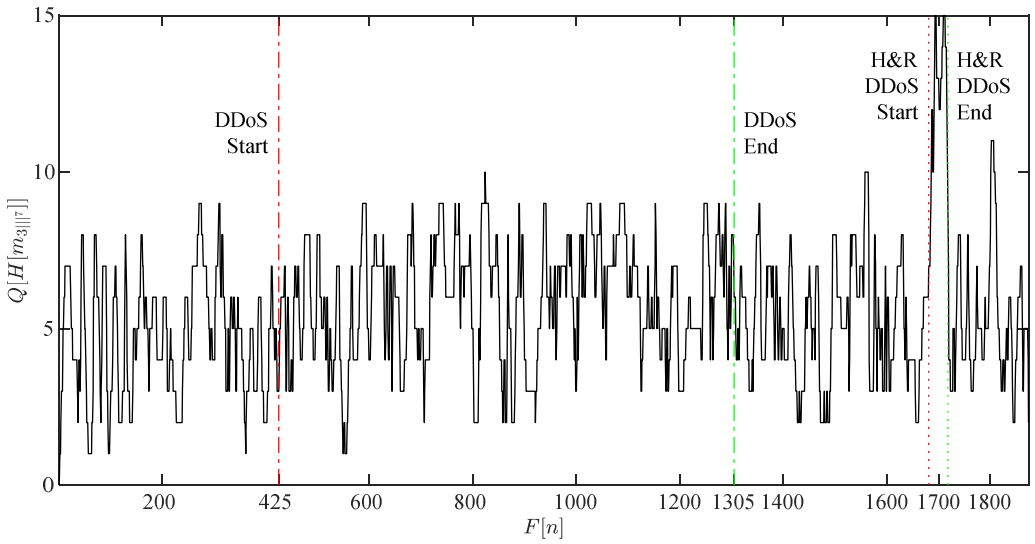


Fig. I.84. Shannon's entropy *H* applied to the skewness multiscalor 7th component ($m_{3|||^7}$) quantized with Lloyd's methodology. The DDoS attack dynamics are clearly seen. Also, a hit and run DDoS attack is seen.

APPENDIX J

CONFUSION MATRICES FOR ART1 PERFORMANCE

The following observations are worth to highlight from the confusion matrix obtained for the classification through ART1 with a $\rho = 0.07$: (i) It presents high values for both precision (97.5%) and recall (98.0%) providing a high degree of confidence when it comes to properly identify the presence of a DDoS attack, (ii) similarly, it shows high values for both precision (98.3%) and recall (97.8%) providing a high degree of confidence when it comes to properly identify the clear traffic, and (iii) a low number of misclassifications is found.

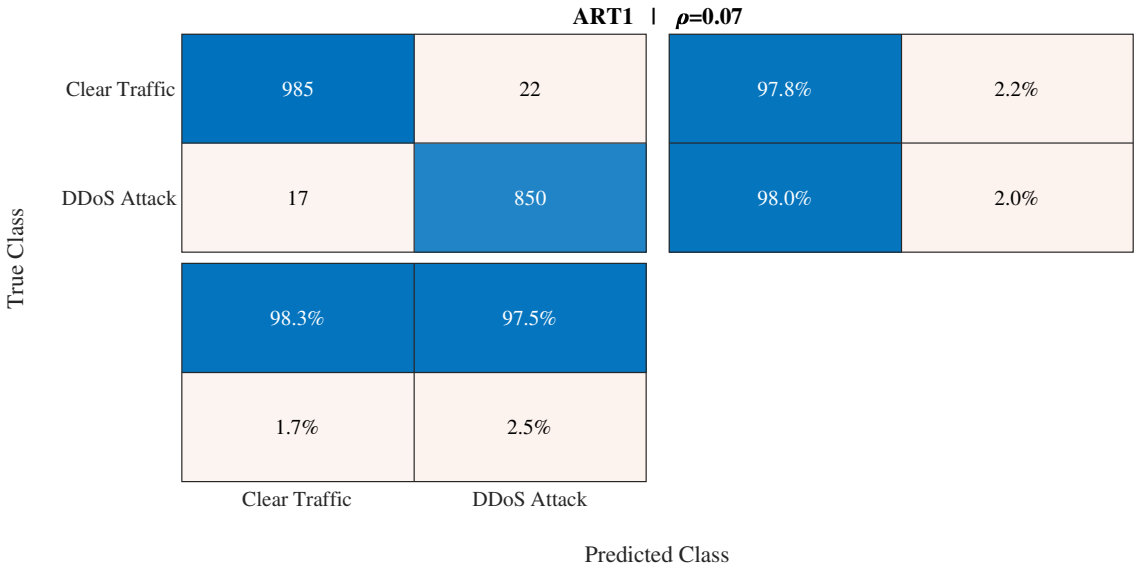


Fig. J.1. Confusion matrix for ART1 with vigilance parameter $\rho = 0.07$. The matrix displays: (i) 985 cases for clear traffic, (ii) 850 cases for a DDoS attack, (iii) 22 false cases for a DDoS attack, and (iv) 17 false cases for clear traffic. The column normalization (precision): (i) 98.3% for clear traffic, and (ii) 97.5% for a DDoS attack. The row normalization (recall): (i)97.8 % for clear traffic, and (ii) 98% for DDoS attack.

For the confusion matrix acquired for the classification through ART1 with a $\rho = 0.88$, the subsequent observations are meaningful for sharing: (i) High values for both precision (98.5%) and recall (97.8%) are present, which provides a high degree of confidence for the proper identification of the presence of a DDoS attack, (ii) similarly, high values for both precision (98.1%) and recall (98.7%) are shown, also providing a high degree of confidence

when it comes to properly identify the clear traffic, and (iii) low number of misclassifications, 13 and 19 that should belong to the clear traffic and to the DDoS attack respectively, are found.

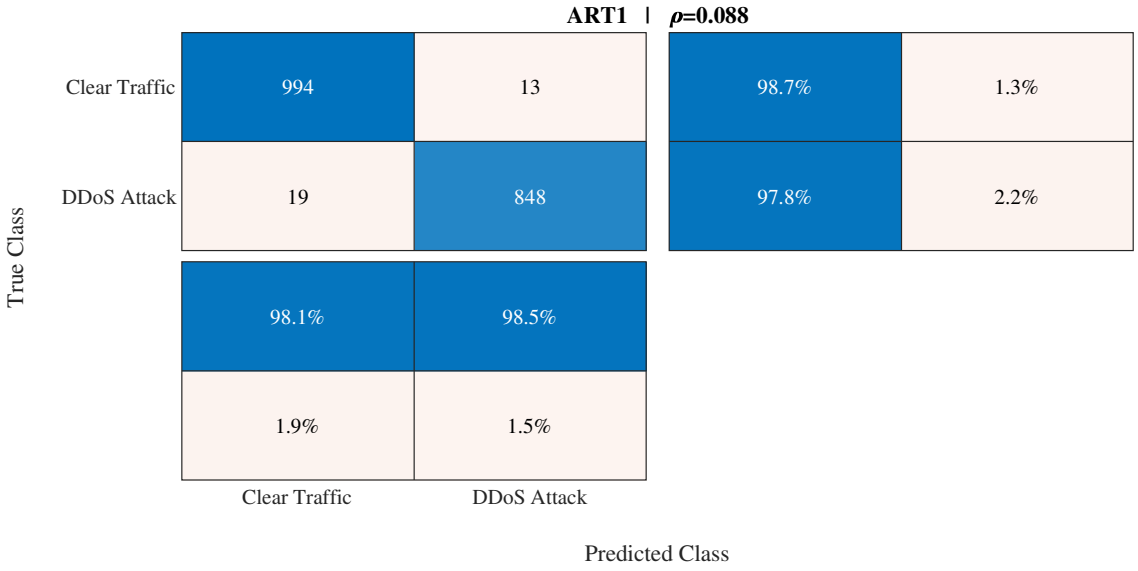


Fig. J.2. Confusion matrix for ART1 with vigilance parameter $\rho = 0.088$. The matrix displays: (i) 994 cases for clear traffic, (ii) 848 cases for a DDoS attack, (iii) 13 false cases for a DDoS attack, and (iv) 19 false cases for clear traffic. The column normalization (precision): (i) 98.1% for clear traffic, and (ii) 98.5% for a DDoS attack. The row normalization (recall): (i) 98.7% for clear traffic, and (ii) 97.8% for DDoS attack.

Tuning the implementation of ART1 to a vigilance parameter $\rho = 0.09$ allows achieving a confusion matrix with the next contemplations: (i) The high values for both precision (98.5%) and recall (97.8%) propose a high degree of confidence for determining the presence of a DDoS attack, (ii) correspondingly, the high values for both precision (98.1%) and recall (98.7%) provide the basis for a high degree of confidence for the classification of clear traffic, (iii) the number of misclassifications is low, 13 and 19 that should belong to the clear traffic and to the DDoS attack respectively, (iv) the overall implementation of this machine learning approach, ART with $\rho = 0.07$, $\rho = 0.088$, and $\rho = 0.09$, is found overperforming when compared to FuzzyART set to the best suitable vigilance parameter values found through the FuzzyART parametogram.

Upon inspection of the confusion matrix computed for the classification through ART1 with a $\rho = 0.1$, the successive comments are noteworthy: (i) A sign about misclassifying some of the DDoS attack occurrences (60) into clear traffic because it shows a low value for recall (93.1.0%), when compared with the previous values for the vigilance parameter in ART1 that were systematically selected from the parametogram. This states that not all the true occurrences for the DDoS attack are detected, (ii) A slightly low value for precision (97.0%), which comes from the fact that some of the true occurrences (25) for the clear traffic are missed, after comparing with the previous values for the vigilance parameter in prior the prior experiments, and (iii) the number of misclassifications for both the clear traffic and a DDoS attack (highest

with 60) is not too excessive.

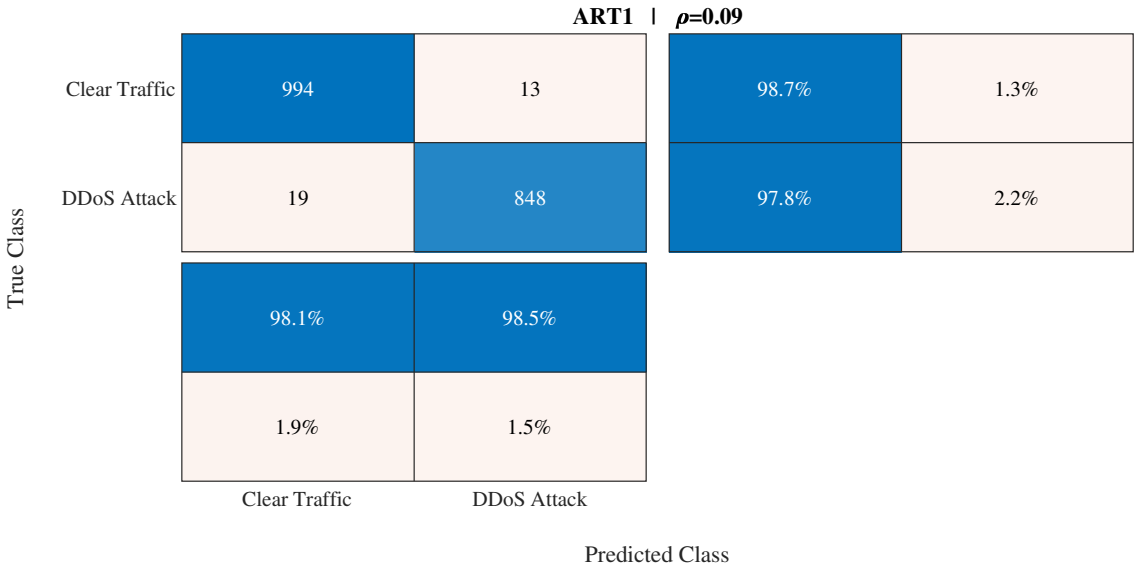


Fig. J.3. Confusion matrix for ART1 with vigilance parameter $\rho = 0.09$. The matrix displays: (i) 994 cases for clear traffic, (ii) 848 cases for a DDoS attack, (iii) 13 false cases for a DDoS attack, and (iv) 19 false cases for clear traffic. The column normalization (precision): (i) 98.1% for clear traffic, and (ii) 98.5% for a DDoS attack. The row normalization (recall): (i) 98.7% for clear traffic, and (ii) 97.8% for DDoS attack.

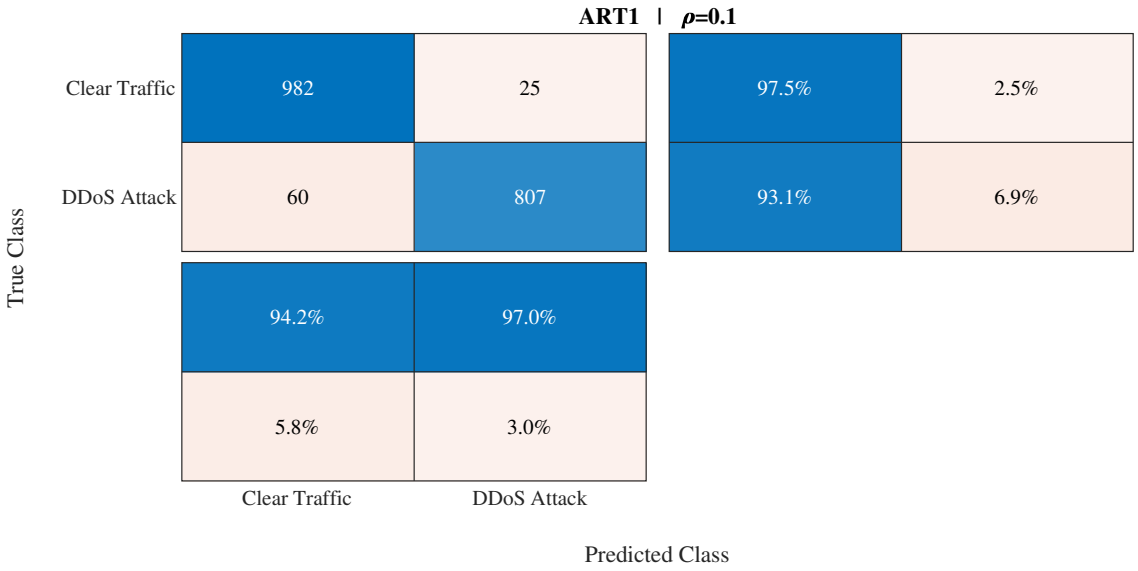


Fig. J.4. Confusion matrix for ART1 with vigilance parameter $\rho = 0.07$. The matrix displays: (i) 0 cases for clear traffic, (ii) 867 cases for a DDoS attack, (iii) 1007 false cases for a DDoS attack, and (iv) 0 false cases for clear traffic. The column normalization (precision): (i) 0% for clear traffic, and (ii) 46.3% for a DDoS attack. The row normalization (recall): (i) 0% for clear traffic, and (ii) 100% for DDoS attack.

Regarding the confusion matrix accomplished for the classification through ART1 with a $\rho = 0.9$, the successive points are striking: (i) For the DDoS attack class, it presents a high value for precision (100%) and a low value for recall (0.3%) providing an indication about misclassifying the a DDoS attack into clear traffic, (ii) for the clear traffic class, it denotes a low value for precision (53.8%) and a high value for recall (100%) providing an indication about a DDoS attack being misclassified into clear traffic, and (iii) a very high number of misclassifications (864) is found for the DDoS attack.

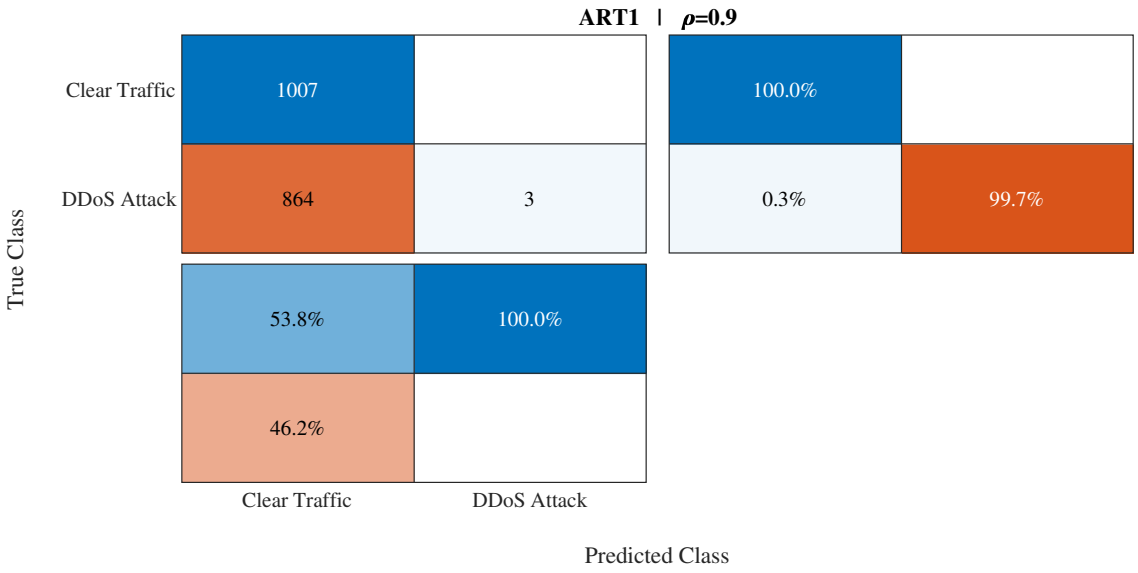


Fig. J.5. Confusion matrix for ART1 with vigilance parameter $\rho = 0.9$. The matrix displays: (i) 1007 cases for clear traffic, (ii) 3 cases for a DDoS attack, (iii) 0 false cases for a DDoS attack, and (iv) 864 false cases for clear traffic. The column normalization (precision): (i) 53.8% for clear traffic, and (ii) 100% for a DDoS attack. The row normalization (recall): (i) 100% for clear traffic, and (ii) 0.3% for DDoS attack.

APPENDIX K

CONFUSION MATRICES FOR FUZZYART PERFORMANCE

Regarding the confusion matrix achieved for the classification through FuzzyART with a $\rho = 0.1$, the succeeding remarks are significant: (i) It presents a low value for precision (46.3%) and a high value for recall (100%) providing an indication about misclassifying the clear traffic into a DDoS attack, (ii) it indicates a low value for precision (0%) and a low value for recall (0%) providing an symptom about clear traffic being misclassified into a DDoS attack, and (iii) a very high number of misclassifications (1007) is found for the clear traffic.

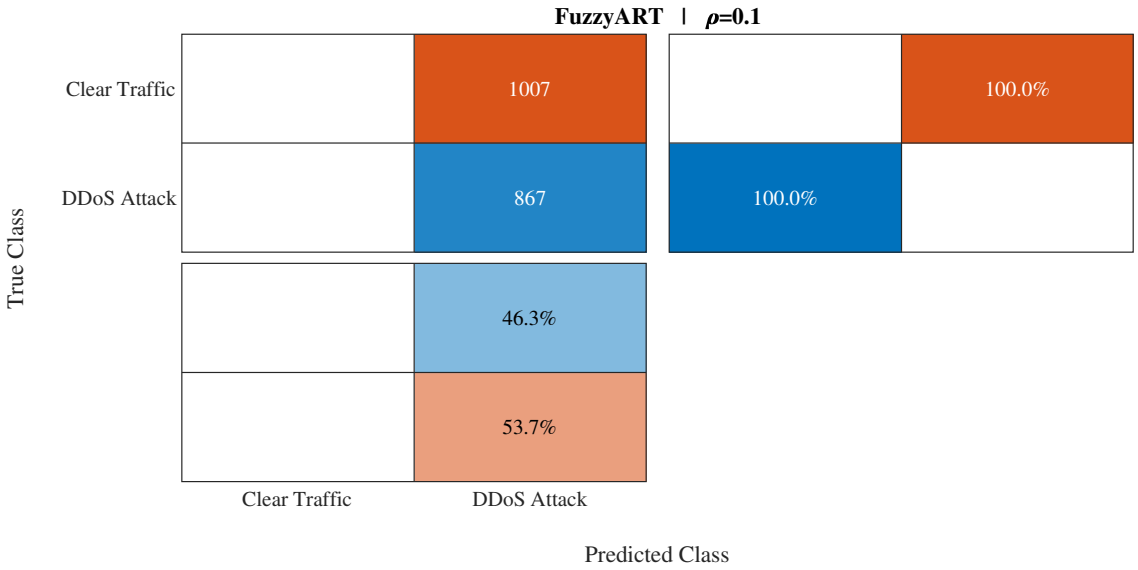


Fig. K.1. Confusion matrix for FuzzyART with vigilance parameter $\rho = 0.1$. The matrix displays: (i) 0 cases for clear traffic, (ii) 867 cases for a DDoS attack, (iii) 1,007 false cases for a DDoS attack, and (iv) 0 false cases for clear traffic. The column normalization (precision): (i) 0% for clear traffic, and (ii) 46.3% for a DDoS attack. The row normalization (recall): (i) 0% for clear traffic, and (ii) 100% for DDoS attack.

The confusion matrix acquired for the classification through FuzzyART with a $\rho = 0.632$ yields the succeeding meaningful reflections valuable of noting: (i) A high degree of confidence for the proper identification of the presence of a DDoS attack is concluded due to the high values for both precision (88.7%) and recall (84.8%), (ii) analogously, a high degree of confidence when it comes to properly identify the clear traffic is also sustained due to the high values for

both precision (87.4%) and recall (90.7%), (iii) the number of misclassifications starts to normalize (when compared to FuzzyART with a $\rho = 0.1$), 94 and 132 that should belong to the clear traffic and to the DDoS attack respectively, and (iv) the overall performance of this machine learning approach, FuzzyART with a $\rho = 0.632$, is found underperforming when compared to ART1 set to suitable vigilance parameter values.

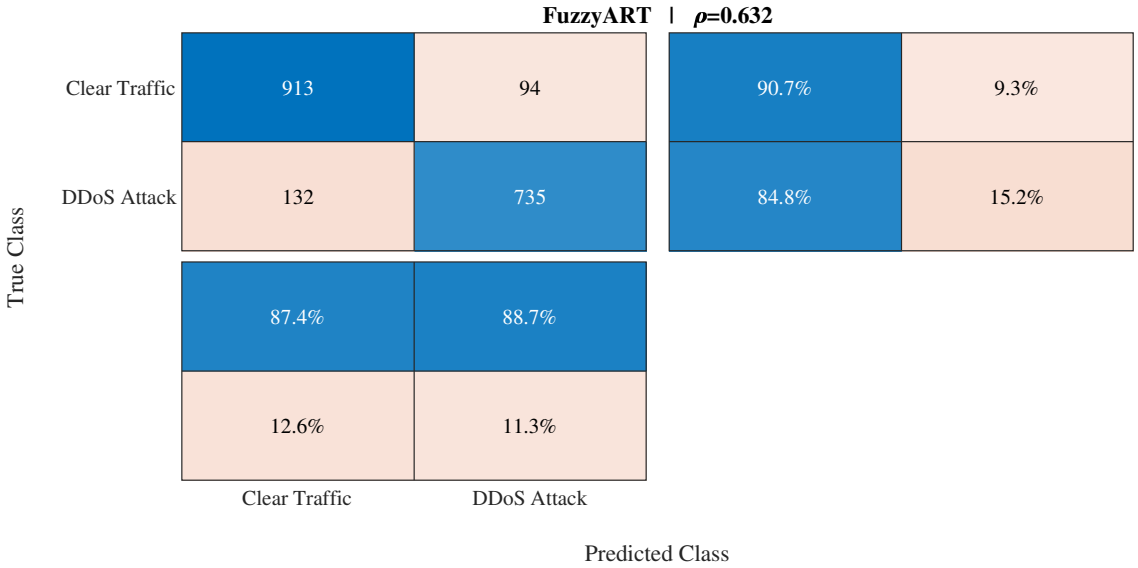


Fig. K.2. Confusion matrix for FuzzyART with vigilance parameter $\rho = 0.632$. The matrix displays: (i) 913 cases for clear traffic, (ii) 735 cases for a DDoS attack, (iii) 94 false cases for a DDoS attack, and (iv) 132 false cases for clear traffic. The column normalization (precision): (i) 87.4% for clear traffic, and (ii) 88.7% for a DDoS attack. The row normalization (recall): (i) 90.7% for clear traffic, and (ii) 84.8% for DDoS attack.

Setting the implementation of FuzzyART to a vigilance parameter $\rho = 0.633$ allows attaining a confusion matrix with the next contemplations: (i) The high values for both precision (89.7%) and recall (87.8%) postulate a high degree of confidence for pinpointing the presence of a DDoS attack, (ii) correspondingly, the high values for both precision (89.7%) and recall (91.4%) support a high degree of confidence for properly identifying the clear traffic, (iii) the number of misclassifications is lower to FuzzyART with a $\rho = 0.1$, 87 and 106 that should belong to the clear traffic and to the DDoS attack respectively, and (iv) the overall performance of this machine learning approach, FuzzyART with a $\rho = 0.633$, is also found underperforming when compared to ART1 set to appropriate vigilance parameter values.

For the confusion matrix processed for the classification through FuzzyART set to $\rho = 0.634$, the following remarks are meaningful: (i) High values for both precision (88.5%) and recall (87.5%) are existent, which affords a high degree of confidence for the classification of the presence of a DDoS attack, (ii) similarly, high values for both precision (89.4%) and recall (90.2%) are shown, also delivering a high degree of confidence when identifying the clear traffic, (iii) low number of misclassifications, 99 and 108 that should belong to the clear traffic and to the DDoS attack respectively, are revealed, and (iv) the overall performance of this

machine learning approach, FuzzyART with a $\rho = 0.634$, is also found underperforming when compared to ART1 set to appropriate vigilance parameter values.

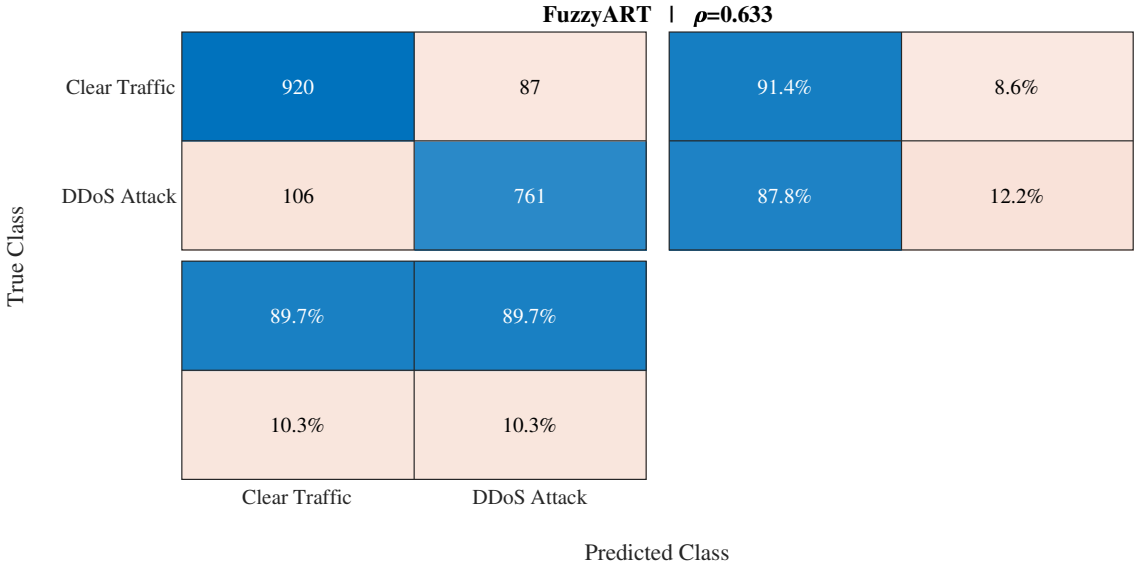


Fig. K.3. Confusion matrix for FuzzyART with vigilance parameter $\rho = 0.633$. The matrix displays: (i) 920 cases for clear traffic, (ii) 761 cases for a DDoS attack, (iii) 87 false cases for a DDoS attack, and (iv) 106 false cases for clear traffic. The column normalization (precision): (i) 89.7% for clear traffic, and (ii) 89.7% for a DDoS attack. The row normalization (recall): (i) 91.4% for clear traffic, and (ii) 87.8% for DDoS attack.

Upon assessment of the confusion matrix calculated for the classification through FuzzyART set to $\rho = 0.9$, the following arguments are outstanding: (i) An indication about misclassifying the a DDoS attack into clear traffic is sustained because it presents a high value for precision (100%) and a low value for recall (1.6%), (ii) a second signal about a DDoS attack being misclassified into clear traffic comes from the fact that it holds a low value for precision (54.1%) and a high value for recall (100%), and (iii) a very high number of misclassifications (853) is exposed for the DDoS attack.

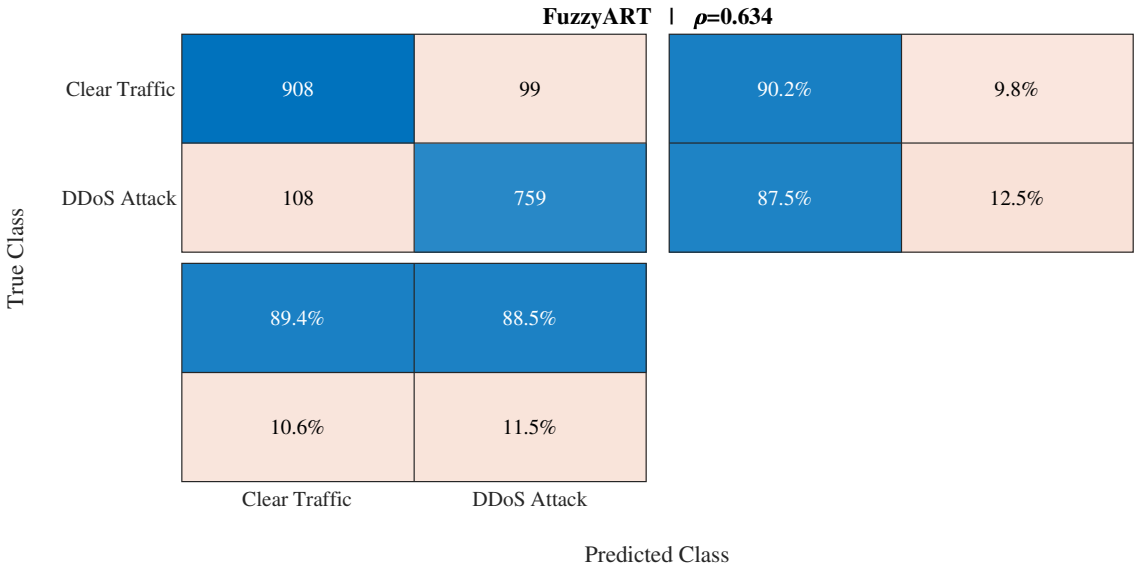


Fig. K.4. Confusion matrix for FuzzyART with vigilance parameter $\rho = 0.634$. The matrix displays: (i) 908 cases for clear traffic, (ii) 759 cases for a DDoS attack, (iii) 99 false cases for a DDoS attack, and (iv) 108 false cases for clear traffic. The column normalization (precision): (i) 89.4% for clear traffic, and (ii) 88.5% for a DDoS attack. The row normalization (recall): (i) 90.2% for clear traffic, and (ii) 87.5% for DDoS attack.

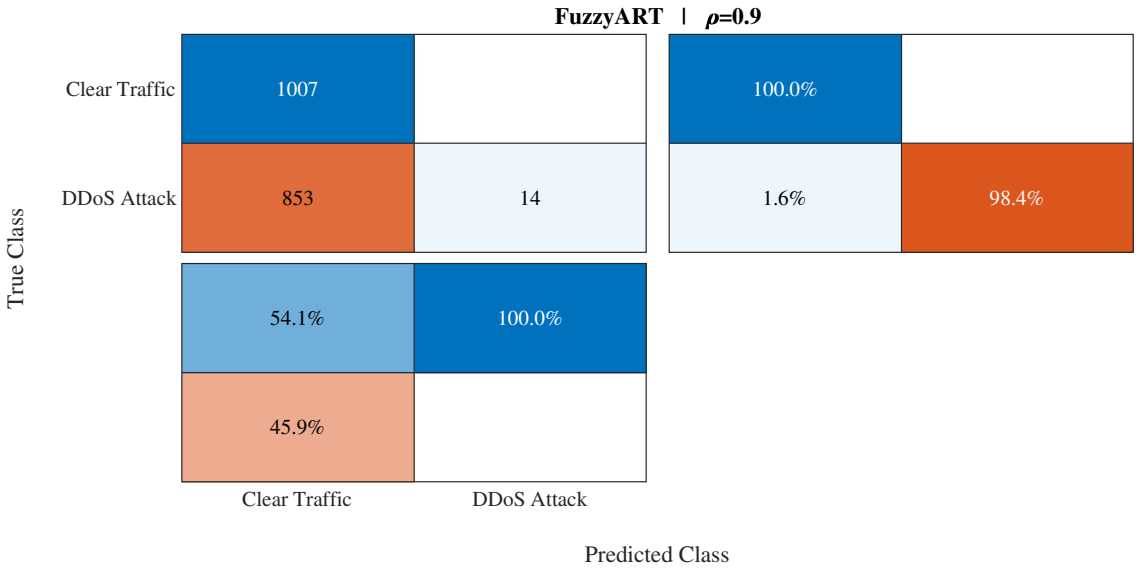


Fig. K.5. Confusion matrix for FuzzyART with vigilance parameter $\rho = 0.9$. The matrix displays: (i) 1007 cases for clear traffic, (ii) 14 cases for a DDoS attack, (iii) 0 false cases for a DDoS attack, and (iv) 853 false cases for clear traffic. The column normalization (precision): (i) 54.1% for clear traffic, and (ii) 100% for a DDoS attack. The row normalization (recall): (i) 100% for clear traffic, and (ii) 1.6% for DDoS attack.

APPENDIX L

INDUSTRIAL INTERNET OF THINGS

Recently, one can observe a flourish of proposals aimed at giving social-like capabilities to the objects in the IoT. Such proposals address the design of conceptual platforms, software implemented, to develop and implement complex applications requiring direct interactions among objects. The major goal is building techniques to enhance the level of trust between objects that are "friends" with each other in a network. A social paradigm could definitely guarantee network navigability even if the number of nodes becomes orders of magnitude higher than in the traditional Internet. This navigability requires unique addressing schemes over the standard communication protocols to provide information and services to the final users. Trillions of objects are expected to take a major active role in the future network, bringing physical world data into the world of digital content and services. This resulting networking paradigm, the IoT, would provide a paramount set of opportunities to users, manufacturers, and service providers with a wide applicability in many productive sectors (*e.g.*, automation, environmental monitoring, healthcare, inventory and product management, smart grid, smart home and workplace, security and surveillance). Social networking concepts integrated into the IoT would allow objects to interact in a human-like fashion in an advanced machine-to-machine (M2M) communication [AtIM014].

It has been more than 20 years since the first popular graphical web browser, Mosaic, was released. Now there is a cloud, a digital universe 4,000 Exabytes (a stack of books from Earth to Pluto and back 80 times) which is freely available. Everyday computers in the form of smartphones and tablets tap into this cloud. Soon wearable embedded computers will also interact with this cloud. The world has embarked into the next Internet inception through the IoT. Its forecast announces that it will make the current Internet look trivial. The significance of this is because the physical world (the planet and everything on it) is seen as part of the Internet. Real physical presences, things (*e.g.*, goods, objects, machines, appliances, buildings, vehicles, animals, people, plants, soil), can be observed and controlled. An object in the physical world is abstracted into an entity in the Internet. This object abstraction requires: A unique identity (the IPv6 provides an unlimited number of identities), the ability to communicate, to count with senses, and to respond to remote control instructions [Barr012].

Different scenarios for understanding the IoT impact to society are: (i) *Connect with things*. In the IoT, a smartphone becomes a channel to tap into things and exploit them; (ii) *monitor things*. A pacemaker can be monitored through the following: A smartphone to provide early warning, a remote computer running powerful algorithms 24/7 and predict weeks or even months ahead that a patient is heading for a problem, relatives wanting to know that a loved one's heart is still beating. Telemedicine and eHealth are set to become one of the big areas in

the IoT. Disarray and confusion in the health system could possibly be eliminated or kept to a controllable degree; (iii) *search for things* through reality search engines (*e.g.*, where are my keys?, where is my child?, what is the temperature of my food?); (iv) *manage things*. 51% of the world's population lives in cities (some becoming megacities). Knowing where traffic, citizens, or energy are moving/flowing would allow managing resources better and avoid unwanted scenarios (*e.g.*, traffic congestions, make better use of renewals, use energy more efficiently, look for the health and security of all citizens); (v) *control things*. Smart meters communicating between the appliances at home and the grid allow deciding when laundry takes place. The grid decides based on load balancing, energy efficiency, and use of renewals when laundry should go on. Decisions about which type of electricity would be used (*e.g.*, green electricity or cheap electricity) are possible; and (vi) *play with things*. Superimpose a game environment in the real world around a gamer. The objects and people around become part of the game. The gaming industry as it is known today is set to be transformed [Barr012].

By 2030, each person is expected to be surrounded by 3,000 to 5,000 connected everyday things. Possible utopic scenarios either global or partial, or elements of them could appear (*e.g.*, humans delegating control of the planet and its resources to a network of cognitive informatics/computing). Such network would manage, look after, and allocate the resources to everybody according to their needs. Society may experience a transition from democracy to technocracy where the planed is ruled by technology companies. The IoT could create the ultimate global panopticon where all things can be seen by anyone and privacy may become meaningless. The IoT could become a weapon of mass disruption due to terrorism and hacking in systems (*e.g.*, industrial, energy, transport, healthcare, safety and security) connected to a global network. Recent reported hacks are the next: Deadly Wi-Fi pacemaker hack, computer worms affecting nuclear programs, insulin pump caused to deliver a fatal dosage, car electronics taken over remotely, and smart meters, to mention a few. Regardless of what the ultimate state of the world would be decades from now due to the IoT one thing is certain, it is set to change humanity. In order to make the IoT for the common good, the input and the support from people in the human and social sciences. Engineers need to interface and work alongside ordinary people to make the IoT for the good of society and individuals. The common good for societies would be based on exploiting *senses*, *data*, *information* (extracted through data mining), *knowledge* (extracted through knowledge engineering), and *wisdom* (ultimately derived from knowledge to move the human race forward) having the benefit of humanity as the main goal [Barr012].

The Internet has been one of the most important and transformative technologies ever invented by humans for humans. The Internet is like a fabric that is woven into everyone lives whether they are conscious about it or not. Countless services that humanity is dependent on would be inexistent without the Internet. The Internet in its Internet-of-People (IoP) form has certainly changed the world. The new form of the Internet, IoT, focuses not only on people but also on things. The IoT is the space where things share their *sensory experiences* through *communication* capabilities and can have a physical impact (*e.g.*, control) in the real world. The IoP and the IoT intersect in things and humans having the capability to sense (*e.g.*, touch, smell, see, taste, and hear) their environment. One has within reach the most advanced IoT device available, a smartphone. A smartphone knows: Its geo-location through the global positioning system (GPS), its tri-dimensional position through an accelerometer, how much light is

surrounding it, how close it is to the user's face, it knows what the user is saying to it, it has an eye to see its surroundings, and of course ability to communicate. There are already far more advanced devices in the IoT available like: (i) Jawbone bracelets capable of tracking user steps and physical activity, knowing how well the user slept, and communicating over the network; (ii) Google Nest thermostats know if there is people in a room or not, learn and track home owner's patterns to ensure comfort and save energy, and communicate over the Internet to be controlled; (iii) Philips Hue light bulbs create moods in a room by illuminating it matching a picture colours, dimming, respond to other things or remotely to humans; (iv) garage doors capable of being operated remotely; (v) weight scales that automatically log in the senses weight into an mobile app; (vi) pet trackers capable of logging in activities; (vii) Opto22 *programmable automation controllers* (PACs) give things the ability to sense and communicate. PACs are an alternative to programmable logic controllers (PLCs). Standard and commercial hardware, diverse protocols and open standards, an exception based logic foundation, distributed processing, common tag database, and multitasking, are PACs features. Correspondingly, proprietary network, communication, and programming and function (modelled after the relay ladder circuit logic foundation), continuous scanning, and mandatory duplication of data tags for interoperability, are PLCs features; (viii) the Hive Thermostat and Heating Control allows customers to remotely control the central heating system, locks, alarms, and security cameras. IoT devices in homes are expected to rise to trillions; (ix) Tado also provides similar IoT solutions for cooling and heating systems; (x) wearable medical devices linked to the IoT contributed to save 6.5 billions GBP during 2015 for the National Health Service (NHS) in the United Kingdom. The peace of mind put in the patient and families is priceless; and (xi) mobile apps providing services based on users geo-location and behaviour are set to indent new angles in marketing and insurances ([Houg015] and [Skeg015]).

Big businesses (*e.g.*, Apple, Cisco, General Electric, Google, Microsoft, and Samsung) care about the IoT ecosystem to the tune of billions of dollars. Start-ups have been acquired by big businesses as part of their strategic plan (*e.g.*, Samsung buying SmartThings for 200 million USD or Google buying Nest for 3.2 billion USD) in the IoT industry. This may be because the IoT was born in 2008, a point in time having more things than people in the Internet. In 2015 there are around 10 billions of things in the IoT on the planet and this number is expected to balloon to 50 billion (including automobile, healthcare, utilities and consumer electronics related IoT devices) by 2020. This is expected to cause probably the most massive financial movement, economic growth, humanity has ever experienced. The IoT with its inherent technologies, like ability to sense/acquire data and communicate, is expected to help making processes and systems that would make life easier in this planet in distinct aspects (*e.g.*, health, safety, comfort, convenience, and wisdom) [Houg015]. As an example of wisdom, consider a cognitive system capable of warning inhabitants in a given region about a natural disaster (*e.g.*, avalanches, earthquakes, forest fires, or tsunamis) long before it occurs through the IoT ecosystem.

An alternative scenario is described as follows: An eHealth device or skin electronic patch monitors its user vital signs overnight and finds out its user has high-blood pressure and erratic breathing. This device analysed the user's sleep for the last 8 hours, because of this, it is suggesting its user to take given medications and meet with the physician due to a pre-made appointment to further investigate health concerns. In this scenario, the physician has data available before the patient shows up. Would the user require immediate attention due to an

emergency, the required units (*e.g.*, an ambulance) can be dispatched to pick the patient up immediately, bring the patient to the hospital, and put the patient under care and observation. The culmination of this could be the physician saying to the patient: "*Everything will be fine, you were suffering a heart attack, but major damages were avoided because you got the required treatment in just a nick of time*" [Houg015].

Nevertheless, IoT brings challenges, pitfalls and blind spots like: (i) Humanity resisting change, (ii) the IoT technical side being highly complex because of a high amount of embedded technologies, and (iii) the most important of all certainly is security and privacy [Houg015]. The BigBrother, a mega state watching citizens' moves arbitrarily, has been an overall concern for Internet users. The big question is: How can arbitrary spying actions be counteracted? *Legislation* is the biggest defense/weakness against these actions because it provides freedom to or takes rights away from society. Legislation has a pivotal role for how society embraces the new technologies built upon the IoT (*e.g.*, The United States of America recently enacted/ratified a bill setting the terms and conditions within the National Security Agency (NSA) could monitor citizens). Legislation should be used for developing laws protecting identities and data, but it should also provide mechanisms for anonymizing and sharing data (in a safe and sensible manner) for the Big Data ecosystem causing further benefits to humanity. This is undoubtedly a concern with the IoT potential to generate annually 35 trillion GB of data by 2020. This is certainly a huge amount of information even for BigBrother to handle successfully [Skeg015]. It is incomparable and amazing to control a thing remotely, but it is absolutely frustrating when someone unauthorized somehow hijacks that capability. Security and privacy policies pose a large amount of research questions for the IoT. This is a key focal point in this thesis proposal. The author of this document is working towards making a bulb light that could be lit in the IoT as having a comparable degree of security as the banking system. Introducing new technologies has situated humanity dichotomising between users speaking the industry language or vice versa.

In the IoT, each person and thing would have a locatable, addressable, and readable counterpart in the Internet [Butt015]. The indisputable IoT wonder is not that all persons and devices would be connected to the Internet, but that every device and person interconnects and communicates with others [Fard015]. Objects can produce and consume services and interact/collaborate with counterparts toward a common goal. "Smart objects" able to discover new services, start new acquaintances, exchange information, connect to external services, exploit other objects' capabilities, and collaborate toward a common goal have already been conceptualized and designed to achieve a fully networked human society. Smart objects in the IoT need to operate in an extremely complex context full of *opportunities* as well as *difficulties* and ***threats***. These smart objects are a new generation of social entities that can: (i) Interact with other objects in an autonomous way with respect to the owners; (ii) Crawl easily the IoT made of billions of objects to discover services and information in a trust-oriented way; and (iii) Advertise their presence to provide services to the rest of the network. The concepts and technologies typical of social networks are applied to the IoT to foster resource visibility, service discovery, object reputation assessment, source crowding, and service composition. Currently, a generational leap from objects with a certain degree of *smartness* to objects with an actual social *consciousness* is taking place. This progression encloses three stages of increasing levels of objects social involvement: (i) Objects can post information about their state in the social networks of humans, (ii) objects can interact at the application layer in social networks with

humans and other objects, and (iii) objects socially interact with each other to build a communication network. The increasing autonomy of objects in IoT would require the application of Asimov's Three Laws of Robotics [(i) a robot may not injure a human being or, through inaction, allow a human being to come to harm; (ii) a robot must obey the orders given it by human beings, except where such orders would conflict with the First Law; and (iii) a robot must protect its own existence as long as such protection does not conflict with the First or Second Laws] to their social communities to keep an advantage for humans. Smart objects in IoT would translate the awareness of causal relationships into actions that have an impact in a social community [AtIM014].

Limited and fragmented small islands of heterogeneous smart objects, disconnected from each other in some IoT solutions built in isolation, are being avoided by implementing web protocols [*e.g.*, Device Profile for Web Services (DPWS) or Representational State Transfer (RESTful) APIs] into either the objects themselves or specific objects' proxies/gateways in the Web-of-Things (WoT). The services and information provided by the things can be incorporated in the open ecosystem of the Internet-of-Services (IoS) where applications can be created by using standard web languages and tools. The WoT paradigm is limited by difficulties in advertising, discovering, accessing, and exploiting the objects and their services. Sensing the physical world and acting on it is a desired capability by Internet users and services in the IoT [*e.g.*, SenseWeb, Xively (by LogMeIn), and Paraimpu (by CRS4) provide a central platform to share sensory data, develop and develop applications]. One's basketball shoes would be capable of posting data in a social network (*e.g.*, Nike+). Ericsson Research has worked/developed the Social Web of Things, which provides higher degrees of autonomy and interaction between objects. One of its objectives is to help people master the complexity involved in the IoT networking paradigm by having a clearer vision of the rationale governing interactions between the IoT elements. Ericsson Scientists have observed that people familiarize better with IoT technologies if the interactions between IoT objects are presented in analogy to the interactions they usually experience in known social networks. Everything gives each individual object a unique active digital identity (ADI). This ADI provides a permanent presence online. An ADI corresponds to a Thing accessible on the web. Everything has the required environment and engines to manage ADIs. Their business idea is that manufacturers may want to provide ADIs along with the assets they produce. ADIs can be linked by relationships resembling social relationships. Nevertheless, most of the envisioned interactions still occur between objects and humans (through a smartphone) [AtIM014]. Jeff Hagins, SmartThings founder and chief technology officer (CTO) shares important lessons from IoT manufacturers/developers/businesses: (i) The number of connected IoT devices produces no profit/value, (ii) solving real-world problems produces profit/value, and (iii) the apps installed on the products produce profit/value. According to Jeff Hagins, the IoT can be defined as "*an evolutionary development of the Internet in which software applications can easily make use of connected, everyday objects (accessible easily by software to provide rapid innovation), in order to solve real-world problems.*" Hence, the IoT came to transform the way in which humans live and work by making their lives safer, smarter, and more productive [Hagi013]. It is important to notice that the Jeff Hagins definitions do not consider the intrinsic security and privacy of IoT devices.

The main driving forces behind IoT are providing computing and communication power

and social entity to physical devices/objects. A degree of interaction is provided to these objects looking for the betterment of society. Vehicles are physical objects that started to incorporate social capabilities through the IoT. In 2014 alone, 85 million vehicles were shipped worldwide. Each of these cars had between 60 to 100 sensors (*e.g.*, engine management, safety, and security) on-board. By 2020, the number of sensors per vehicle is expected to grow to 200 placing 26 billion devices on the road by then (not including driver-less cars) [Skeg015]. Vehicles are now claiming a spot, known as Internet-of-Vehicles (IoV), for them in the IoT. The availability and steep advancement in new communication technologies has highly contributed to link vehicles to the Internet. Now, vehicles can easily exchange sensory, safety, efficiency, infotainment, and comfort-related information with other vehicles and infrastructures using Vehicular Ad Hoc Networks (VANETs) creating Vehicular Social Networks (VSN) sharing users centric information alike Mobile Social Networks (MSN) [AlSS015]. An example of such platform is the Toyota Friend Network where automobiles data is made available over a private social network. This network aims for improving customer service and building a virtual community among the owners in order to increase the brand customer loyalty. Nevertheless, the identification of the killer application and the definition of the underlying business model in IoT are still missing. Some efforts focus on smart environment applications, but it is not clear who should pay and why (*e.g.*, user, social instances cost included in the object's price, or user paying when applications are not yet available). Hence, these aspects are still open research questions [AtIM014].

Communication configurations supported by traditional networks are unicast, broadcast, multicast, anycast, and geocast. Smart objects social networks require new communication configurations in which the data consumers/receivers are characterized by their role/position in the social network. Such functionality can be realized as services at the application layer, but it would be much more effective to embed them as networking primitives. The new primitives should allow distinguishing whether a node is to be included among the data destinations based on its distance from the source, the types of relationships linking it to the source, and the policies (*e.g.*, privacy) set by both the source and the node itself [AtIM014].

Cyber-physical architectures supporting highly complex and mobile devices (*e.g.*, vehicles) in the Social IoT (SIoT) leverage on cloud-based VANETs. The networked interactions M2M in the Social IoV (SIoV) are a critical example of a complex environment in which a physical object (involving a vast number of distinct industries in its design, creation, production, and operation) exists [AlSS015].

The stored and transmitted data in existing objects, systems, industries, and concepts (*e.g.*, Smart Cities portraying home-to-home or building-to-building interactions and exploiting social network relationships to solve various necessities) that would become a reality in the near future is a concern for humanity. Technological advancements in Information Technology continue enhancing existing services and creating new ones in our society (*e.g.*, embedding microprocessors and communication capabilities into objects). Unlike Human Social Networks (HSN), IoT Social Networks (IoTSN) have features like: (i) Highly dynamic nodes, (ii) relations are based on similar configurations, owner interests, or manufacturer, and (iii) social interactions remain anonymous and include sensory data exchange [AlSS015].

A completely integrated world is the utopic scenario for sharing data and also experiences. The Internet allows us to watch live events taking place anywhere and talk

instantaneously to people in another country through protocols (*e.g.*, IP). The Internet protocol has a set of rules/guidelines allowing a person to connect with others worldwide. The Internet protocol forms the basis for the communications (*e.g.*, web surfing, phone calling, emailing, or texting) occurring worldwide today. Furthermore, the IP allows creating and connecting the electronic ecosystem of the IoT. Technologies are implemented, mutually exclusive, by either integration (*e.g.*, Apple technologies as a closed ecosystem) or a modular approach. An ecosystem is created with infrastructure, ideas, culture, and technology. An ecosystem should absorb and develop both evolutionary and disruptive components. Otherwise, it falls apart. The IoT ecosystem should be able to absorb modules attempting to disrupt its evolutionary process. Technologies, as the IP, are very good at supporting the IoT base, but its users are very good to disrupt everything. Hence, the IoT ecosystem must be incredibly robust to cope with a fast evolution and the complex disruptions occurring in it. All of this, considering humans as part of the ecosystem and not as only users [Savo015].

Platforms allowing users to interact with the physical world are set to become the most security hungry entities in the future because of the implications of the objects actions in IoT. This is expected to escalate even further in industrial contexts. “Conversations” among IoT objects may have deep implications in promoting the development of human society. Just imagine a fridge sending a self-driving car to the superstore to pick groceries to contribute to the family comfort. In the e-Health aspect, consider a healthcare manager capable of monitoring medicines deadline through social pills bottles, alert the physician system and get new prescriptions in time, get drugs availability from pharmacies through location services, and keep drugs in a safe place. Through the IoT, we would be witnessing a transition from networked robotics to social robotics, where robots would ask for help to their *friends* for a given purpose. Consider an industrial environment where a plant is capable of self-assessing its health so that critical parts that would start malfunctioning are automatically quoted and purchased so that proactive maintenance routines would be triggered to keep production in optimal levels. Certainly, these scenarios demand technologies capable of providing data hyper-security so that actions in the physical world would be safe to humanity. Data hyper-security is expected to become more intense in strategic cryptographic research agendas. The demand for better systems for securing the data created by the IoT is increasing inherently given the growth of networked intelligent objects.

The development of smart objects social networks for both IoT and IIoT raises serious concerns on security and privacy of sensitive data and information. This fact requires research efforts that confront the security of the communications and evaluate the objects’ trustworthiness. The proposed models should account for the way the resources interact with each other over time and promote a shift toward trust-centric communication models in which it shall be possible to infer the degrees of trustworthiness from computation of the degree of the shaped relationships [AtIM014]. All things in the IoT are ideally connected at anytime, available from anywhere, and reachable by anyone. As mentioned earlier, people are seen as having unique addresses and be part of the IoT ecosystem, but not only that, IoT devices would also be inside people (*e.g.*, smart pacemakers IoT capable and other advanced medical instruments like pressure and oxygen sensors). The high connectivity of devices in the IoTT, accessing critical infrastructure (*e.g.*, national defense assets, power plants, or smart traffic systems) of entire nations, is a tremendous concern in the domain of security and privacy ([ChPB013] and

[Butt015]).

In 1999, Neil Gross stated: "*In the next century, planet Earth will don an electric skin. It will use the Internet as a scaffold to support and transmit its sensations*". Humans are now living actively this revolution, embodied like the IoT, as a consequence of the accelerating attribute of technology. Some important time gaps between technologies manifestations are: 15,000 years from painting and drawing to agriculture; 5,000 years to writing and the wheel; 2,500 years to cities and states; 1,900 to the experimental method; 325 years to industrialism; 95 years to electricity, the telephone and the radio; 65 years to the vacuum tube computers; 38 years to modern personal computers; 15 years to the Internet; and 12 years to smartphones, the cloud and mobile computing. The shortening gap phenomenon between technologies manifestations was introduced by Ray Kurzweil as "*Law of accelerated returns*". This law implies that the more advanced humanity becomes, the faster humanity becomes at advancing. The IoT, with the Internet as its nervous system, is impacting every aspect (*e.g.*, aerospace, economy, education, finances, health care, manufacturing, retailing, security, or transportation) of human life on Earth [Fard015]. Nevertheless, humanity is in its infancy for finding the path to the *one* common global goal of making planet Earth a better place to live considering we all share it. Three maxims supporting this goal are: (i) People above machines, (ii) faces before screens, and (iii) grand challenges ahead of small conveniences. Questions addressing that the IoT is not about the centres of technology and finance or the top 100 cities in the planet are: (i) What physical asset/device requires instrumentation? (ii) What data is required from that device? (iii) Who owns the data? (iv) Who has access to the data? (v) Who determines access rights? (vi) What is the minimum communication method? (Considering data disclosures to keep transparency) and (vii) What happens when someone violates a rule/law? All humans should have a collective role on deciding what they want planet Earth to look like as it is shared with more machines [Reze014]. Humanity is moving from social media to social businesses that encompass multiple venues (*e.g.*, marketing and communication, sales, services and support, human resources, information technology, public relations and external communications, research and development, enterprise 2.0, Industry 4.0, and so on and so forth) where proximity, availability, responsiveness, usage of technology, transparency, expression, and the power to mobilize people to change systems and things are highly important. Nevertheless, the Internet, not branded, not owned by one company, open and considered a human right, through social networks is setting a new world owned by few and regulated by even less individuals. These facts make humanity concerned about issues like privacy, economical and ideological censorship, business cases, and data ownership [Caud012].

All concerning questions around **security and privacy** boil down to this one: *What if something*, *either a device or data*, *is accessed by someone and used in a way it is not intended to?* The granularity, layers, of the impact span of security flaws in the IoT is identified decreasingly as: Humanity, global infrastructures, national critical infrastructures, computational devices in organizations, smart homes, smart cars, humans as individuals, smartphones, wearable electronics, and implantable devices in the human body. A security flaw exploited in any of these layers can be either deadly for a person up to a numerous group or disruptive for a given entity (*e.g.*, user, organizations or countries). Ecosystems dependent on the Internet, as IoT, require both strong/reliable security technologies and them being easy to use. Security technologies require simplification so that all persons actually use them. Validation of security technologies

through disclosure is being challenged more and more either for letting business owning them profiting more or as an attempt to stop vulnerabilities. An example of this is version 5 of the hypertext mark-up language (HTML5) fully validated in October 2014, which includes closed source digital locks. This makes digital devices using HTML5 potential vessels of undisclosed vulnerabilities, which is an acute aspect when thinking that HTML5 is meant to replace mobile apps with the potential to make visible changes and manipulations in the real world. Providing security and privacy in the digital world is one of the toughest missions in engineering.

APPENDIX M

SIGNAL PROCESSING RELATED DEFINITIONS

M.1 Data

Digital data are defined as a collection (bag) or arbitrary finite-state representations of source information, with no concept of temporal or special separation between the bag elements and no concept of the bag origin or destination. Bag elements may be equal, according to bag theory, while set elements must differ. Examples of data could include either an intercepted encrypted stream of bits (without a known beginning or end), or a financial file, or a computer program ([Kins004] and [Kins017]).

M.2 Signals

A signal is a function of independent variables such as time, distance, temperature, or pressure. The value of the function is called its amplitude, and the variation of its amplitude forms its waveform. The waveform can be either (i) unchanging (DC), (ii) *periodic* such as *alternating* (AC) or oscillating, (iii) *aperiodic*, (iv) *chaotic*, or (v) *random* (stochastic). The signals are either (i) *analog* (continuous with infinite resolution), or (ii) *discrete* (sampled in time and space, but still with infinite resolution), or (iii) *digital* (discrete and quantized to a specific resolution), or (iv) *boxcar* (continuous, piecewise constant with step displacements, as formed after a digital-to-analog converter known as DAC) [Kins004].

M.2.1 Linear Time Invariant Signals

The signals are classified as linear time invariant (LTI), additive invariance, or scale invariant, multiplicative invariance. The LTI system theory is based on the idea that periodic waveforms shifted by multiples of the period are the same. This also applies to stationary or cyclostationary signals in the sense that that their statistics do not change (*i.e.*, either the *wide sense stationarity*, WSS, in which the first two moments do not change, or the *strict sense stationarity*, SSS, where none of the moments could change). Fourier (spectral) and wavelet (spectral and scale) transforms may be applied to such signals in order to extract appropriate features [Kins004]. Stationarity testing methods are also widely available in the literature [WiKP998].

M.2.2 Scale-Invariant Signals

Scale-invariant (fractal) are fundamentally different from the LTI signals [Worn996].

Their short-scale and long-scale behaviour are similar (*i.e.*, they have no characteristic scale). Such self-similar signals (*i.e.*, signals with one scale for time and amplitude) or self-affine signals (*i.e.*, different scales for time and amplitude) must be processed differently because well-separated samples in the signal may be correlated strongly. Unlike the LTI signals (whose Gaussian distributions have very short tails), the SI signals have power-law distributions that have long tails. Their higher order moments do not vanish. Consequently, detection, estimation, identification, feature extraction, and classification of fractal signals are all different from the LTI signals. Most of the physical signals are not LTI. Examples of such signals include speech, audio, image, video, telecommunication traffic signals, biomedical signals such as the electrocardiogram (ECG) and electromyogram (EMG), sonar, radar, seismic waves, turbulent flow, resistance fluctuations, noise in electronic devices, frequency variations of atomic clocks, and time series such as stock market and employment. They are often highly non-Gaussian, non-stationary, and in general have a complex and intractable (broadband) power spectrum ([Kins004] and [Kins017]). Many dynamical systems produce signals that are chaotic (deterministic, yet unpredictable in a long-term (*e.g.*, [Kins020], [PeJS004], [Spro003], [KaSc004], and [Schr991]) ([Kins004] and [Kins017]). The common assumption that both LTI and SI signals originate from (and are processed by) systems that do not change in time and space can rarely be assured because both artefacts (such as electronic and mechanical systems) and living organisms age and change with the environment [Kins004].

M.3 Symbols and Alphabets

A symbol σ_j is defined as a unique entity in a set. The form a symbol can take is limitless. Examples of symbols are: A letter or a punctuation mark in a specific natural language (e.g., a, A, α, ℵ, a Braille symbol, or a sign in the American sign language), a digit in a specific number system (*e.g.*, unary {1}, binary {0, 1}, octal {0, 1, …, 7}, hexadecimal {0, 1, …, F}), morphs (*e.g.*, an arbitrary font, an iconic language as Chinese, music notation, chemical expressions), pixels (*e.g.*, binary, grayscale, colour), or phonemes (elementary and indecomposable sounds in speech) [Kins004].

A set of such unique symbols form an alphabet. A source alphabet, Σ, is a set of symbols used to generate a message by the source. It is denoted by

$$\Sigma = \{\sigma_1,\ \sigma_2,\ ...,\ \sigma_N\} \tag{M.1}$$

where N is the cardinality (size) of Σ, and it is denoted by $N = |\Sigma|$. The $|\phi$ operator shall not be confused with absolute value in this case. Also, each symbol in Σ is independent from one another. Symbol independence leads to a message with symbols arranged in either a random or correlated pattern as given by the message pmf [Kins004].

For transmission and storage, each symbol σ_j must be encoded with other symbols from a coding alphabet, Γ_c, which is denoted by

$$\Gamma_c = \{\gamma_{c_1},\ \gamma_{c_2},\ ...,\ \gamma_{c_b}\} \tag{M.2}$$

where the cardinality $b = |\Gamma_c|$ gives a base of the number system from which the digits γ_{c_b} are drawn. This is also the base of the logarithm used in all the subsequent calculations. For example, the binary coding alphabet is $\Gamma_c = \{0,\ 1\}$ with $b = 2$.

The encoded symbols γ_j corresponding to the source symbol σ_j constitute the code alphabet, Γ, denoted by

$$\Gamma = \{\gamma_1,\ \gamma_2,\ ...,\ \gamma_N\} \tag{M.3}$$

The source and the code alphabet cardinalities usually match. As a side note, the formation of compact messages requires alternative alphabets and dictionaries [Kins004].

M.4 Strings and Messages

A string is a collection of symbols (a bag, according to bag theory) forming an entity s_j larger than a symbol, and still smaller than a message (*e.g.*, A string in English, "the", coded as a unit results in a more compact representation than coding it to three separate symbols).

A bag of all the symbols and strings forms a message denoted by

$$\mathrm{M} \equiv \mathrm{M}[\sigma_1,\ \sigma_2,\ ...,\ \sigma_M] \tag{C.4}$$

Where $\mathrm{M} \equiv |\mathrm{M}|$ is the size of the message and $\equiv$ denotes equivalence. The vectorial notation $[\circ]$ allows $\sigma_i = \sigma_j$ for $i \neq j$ versus the set notation $\{\circ\}$ precluding elements equality [Kins004].

M.5 Probability

If a message M has been formed, transmitted, and received, the pmf can be estimated directly from M. If the symbol σ_j occurs n_j times in the message of size $\mathrm{M} \equiv |\mathrm{M}|$, the *relative frequency of occurrence* of this symbol is defined

$$f(\sigma_j) \triangleq \frac{n_j}{\mathrm{M}} \text{ [dimensionless]} \tag{M.5}$$

where the symbol Δ above the equality sign denotes the relation by definition. With this definition, the following conditions are satisfied

$$0 \le f(\sigma_j) \le 1,\ \sigma_j \tag{M.6}$$

$$\sum_{j=1}^{N} f(\sigma_j) = 1 \tag{M.7}$$

$$\Pr = \lim_{\mu \to 0}(f(\sigma_j)) \tag{M.8}$$

where Pr stands for probability. Hence, probability in this research is defined as the relative frequency of occurrence when the population grows to infinity [Kins004].

APPENDIX N

TAXONOMIC IDENTIFICATION OF

DISRUPTIONS IN COMPUTER SYSTEMS

Taxonomies have proven valuable when categorizing both real-life and artificial phenomena. Taxonomies provide an overall reference for studying a research problem and allow a systematic approach and have become an important part of the scientists' toolbox for a long time [LaJL008]. In network security, taxonomies provide a useful and consistent framework for attacks classification. However, network security researchers still do not use taxonomies extensively. Previous taxonomies are focused on vulnerabilities rather than attacks. Hence, lacking abilities for describing attacks in and difficult for using them [WuOL011].

A taxonomy is not simply a neutral structure for categorizing specimens, it also embodies implicitly a theory of the universe from which those specimens are drawn. It defines what data are to be recorded and how like and unlike specimens are to be distinguished [LBMC993].

When creating a new taxonomy, the following four general properties need to be considered [LiJo997]: (i) The categories in a taxonomy should be mutually and *collectively exhaustive*, (ii) every category should be accompanied by *clear and unambiguous classification* criteria defining what specimens are to be put in that category, (iii) the taxonomy should be *comprehensible and useful for everybody*, not only to experts but also to users and administrators with less knowledge and experience of the field, and (iv) the terminology of the taxonomy should *comply* with the established terminology of the field [LaJL008].

N.1 Taxonomy of Data Collection Mechanisms

Robust data collection mechanisms (DCM) (of ultimate importance in computer systems and many high-impact activities such as debugging, optimization, measurement, profiling and detection) fed to a detection engine is critical in disruption detection systems. Most systems rely on network and system call data as the input to the *detection engine*. The taxonomy of DCM considered here (providing a framework for inspecting, evaluating, and comparing) is supported by Axelsson [Axel000], Debar et al. [DeDW999], Larson and Jonsson [LaJo006a], Larus [Laru993], Lunt [Lunt993], Schroeder [Schr995], a survey of existing mechanisms as presented in [LaJo006b]).

The *type and quality of data* relies heavily on how the *collection mechanism* is implemented and how it behaves during operation. In the disruption detection area, high quality input data is vital for the detection, and the properties of the collection mechanism are highly important for producing such data. The data collection taxonomy discussed excludes (i)

hardware implemented mechanisms, (ii) time or synchronization issues between parallel execution on multi-processor systems, and (iii) and distributed data collection [LaJL008].

This survey of DMC is included in this research because it considers a vast number of previously proposed taxonomies [LaJL008].

N.1.1 The Ecosystem of a Data Collection Mechanism

An ecosystem for DCM is constituted by: (i) The *executing process* is a running application or system program. (ii) The *log-trigger* is a set of machine instructions contained within the executing process. (iii) The *log-control* is a set of machine instructions also located in the executing process, or in a separate process. (iv) The DCM is then the combination of the log-trigger and the log-control. (v) The *data-target* is an addressable memory area within the system, and either internal or external to the running process. (vi) The *clock* (typically a local clock) providing the current time. And (vii), the *output device* to which the collected data is transmitted for display, storage, or further processing [LaJL008].

N.1.2 A Data Collection Mechanism in Action

When the *log-trigger*, part of the *executing process*, is reached by the instruction flow, an alert is sent to the *log-control*. Then the *log-control* collects the content of the *data target* (memory area) and time-stamps it with the *clock*. The collected and time-stamped data becomes a log-data record, which is sent by the *log-control* to the *output device*. This releases the log-control from its operations for awaiting next alerts sent by the *log-trigger* [LaJL008].

N.1.3 Classes in Taxonomy of Data Collection Mechanisms

The term class is used for the property upon which categorization is based. The following three classes are used in the revised taxonomy: *Realization* (when, how, and where is the mechanism implemented?), *behaviour* (when is the mechanism inserted during operation and what information is logged?), and *log-data* (what type of data does the mechanism produce?) [LaJL008].

The *realization* of a DCM consists of: (i) The point in time the log-trigger is inserted into the executing process (*pre-runtime* or *runtime*). The log-control is introduced either before or simultaneously with the log-trigger. (ii) The level of granularity, instruction or program level, of the log-trigger. An instruction level log-trigger has the resolution of the hardware architecture (microprocessor hardware registers and single instructions can be resolved), but its semantic meaning is harder to comprehend. (iii) Implementation of the log-trigger (*application level* or *system level*), and (iv) Implementation of the log-control (*application level* or *system level*).

The *behaviour* class categorizes possible methods for the log-trigger to activate the log-control, and what actions the log-control performs when activated. These methods are *trigger* (*e.g.*, time, event, or hybrid) or *action* (*e.g.*, state save, event save, or hybrid save) based.

A simple *log-data* record has both *descriptive data* (the observed state and event information regarding a specific target) and *temporal data* (time stamp of the observation).

It is important to highlight that the literature for DCM for disruption detection does not include sources considering information-based features. This point becomes even more outstanding when talking about polyscale analysis methods like the VFDT described earlier.

N.2 Taxonomy of Computer and Network Attacks

Preventing the success of an attack or reduce its harm led to the development of Automated Intrusion Response (AIR), which is an automatic response method to live (occurring) attacks. The taxonomy of computer and network attacks (CAN) from the point of view of AIR is studied and considered in this research. This CAN taxonomy considers three main classes: (i) Localities/sources (local and remote) from which attacks initiate, A_{sou}, (ii) possible methods and techniques (infection, exploitation, probing, deception, cracking, concurrency, and unknown) attackers adopt, A_{tec}, and (iii) harmful results (none, information leakage, rights escalation, and harm implementation) attacks cause, A_{res}. An automatic response and classification, considering the previous classes, is then defined as: $A = (A_{sou}, A_{tec}, A_{res})$ [WuOL011]. From all the distinct form of attacks, this research focuses in DDoS. This form of concurrent attack is described in the next subsection.

APPENDIX O

HISTOGRAM BINNING

The histogram, either a representation of the distribution of data or an estimate of the probability distribution data, is an analysis tool in widespread use within many sciences. As an example, particle physics data, both experimental and phenomenological, are analysed with histograms [KrKr014].

Histograms are nonparametric pdf estimators to both visualize data and to obtain descriptors, such as the entropy, of the underlying pdf. Quantities estimated from histogram-based pdf models depend on the choice of the number of bins [Knut013].

The most important parameter of a histogram is the bin width (length of the subintervals in the real line that is the histogram's base) because it controls the trade-off between presenting a picture with either too much detail "*undersmoothing*" or too little detail "*oversmoothing*" with respect to the pdf [Wand997].

The histogram represents the data to be comparable to the underlying probability distribution function (updf) of the phenomenon or model prediction that generated such data. The histogram, or any other representation, of a given data set as an observed probability distribution function (opdf). The updf depends on a set of parameters causing a set of bins for the opdf to be chosen in terms of those parameters. The analysis of histogrammed data can be highly dependent upon the set of bins chosen, which is not trivial. Histogram based analysis conducted at the Large Hadron Collider (LHC) yielded distinct results when distinct bin sizes are used to fit pdfs on a histogram. Hence, the choice of a bin set affects the analysis outcome. Bin sets are typically chosen by eye to be the smallest width bins such that there are "enough" statistics in the bins of greatest interest. Ofttimes the bin set is chosen under the constraint of aesthetic rather than scientific reasons [KrKr014].

O.1 Sturges' Binning Rule

The earliest published rule for selecting the bin width appears to be that of Sturges in 1926. In 1992, Scott points out that Sturges's method is more of a number-of-bins rule rather than a bin width rule itself as defined by the equation:

$$\hat{h}_S = 1 + \log_2 n \qquad \text{(O.1)}$$

where n is the number of samples [Stur926]. The bin-width defined by equation (H.1) leads to an oversmoothed histogram, especially for large samples. Smirnov concluded in the 50s that the optimal rate of decay of the bin width is $n^{-1/3}$ with respect to L_p norms [Wand997].

O.2 Doane's Binning Rule

In 1976, Doane [Doan976] proposed a modification to Sturges' rule that incorporates skewness. Both Sturges' rule and Doane's rule yield oversmoothed histograms, especially for a large number of samples. Doane's binning rule is defined as:

$$\hat{h}_D = 1 + \log_2(n) + \log_2\left(1 + \frac{|m_3|}{\sigma_{m_3}}\right) \tag{O.2}$$

where m_3 is the third statistical moment (skewness) of the pdf [Doan976].

The parameter σ_{m_3} is defined as [Doan976]:

$$\sigma_{m_3} = \sqrt{\frac{6(n-2)}{(n+1)(n+3)}} \tag{O.3}$$

O.3 Scott's Binning Rule

Scott's rule proposed in 1979 is optimal if the data under analysis is normally distributed. This normal reference rule is defined by

$$\hat{h}_{Sc} = 3.49\hat{\sigma}n^{-1/3} \tag{O.4}$$

where $\hat{\sigma}$ is an estimate of the standard deviation, so named because it is based on calibration with the normal distribution with variance σ^2. Modifications to this idea to allow for varying degrees of skewness and kurtosis have also been developed by Scott [Wand997].

O.4 Freedman-Diaconis' Binning Rule

Freedman and Diaconis research in 1981, allowed to gain understanding about that the asymptotic effect of the bin width on the mean L_2 error, or mean integrated squared error (MISE). Their work has led to the proposal of several rules of the form:

$$\hat{h}_{FD} = \hat{C}n^{-1/3} \tag{O.5}$$

where $\hat{C}$ is some statistic (*e.g.*, standard deviation) [Wand997].

Scott and Freedman-Diaconis rules appear to be useful estimates for unimodal pdfs similar to a Gaussian. However, they are known to be suboptimal for multimodal densities. This is because these rules are derived by assuming particular characteristics of the underlying pdf. In particular, the result obtained by Freedman and Diaconis is not valid for some pdfs, such as the Uniform pdf [Knut013].

O.5 Shimazaki-Shinomoto's choice

This histogram binning method is proposed to capture the time-dependent rate of neuronal spikes. The neurophysiological literature addresses that the bin size critically determines the goodness of the fit of the time histogram to the underlying spike rate. The Shimazaki-Shinomoto's method objectively selects the bin size from the spike count statistics, so that the resulting bar or line graph time histogram best represents the unknown underlying spike rate [ShSh007].

In the Shimazaki-Shinomoto's method, the optimal bin size is obtained by minimizing the cost function $C_n(\Delta)$ as:

$$\Delta^* \equiv \arg\min_{\Delta} C_n(\Delta) \tag{O.6}$$

where $\arg\min$ stands for the argument of the minimum operator that seeks to attain the smallest value and Δ defines the bin size [ShSh007].

The cost function $C_n(\Delta)$ is defined as:

$$C_n(\Delta) = \frac{2m_{1_{SS}} - m_{2_{SS}}}{(2n)^2} \tag{O.7}$$

where $m_{1_{SS}}$ and $m_{2_{SS}}$ represent the first statistical moment, mean, and the second statistical moment, variance, respectively, and Δ is changed until the smallest value for the cost function is achieved [ShSh007].

For Shimazaki-Shinomoto's method $m_{1_{SS}}$ and $m_{2_{SS}}$ are defined as:

$$m_{1_{SS}} = \frac{1}{N}\sum_{i=1}^{N} n_i \tag{O.8}$$

$$m_{2_{SS}} = \frac{1}{N}\sum_{i=1}^{N} (n_i - m_{1_{SS}})^2 \tag{O.9}$$

O.6 Debinning Algorithms

"Debinning" algorithms involve the relationship between the opdf and the observed cumulative distribution function (OCDF) from the data, and use it to construct a representation of the updf. The "binless" algorithm avoids bins, instead it determines the opdf as the smoothed, numerical derivative of the OCDF. The "binfull" algorithm uses the OCDF as a Monte Carlo generator for the opdf. A smoothing function is applied during the generation of a very large number of points [KrKr014].

APPENDIX P

MALICIOUS NETWORKS

Botnets have become the engine of cyberattacks, and it is a typical and dominant malicious network. A botnet (*e.g.*, Agobot, DSNXbot, evilbot, G-Sysbot, RBot, SDbot, and Spybot ([BCJX009], [SCGK011] and [ThSD007])) is a group of compromised computers on the Internet, and is controlled by botmasters through C&C. Botnets are pervasive, existing simultaneously in many commercial, production and control networks. Botnets sizes could be as large as millions [RZMT007]. Because of the number of machines, botnets can be lethal in bringing down targeted networks, either power grids or air traffic control networks, or communication networks. Attackers have mastered techniques, such as steppingstones, reflector, IP spoofing ([PeLR007], [ThSD007], and [WaSZ010]), code obfuscation, memory encryption [IaHa007], and peer-to-peer implementation technology ([ThSD007] and [BCJX009]) to cover and sustain their bots [Yu014].

Botnets have been investigated from various angles for around 10 years like: IP address distribution [McGu008], botnet probing events [LGCP011], Internet connectivity [ShKG012], size [RZMT007], and domain fluxing ([JCJL010] and [YRRR012]). Researchers utilize statistical learning techniques based on lexical features (*e.g.*, domain names length, host names, and number of dots in URLs) and other URLs features to automatically determine if a URL is malicious (*i.e.*, used for phishing or advertising spam ([MSSV009a] and [MSSV009b])). Botnets detection mechanisms based on (i) passive DNS traffic analysis against IP fast fluxing [PCDL009] and (ii) developing regular expression based signatures from a data set of spam URLs [XYAP008], have been developed. Infiltrated or subverted machines (bots) contact the botmaster at regular time intervals. These contact times can yield an opportunity for detection ([SCGK011] and [RZMT007]) (as used by Bothunter [GPYF007] and botminer [GPZL008]. Network telescopes [MSBV006] have been employed to observe malicious traffic [Yu014].

P.1 Data Collection of Malicious Networks

Available datasets are usually collected by honeypots [Womb015], glob experimental networks (*e.g.*, the planet lab [Plan015]), or large scale monitoring systems ([YWFX009] and [HNGH007]). Even with datasets in place, their further processing is a challenge. A DNS request failure dataset is usually the results of multiple botnets. In order to study the features of an individual malicious network, separation of the mixed data into clusters is required. The unsupervised machine learning is an existing and promising tool for the clustering challenge. Unsupervised learning includes two categories: Clustering and blind signal separation. Existing algorithms are principal component analysis, singular value decomposition, mixture models, k-means, hierarchical clustering [DuHS001], and graph spectrum [Mieg011]. Bots from one botnet

have more connections (*e.g.*, the Sybil attacks in cyberspace ([YKGF008] and [YSKG009])), however, connections amongst different malicious networks are either actually very limited or none ([YWWG011] and [Yu014].

P.2 Topology Modeling of Malicious Networks

The topology of a network is a piece of critical information as physicists believe that structure determines functions. It is especially important to understand the topology of botnets or other malicious networks. Knowing the topology of a given botnet allows for figuring out the key nodes of the network. Precise work with organizations (*e.g.*, Internet service providers (ISPs)) to fight against the botnet [Fars013], by terminating possible attacks or blocking communication path of bots, is then possible. However, the data "flatness" restricts this wanted precision because when a malicious packet is intercepted, only its source IP address and destination address are known and the path from the source to the destination is usually hard to obtain [Yu014].

The following two directions are promising for exploring topologies modelling:

Logical topology. Current network topology models are related to physical networks, which may not reflect exactly overlapped networks, such as botnets. A logical model could represent botnets more precisely on top of the physical nodes and links [Yu014].

Dynamic graph. Rather than having static graphs, it is necessary to inject dynamic elements into the classical graph theory because the Internet and botnets are changing constantly [Yu014].

P.3 Dynamics of Malicious Networks

Botnet dynamics includes many aspects, the most important one is the *number of bots* of a given botnet *against time*, and this interplay reflects the size of botnet. A direct method to count the number of bots is performing *botnet infiltration* to count bot IDs or IP addresses [Yu014]. The Torping botnet was hijacked for 10 days before the botmaster could be in command [SCGK011]. The Torping infiltration reported a footprint of 182,800 bots with a median and average size of live population of 49,272 and 48,532. During the 10 days takeover, 49,294 new infections were detected, a dataset of 70 GB was acquired, and it was determined that 78.9% of the infected machines were behind a network address translation (NAT), VPNs, proxies, or firewalls [SCGK011]. *DNS redirection* is another method to estimate a botnet size [DaZL006]. Bots captured by honeypots have been analysed, their C&C servers have been identified using reverse engineering tools to reveal the source code. Availability of the source code allowed manipulation of the DNS entry related to the botnet's Internet relay chat (IRC) server for redirecting the DNS requests to a given sinkhole for further examination and sizing a botnet footprint as big as 350,000 bots [DaZL006].

A challenging question in network security is: *What is the density of bot or malware in the network?* Plenty of research considers epidemic theory ([DeLD009] and [ZGTG005]) for modeling recruitment of malware networks [Yu014]. The epidemic model is the dominant tool for the estimating botnets size. It is the major theory for biology virus propagation modeling, and is also used by computer scientists [DeLD009]. Botnets member recruitment is similar to computer viruses. Henceforward, the usage of epidemic theory is appealing. Nevertheless,

computer virus modeling may lack accuracy after the early stage of propagation [ZGTG005]. Based on information theory, it is known that the probability distribution function is non-uniform [ChJi009]. Furthermore, the network topology has a big impact on the spread of malware [MiOK009].

Botnet dynamics is a time related problem. Therefore, time series analysis methods are particularly suitable to address this problem [NGWL013]. Additional *big research questions in network security* are related to periodicity, frequency of bot recruitment, attacking activities, the distribution of a specific botnet or virus, and Internet nodes compromised since the beginning of a botnet [Yu014].

P.4 Concealed Malicious Activity Detection

There are limited detection algorithms on malicious activity. Many illegal activities go undetected using current detection systems. The false negative rate of these systems is also essential challenge. Especially, when malicious bots demonstrate decent behaviour most of the time in order to fool detection systems [Yu014].

The network security community has started to consider the human aspect of criminal behaviour rather than focusing on technology-oriented methodologies only. This enhances the integration of the human criminal behaviour understanding and information techniques to reduce the false negative rate of detection as much as possible. The game theory [YuLi008] and social network technologies ([YKGF008] and [Yang008]) are suggested in the design of the detection algorithm of concealed malicious activities. Two research directions along these lines are: (i) *Identifying the boundary of detection for a given level of security investment using game theory*. A high frequency of malicious activity results in a high probability of being detected (*e.g.,* frequent vulnerability scanning, or sensitive data downloading would make the compromised computer stand out from its peers); and (ii) *Identifying malicious nodes using social network technologies*. Divide all Internet based nodes into two groups, benign and malicious (*e.g.*, members of one specific botnet). Communication among the nodes within each group being quite rich has been proven. Correspondingly, there is much less communication among nodes from different groups. The probability that the node is malicious increases if its amount of communication with known malicious nodes is high [Yu014].

P.5 Forensics of Malicious Networks

The capability of identifying the actual source of malicious packets sent across the Internet is referred as IP traceback. Methods of traceback rely on independent local networks with no global coordination and are incapable of accurately tracing back cyber criminals at the Internet level ([Lesl014] and [ReRL014]). Methods of IP traceback can be categorized into three major groups: (i) Deterministic packet marking (DPM) ([YKGF008], [DeFS002] and [XiZG009]). The source LAN marks the IP packets. It requires updating all the Internet routers for packet marking. However, it is problematic for scalability because only 25 spare bits are available in IPv4packets. Storage for packet logging for routers is currently unfeasible; (ii) Probabilistic packet marking (PPM) ([AlGo006] and [Good008]). Incoming packets are marked at the edge routers of the LAN where the potential victim resides. It only operates in a local range of the Internet (*e.g.*, ISP networks), where the defender has the authority; and (iii)

Information theoretical based [YZDJ011]. Flow entropy variations are measured at routers for tracing back attack sources ([BhBK015], [DaTh015] and [Yu014]).

The first and second methods require routers to inject marks into individual packets and are vulnerable to hacking (*i.e.*, packet pollution). The third method overcomes the disadvantages of the previous two, although global collaboration is required for a specific tracebacks. An alternative that provides limited results in today's cyberspace status, as direct traceback is almost impossible, is *attack source inferring* [Yu014].

P.6 Malicious Networks for DDoS Attacks

Botnets are established by botnet-writers through programs (bots or agents). These programs are installed on compromised computers (hosts or *zombies* ([PeLR007], [SCGK011] and [CCGP010])) on the Internet. The bots of a given botnet are controlled by a botmaster. A botnet has at least one C&C server to communicate with bots and collect their data. Botmasters change the C&C URL frequently (*e.g.*, weekly [SCGK011]) [Yu014].

Sophisticated botnets are the engines behind DDoS attacks [KSSS014], which are organized by attackers motivated by financial or political reward. Symantec's MessageLabs states that 90.4% of total emails were spam, from which many included viruses, phishing attacks, and web-based malware, in June 2009. Therefore, sending spam through botnets can help to conduct further network attacks [Yu014].

Researchers have applied signature-based methods to detect botnets. However, these methods cannot detect new botnets as their signatures are unknown and some botnets are polymorphic [LiJZ009]. IRC-based approaches overcome this problem partially. Botnet mechanisms and botnet detection techniques have been surveyed and their classes identified: signature-based, anomaly-based [ChMW013], DNS-based, and mining-based ([FeSR009] and [YKPB013]). In order to disguise their traces and malicious activities, botnet writers design new strategies and mechanisms to fly under the radar. Two recent advanced botnet mechanisms are discussed next [Yu014].

P.6.1 Fast Flux Mechanism and Detection

Fast flux (FF) refers to changing rapidly the mapping between multiple IP addresses and one single domain name [Ride008]. This technique makes it sophisticated to take down the C&C server. Networks that apply fast flux techniques are called fast fluxing network (FFN). Both legitimate and suspicious FFNs show characteristics like short time-to-live (TTL) and large IP pools [YuZW010]. Fast flux can be classified into: (i) *Single flux* where a domain name may be resolved a by *flux agents* (FAs) to different IPs in different time ranges, and (ii) *double flux* in which changes in both the FAs and the registration in DNS servers are done frequently. This provides an additional layer of redundancy within malware networks for a *fast fluxing network attack* (FFNA). Almost all compromised computers become FAs. Bots are added or removed from the botnet agent pool dynamically. Consequently, mechanisms blocking agents cannot take down the whole botnet ([ZHLK009] and [Yu014]).

Metrics to detect fast flux service network (FFSN) empirically have been developed where a higher score indicated a higher fluxing degree. Results showed that distinction between normal network behaviour and FFSN are feasible [HGRF008]. Behaviour analysis models where

the behaviour of FF domains, probed in close locations, is characterized have been also developed. Behaviour analysis showed the number of DNS queries required to confirm an FF domain [ZhLK009]. Real-time (minute level) detection models for FFSN, using both active and passive methods in a distributed fashion, have been developed including components like sensors, FF monitor database, and *fast flux monitor* (FFM) ([CTDB009] and [Yu014]).

P.6.2 Domain Flux Mechanism and Detection

The disadvantage of FF is a single domain name, which is the only failure point once fluxing is identified. Hence, hackers developed *domain flux* (DF), a more survivable mechanism. The DF mechanism generates domain names via a *domain generation algorithm* (DGA). The C&C server and bot agents follow the same algorithm seeded by the same value to obtain consistent dynamic domain names [Yu014].

Cases of researchers cooperating with the FBI by discussing DF techniques and providing research findings when taking over advanced DF based botnets ([Amin008], [PoSY009], [SCGK011], and [Wolf008]). At some point, bots would phone "home" or the "mother ship". This key component can be used by defenders to defeat botnets [Yu014].

By reverse engineering the DGA of Torping [SCGK011], researchers revealed that its botmasters did not pre-register all possible domains in advance. Earlier registration of the related C&C server domain names of Torping, by researchers rather than the botnet owners, allowed taking over the botnet temporarily. The size of the Torping was then estimated by counting unique node identifiers N_{id}. The advantages of this method over IP, potentially misled by DHCP, were analyzed [Yu014].

Supervised machine learning methods to detect and prevent users from visiting malicious web sites have been implemented [MSSV009a]. Classification models include *naive Bayes*, SVMs, and *logistic regression* on datasets (two malicious and two legitimate). These datasets were later used for implementing online learning approaches ([MSSV009b] and [Yu014]).

Lightweight anomaly detection approaches using *DNS failure graphs* based on failed DNS queries have been developed [JCJC10]. Interactions between hosts and unresolvable domain names (auto-generated by botnets) are attributed to correlated failures originated at the botnet DGA. Many other approaches attempting to detect DF are available in the literature ([PKKG010], [Yu014] and [YRRR012]).

P.7 Modelling Malicious Networks

The model for network virus infection and curing has been explored extensively. Models for monitoring and *early* detection of Internet worms based on epidemiology research [ZGTG005] have been developed. These models are appropriate for systems consisting of a large number of vulnerable hosts. Also, these models are effective at the *early* stage of the outbreaks of virus, and the accuracy of the models drops otherwise. The assumptions for this model are: (i) Two possible states, healthy or infected, for a given node in the network are defined, and (ii) The nodes are forever in the system with no curing process [Yu014].

Stochastic branching modelling for characterizing the propagation of Internet worms [ZGTG005] is a variant of the previous. This model focuses on the number of compromised computers against the number of worm scans, and a closed expression for this relationship can be

defined [Yu014].

Models considering time zone information $\alpha(t)$ (describing the impact of the number of botnets live members with diurnal effects) have been also implemented [DaZL006]. The epidemic models are current the mainstream methods for virus or malicious network in cyberspace [Yu014].

Epidemic theory has a long history in the study of biological infectious diseases. In the 1930s, Kermack and McKendrick published a series of papers titled "Contributions to the mathematical theory of epidemics". This seminal research is seen as the basis of current mathematical modelling of infectious diseases spreading. On different assumptions and scenarios, different epidemic models exist, such as the *naive* model, the *susceptible-infectious* model (SI), the *susceptible-infectious-susceptible* model (SIS), and the *susceptible-infectious-recovery* model (SIR) [Yu014].

There are three different states for each individual in epidemic modelling: *Susceptible* (S), *infectious* (I), or *recovered* (R). Susceptible individuals are those who have not been infected but could be infected. Infected individuals are those who have the capability of spreading a disease. Recovered individuals are those who used to be infected by a disease, but they have been cured [Yu014].

P.7.1 Susceptible-Infections (SI) Model

The total population N is finite. No curing process for the disease exists. The dynamics are described by

$$\frac{dI_t}{dt} = \beta I_t (N - I_t) \tag{P.1}$$

where I_t denotes the infected hosts at time *t*, and β is the *pairwise rate of infection* in epidemic theories. The solution to the previous equation is

$$I_t = I_0 e^{\beta N t} \tag{P.2}$$

where I_0 represents the hosts infected initially.

The discrete form of this model is defined as

$$I_t = (1 + \alpha\Delta) I_{t-1} - \beta\Delta I_{t-1}^2 \tag{P.3}$$

where Δ is the unit of time, and $\alpha = \beta N$ defines the infection rate, which represents the average number of hosts vulnerable to infection by one infected host per time unit [Yu014].

P.7.2 Susceptible-Infections-Susceptible (SIS) Model

A curing process exists in this model. An infected individual can be cured, but immunity to the disease is not achieved. Cured individuals stay in a susceptible state. No vertical transmission of the disease (all individuals are born susceptible), no disease-related deaths, and equal birth and death rates maintaining constant population, are assumed. The dynamics of an SIS epidemic model are

$$\begin{aligned}\frac{dS}{dt} &= -\beta SI + \alpha I \\ \frac{dI}{dt} &= \beta SI - \alpha I\end{aligned} \tag{P.4}$$

where α and β denote the recovery rate and the infection rate respectively [Yu014].

If the birth and death dates are unequal, then the population size is variable and defined by

$$\begin{aligned}\frac{dS}{dt} &= -\frac{\beta SI}{N} + (\alpha + \lambda)I \\ \frac{dI}{dt} &= \frac{\beta SI}{N} - (\alpha+)\lambda I \\ N &= S + I\end{aligned} \tag{P.5}$$

where λ denotes the birth rate.

P.7.3 Susceptible-Infections-Recovery (SIR) Model

In this model if individuals become infected, an immune state R can be developed. Hence, future infections are not possible. The SIR model dynamics are defined by

$$\begin{aligned}\frac{dS}{dt} &= -\beta SI \\ \frac{dI}{dt} &= \beta SI - \alpha I \\ \frac{dR}{dt} &= \alpha I\end{aligned} \tag{P.6}$$

For variable population size the SIR model is defined by

$$\begin{aligned}\frac{dS}{dt} &= -\frac{\beta SI}{N} + \lambda(I + R) \\ \frac{dI}{dt} &= \frac{\beta SI}{N} - (\alpha + \lambda)I \\ \frac{dR}{dt} &= \alpha I - \lambda R \\ N &= S + I + R\end{aligned} \tag{P.7}$$

Deterministic models, represented by differential equations of various forms, are popular tools. The size of susceptible and infectious population is a definite function of time in these models. These models describe dynamical interrelations among rates of change and population sizes. Mathematical theories for this type of models are well developed, and they are suitable for making predictions. Alternative modelling methodologies exist (*e.g.*, stochastic based modelling, and random graph based modelling) [Yu014].

APPENDIX Q

RESULTS OF SYNTHETIC CLASSES DETECTION THROUGH ART

Q.1 A Synthetic Class Representing the First Five Letters of the Modern English Alphabet

Here, a synthetic feature vector matching the size of the dataset containing the DDoS attacks in the number of occurrences of a COI is constructed and explored to study the behaviour of the ART1 vigilance parameter with a distinct dataset. A class in the feature vector includes the first letters of the modern English alphabet. The COI in this case is represented by a column vector concatenating five reshaped binary 7x5 matrices that graphically represent the letters A, B, C, D and E. The bottom row in the matrix representing the letter E is truncated so that it matches in size the binary representation of the quantized outcomes of the secondary operators applied to the variance and skewness multiscalors totalling 168 binary scalars.

Once this unrelated class to the DDoS attacks is formulated it would replace the occurrences of the DNS DDoS attack and the occurrences related to the H&R DDoS attack and healthy traffic is replaced with random binary scalars. The corresponding feature vector occurrences are shown in Fig. Q.1 where three segments appear very clear: (i) From the frame 1 to 425 is filled with randomness, (ii) From the frame 426 to 1305 the pattern corresponding to the concatenated version of the first letters of the alphabet is seen, and (iii) a final section of random binary scalars after frame 1305.

The method for drawing suitable values for ρ previously introduced is applied to the feature vectors shown in Fig. Q.1 in order to investigate the behaviour of the vigilance parameter. The corresponding results for this case are shown in Fig. Q.2.

The waveform shown in Fig. Q.2 has the following behaviour according to the zones identified in the dataset containing DDoS attacks: (i) The OG zone spans for values of ρ in the interval $[0,\ 0.08]$; (ii) the COI zone occupies the rest of the values for ρ denoted by the interval $(0.08,\ 1]$; and (iii) the OS zone is non-existent for the class describing the occurrences of the alphabet pattern.

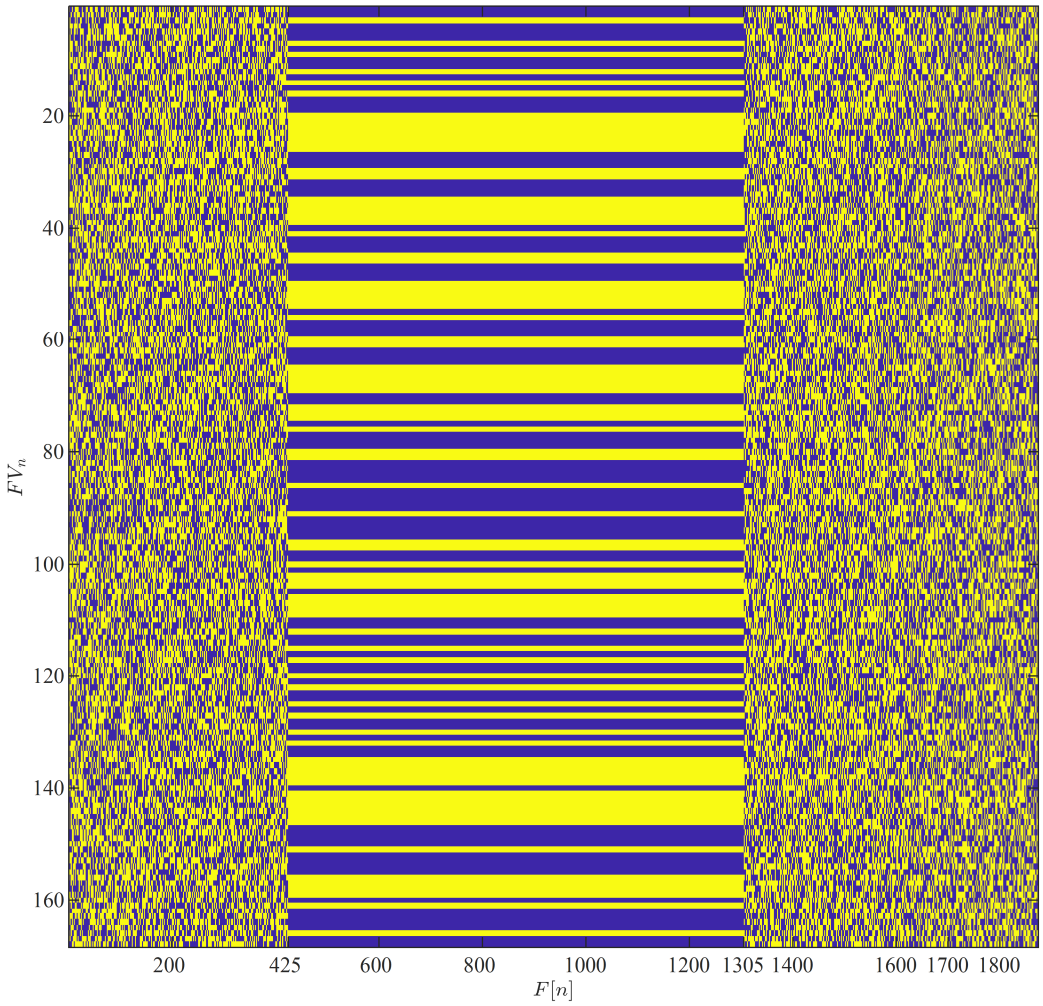


Fig. Q.1. Synthetic dataset with occurrences of the alphabet first five letters (represented in matrices 7x5 reshaped into a 168 binary vector) replacing the DNS DDoS attack. The rest of the occurrences are formed with random binary vectors that replace the H&R DDoS attack and the healthy traffic.

Figure Q.2 shows interesting behaviours: (i) The OG zone (left to the dash-dotted red line) can be described by a decaying exponential. One shall recall that there are 880 occurrences of the same event characterized by exactly the same feature vector values. Hence, when the vigilance parameter ρ is fixed close to zero, occurrences of events that are not related to the alphabet descriptor are merged with it, which causes the number of occurrences for the majority class to go over 880; (ii) the COI zone (starting at the dash-dotted red line and extending to the left until reaching $\rho=1$) maintains a constant value of 880 because the feature vector values describing the COI occurrences are identical; (iii) the OS zone has vanished because of the

descriptors describing the COI occurrences are identical.

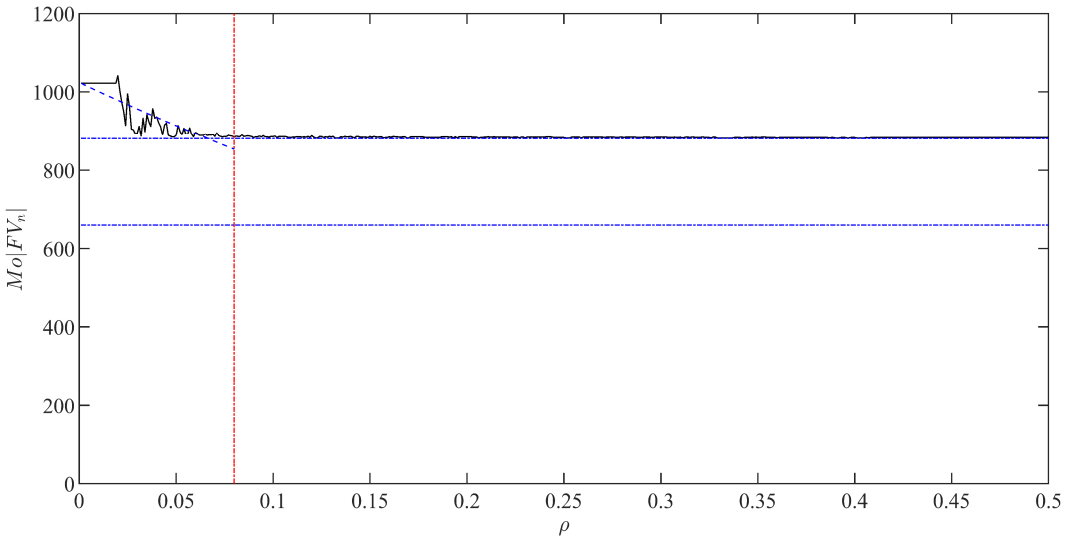


Fig. Q.2. Unsupervised classification of feature vector instances FV_n (containing 168 binary scalars matching the DDoS dataset) through ART1 with a vigilance parameter values for ρ spanning in the interval [0, 1].

The OG zone present in Fig. Q.2 is generalized through a piecewise single term exponentials of the form $C_1 e^{\rho C_2}$, where C_1 and C_2 are exponential coefficients and e is the natural logarithm. The optimization curve fitting Trust-Region algorithm is used to find the coefficients characterizing the OG zone. The COI zone present in Fig. Q.2 is described by a piecewise single section with constant value of approximately 880. The following equation represents this analysis in a compact form:

$$\mathrm{Mo}\,|\,FV_n\,| = \begin{cases} 1{,}024(e^{-2.284\rho}) & \text{for} \quad \rho = [0,\ 0.08] \quad \text{Overgeneralization} \\ \simeq 880 & \text{for} \quad \rho = (0.08,\ 1] \quad \text{COI} \\ & \text{Overespecialization nonexistent} \end{cases}$$

Q.2 A 10 Percent Noisy Synthetic Class Comprising Representing the First Five Letters of the Modern English Alphabet

The synthetic feature vector previously constructed is mixed with 10 percent of Uniform random noise. Therefore, this noisy feature vector is used to analyse the behaviour of the ART1 by changing the vigilance parameter.

The corresponding feature vector occurrences are shown in Fig. Q.3 where three

segments are distinguishable: (i) A section comprised of random scalars from the frame 1 to 425; (ii) a 10 percent noisy pattern corresponding to the concatenated version of the first letters of the alphabet from the frame 426 to 1305; and (iii) a final section of random binary scalars after frame 1305.

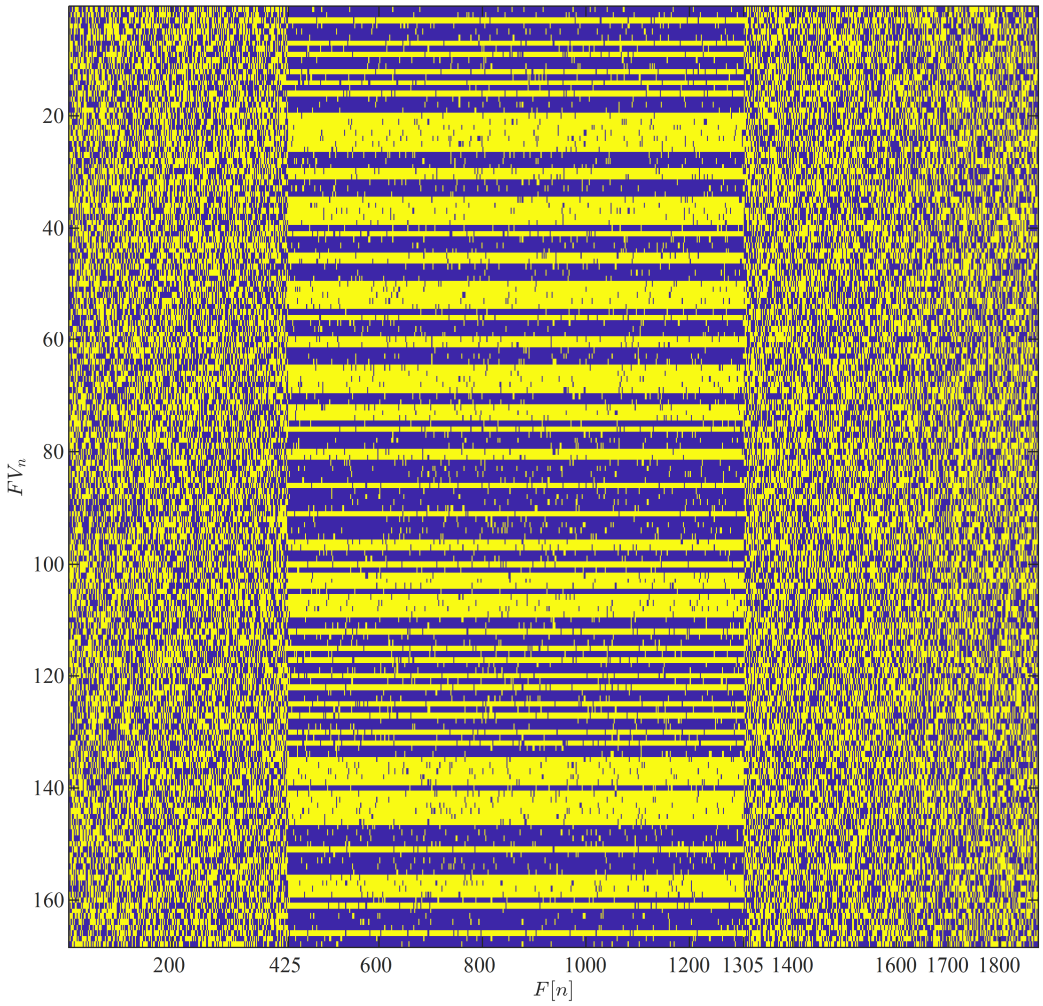


Fig. Q.3. Synthetic dataset with 10 percent noisy occurrences of the alphabet first five letters (represented in matrices 7x5 reshaped into a 168 binary vector) replacing the DNS DDoS attack. The rest of the occurrences are formed with random binary vectors that replace the H&R DDoS attack and the healthy traffic.

The method for drawing suitable values for ρ previously introduced is applied to the feature vectors shown in Fig. Q.3 in order to investigate the behaviour of the vigilance parameter. The corresponding results for this case are shown in Fig. Q.4.

The waveform shown in Fig. Q.4 has the following behaviour according to the zones

identified in the dataset containing DDoS attacks: (i) The OG zone spans for values of ρ in the interval $[0,\ 0.08]$; (ii) the COI zone occupies the rest of the values for ρ denoted by the interval $(0.08,\ 1]$; and (iii) the OS zone is non-existent for the class describing the occurrences of the alphabet pattern.

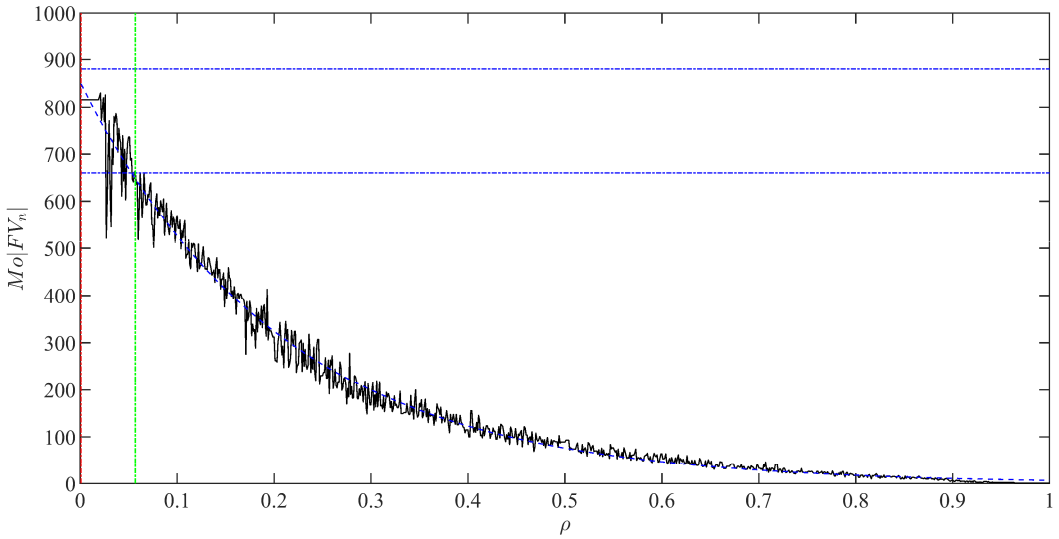


Fig. Q.4. Unsupervised classification of feature vector instances FV_n (containing 168 binary scalars matching the DDoS dataset) through ART1 with a vigilance parameter values for ρ spanning in the interval $[0,\ 1]$.

The behaviours captured in Fig. Q.4 are: (i) The OG zone (left to the dash-dotted red line) has vanished because of the potential uniqueness of each occurrence (caused by the added random noise onto the alphabet pattern) in the feature vector. Hence, now it is more difficult for ART1 merging non-overlapping classes (overgeneralize), which causes a smaller OG zone; (ii) the COI zone (starting at the dash-dotted red line and extending to the left until reaching $\rho = 0.057$ defined by the dash-dotted green line) is described by a decaying exponential portraying COI values in the interval [660, 880]; and (iii) the OS zone continues being described by a decaying exponential for values of the vigilance parameter $\rho = (0.057,\ 1]$ rendering values for $\mathrm{Mo}\left|FV_n\right|$ in the interval [0, 660]. The appearance of the OS zone follows from the uniqueness provided by the random noise when added to the alphabet pattern. Hence, ART1 disassociates the COI into smaller sets causing in the end that each occurrence is put into a single class with a single occurrence.

The waveform present in Fig. Q.4 is generalized through a single term exponentials of the form $C_1 e^{\rho C_2}$, where C_1 and C_2 are exponential coefficients and e is the natural logarithm. The optimization curve fitting Trust-Region algorithm is used to find the coefficients characterizing this zone. The following equation represents this analysis in a compact form:

$$\mathrm{Mo}\,|\,FV_n \models \begin{cases} & \text{Overgeneralization is nonexistent} \\ 853.5(e^{-4.837\rho}) \quad \text{for} \quad \rho = [0,\ 0.057] & \text{COI} \\ 853.5(e^{-4.837\rho}) \quad \text{for} \quad \rho = (0.057,\ 1] & \text{Overespecialization} \end{cases}$$

Q.3 A 20 Percent Noisy Synthetic Class Comprising Representing the First Five Letters of the Modern English Alphabet

Now, the synthetic feature vector is mixed with 20 percent of Uniform random noise. Consequently, this feature vector with a higher degree of noise is now utilized to analyse the behaviour of the ART1 when changing the vigilance parameter.

The corresponding feature vector occurrences are shown in Fig. Q.5 where three segments are distinguishable: (i) A section comprised of random scalars from the frame 1 to 425; (ii) a 20 percent noisy pattern corresponding to the concatenated version of the first letters of the alphabet from the frame 426 to 1305; and (iii) a final section of random binary scalars after frame 1305.

The method for drawing suitable values for ρ previously introduced is applied to the feature vectors shown in Fig. Q.5 in order to investigate the behaviour of the vigilance parameter. The corresponding results for this case are shown in Fig. Q.6.

The waveform shown in Fig. Q.6 has the following behaviour according to the zones identified in the dataset containing DDoS attacks: (i) The OG zone spans for values of ρ in the interval $[0,\ 0.08]$; (ii) the COI zone occupies the rest of the values for ρ denoted by the interval $(0.08,\ 1]$; and (iii) the OS zone is non-existent for the class describing the occurrences of the alphabet pattern.

The behaviours captured in Fig. Q.6 are: (i) The OG zone (left to the dash-dotted red line) has vanished because of the potential uniqueness of each occurrence (caused by the added random noise onto the alphabet pattern) in the feature vector. Hence, now it is harder for ART1 merging non-overlapping classes (overgeneralize), which causes a smaller OG zone; (ii) the COI zone (starting at the dash-dotted red line and extending to the left until reaching $\rho = 0.057$ defined by the dash-dotted green line) is described by a decaying exponential portraying COI values in the interval [660, 880]; and (iii) the OS zone continues being described by a decaying exponential for values of the vigilance parameter $\rho = (0.057,\ 1]$ rendering values for $\mathrm{Mo}\left|FV_n\right|$ in the interval [0, 660]. The appearance of the OS zone follows from the uniqueness provided by the random noise when added to the alphabet pattern. Hence, ART1 disassociates the COI into smaller sets causing in the end that each occurrence is put into a single class with a single occurrence.

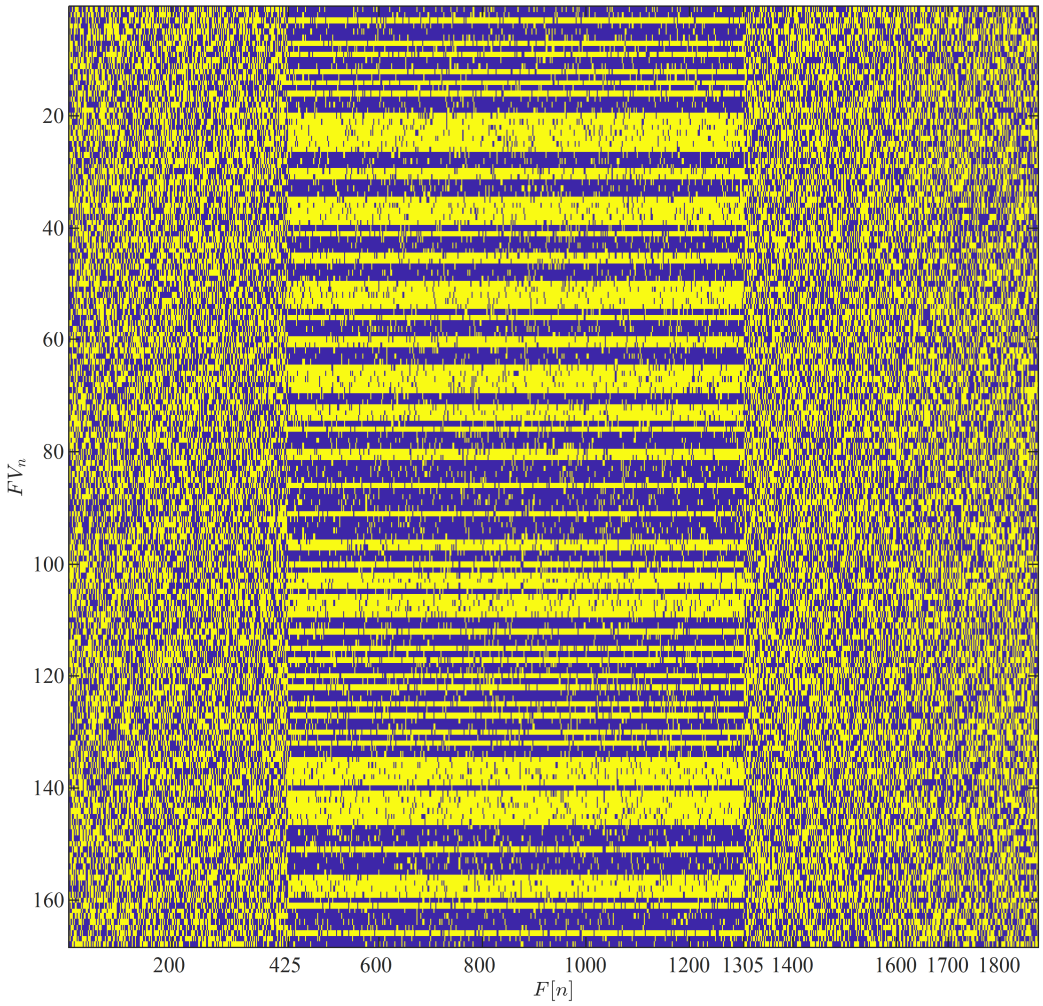


Fig. Q.5. Synthetic dataset with 20 percent noisy occurrences of the alphabet first five letters (represented in matrices 7x5 reshaped into a 168 binary vector) replacing the DNS DDoS attack. The rest of the occurrences are formed with random binary vectors that replace the H&R DDoS attack and the healthy traffic.

The waveform present in Fig. Q.6 is generalized through a single term exponentials of the form $C_1 e^{\rho C_2}$, where C_1 and C_2 are exponential coefficients and e is the natural logarithm. The optimization curve fitting Trust-Region algorithm is used to find the coefficients characterizing this zone. The following equation represents this analysis in a compact form:

$$\mathrm{Mo}\,|\,FV_n \models \begin{cases} & & \text{Overgeneralization is nonexistent} \\ 699.9(e^{-8.752\rho}) & \text{for} \quad \rho = [0,\ 0.022] & \text{COI} \\ 699.9(e^{-8.752\rho}) & \text{for} \quad \rho = (0.022,\ 1] & \text{Overespecialization} \end{cases}$$

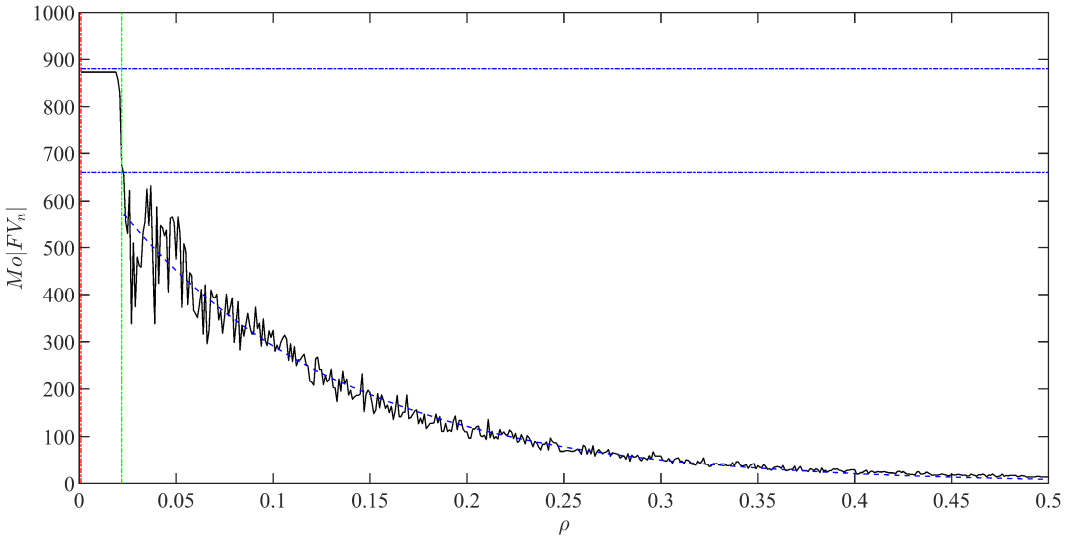


Fig. Q.6. Unsupervised classification of feature vector instances FV_n (containing 168 binary scalars matching the DDoS dataset) through ART1 with a vigilance parameter values for ρ spanning in the interval $[0,\ 1]$.

Q.4 A 30 Percent Noisy Synthetic Class Comprising Representing the First Five Letters of the Modern English Alphabet

The case for which the synthetic feature vector is mixed with 30 percent of Uniform random noise is now explored. Hence, this feature vector with a higher degree of noise is utilized to analyse the behaviour of the ART1 for a varying vigilance parameter.

The corresponding feature vector occurrences are shown in Fig. Q.7 where three segments are distinguishable: (i) A section comprised of random scalars from the frame 1 to 425; (ii) a 30 percent noisy pattern corresponding to the concatenated version of the first letters of the alphabet from the frame 426 to 1305; and (iii) a final section of random binary scalars after frame 1305.

The method for drawing suitable values for ρ previously introduced is applied to the feature vectors shown in Fig. Q.7 in order to investigate the behaviour of the vigilance parameter. The corresponding results for this case are shown in Fig. Q.8.

The waveform shown in Fig. Q.8 has the following behaviour according to the zones identified in the dataset containing DDoS attacks: (i) The OG zone spans for values of ρ in the interval $[0,\ 0.08]$; (ii) the COI zone occupies the rest of the values for ρ denoted by the interval

(0.08, 1]; and (iii) the OS zone is non-existent for the class describing the occurrences of the alphabet pattern.

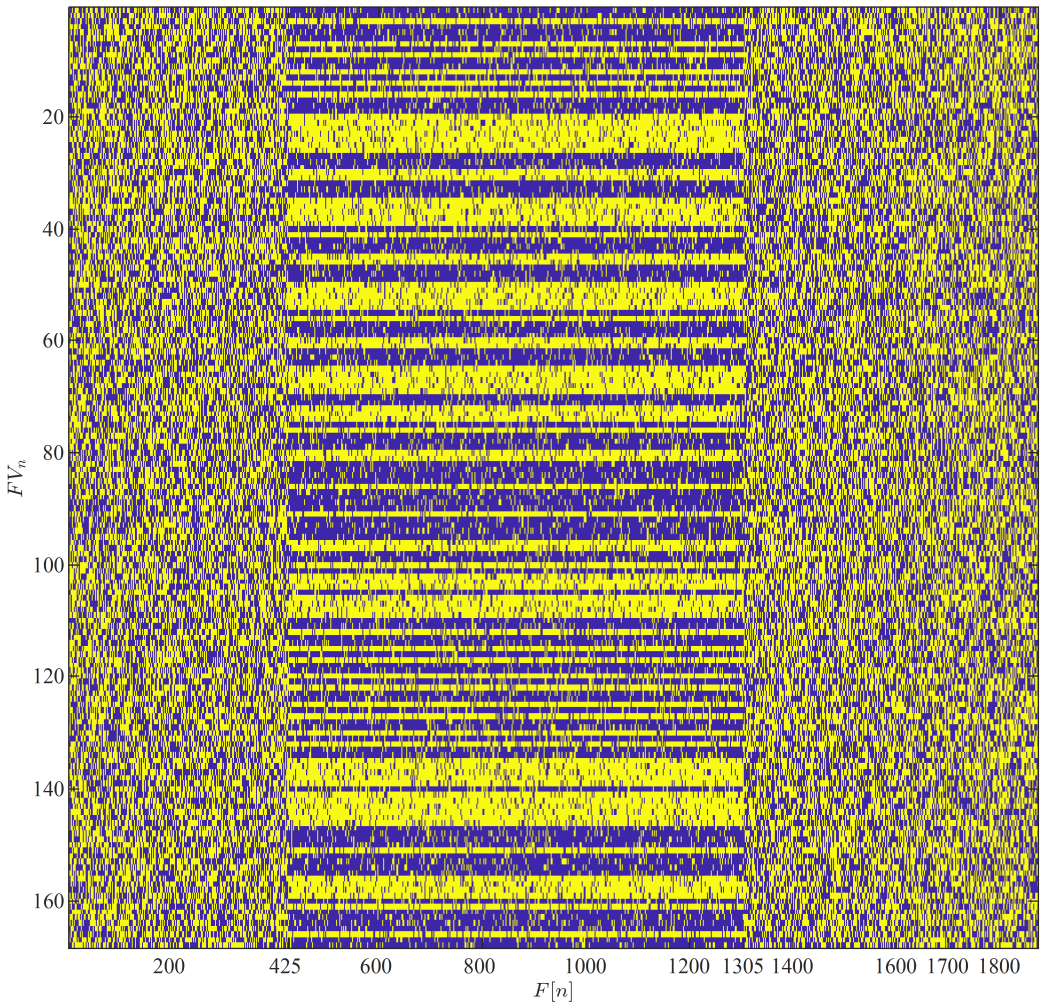


Fig. Q.7. Synthetic dataset with 30 percent noisy occurrences of the alphabet first five letters (represented in matrices 7x5 reshaped into a 168 binary vector) replacing the DNS DDoS attack. The rest of the occurrences are formed with random binary vectors that replace the H&R DDoS attack and the healthy traffic.

The behaviours captured in Fig. Q.8 are: (i) The OG zone (left to the dash-dotted red line) has vanished because of the potential uniqueness of each occurrence (caused by the added random noise onto the alphabet pattern) in the feature vector. Hence, now it is more challenging for ART1 merging non-overlapping classes (overgeneralize), which causes a smaller OG zone; (ii) the COI zone (starting at the dash-dotted red line and extending to the left until reaching

$\rho = 0.057$ defined by the dash-dotted green line) is described by a decaying exponential portraying COI values in the interval [660, 880]; and (iii) the OS zone continues being described by a decaying exponential for values of the vigilance parameter $\rho = (0.057,\ 1]$ rendering values for $\mathrm{Mo}\left|FV_n\right|$ in the interval [0, 660]. The appearance of the OS zone follows from the uniqueness provided by the random noise when added to the alphabet pattern. Hence, ART1 disassociates the COI into smaller sets causing in the end that each occurrence is put into a single class with a single occurrence.

The waveform present in Fig. Q.8 is generalized through a single term exponentials of the form $C_1 e^{\rho C_2}$, where C_1 and C_2 are exponential coefficients and e is the natural logarithm. The optimization curve fitting Trust-Region algorithm is used to find the coefficients characterizing this zone. The following equation represents this analysis in a compact form.

$$\mathrm{Mo}\,|\,FV_n \models \begin{cases} & \text{Overgeneralization is nonexistent} \\ 602.2(e^{-12.14\rho}) \quad \text{for} \quad \rho = [0,\ 0.023] & \text{COI} \\ 602.2(e^{-12.14\rho}) \quad \text{for} \quad \rho = (0.023,\ 1] & \text{Overespecialization} \end{cases}$$

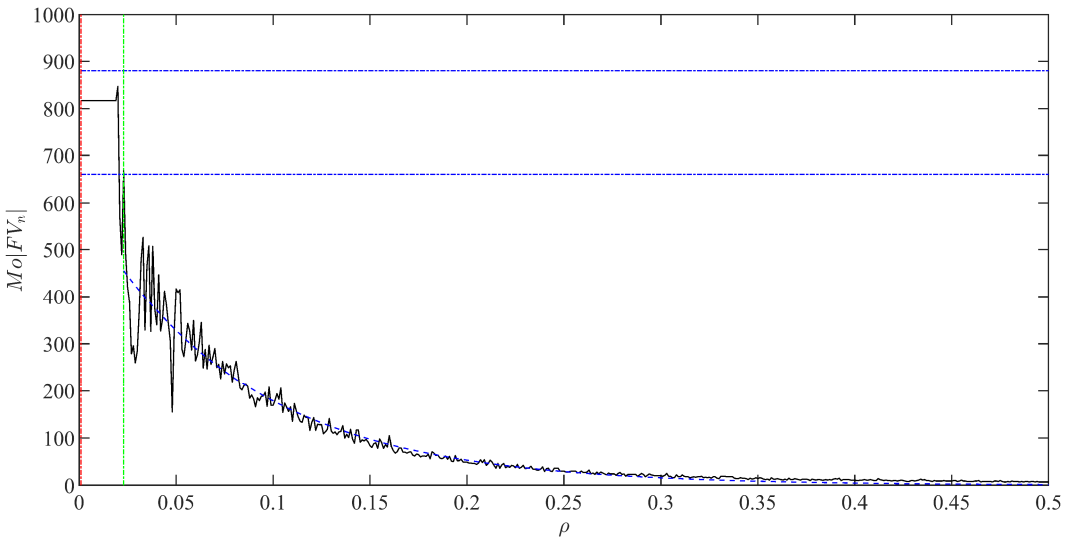


Fig. Q.8. Unsupervised classification of feature vector instances FV_n (containing 168 binary scalars matching the DDoS dataset) through ART1 with a vigilance parameter values for ρ spanning in the interval [0, 1].

APPENDIX R

RESULTS OF SYNTHETIC CLASSES DETECTION THROUGH FUZZYART

R.1 A Synthetic Class Condensing the First Five Letters of the Modern English Alphabet

Accordingly, a synthetic feature vector equivalent in size, because it contains the COI occurrences, to the dataset with the DDoS attacks. This synthetic feature vector is assembled for exploring the behaviour of the vigilance parameter of FuzzyART with an alternate dataset. A class in the feature vector includes the first letters of the modern English alphabet. The COI is represented by a real valued column vector (normalized) that is representative of the concatenation of five binary 7x5 matrices representing the letters A, B, C, D and E. The bottom row in the matrix representing the letter E is truncated, as beforehand, so that it matches the binary representation size of the quantized outcomes of the secondary operators applied to the variance and skewness multiscalors. The feature vector, composed of 42 real valued scalars, is then normalized in the interval $[0, 1]$ in order to be processed by FuzzyART.

Once this synthetic class, unrelated to the DDoS attacks, is formulated, the occurrences of the DNS DDoS attack are replaced by it. The healthy traffic and the occurrences related to the H&R DDoS attack are replaced by surrogate data through random shuffling. The corresponding feature vector occurrences are shown in Fig. R.1 where three segments appear very clear: (i) From the frame 1 to 425 is filled with surrogate data, (ii) from the frame 426 to 1305 the pattern corresponding to the concatenated, real valued, and normalized version of the first letters of the alphabet is seen, and (iii) a final section filled with surrogate data after frame 1305.

The method for drawing suitable values for ρ previously introduced for FuzzyART is applied to the feature vectors shown in Fig. R.1 in order to investigate the behaviour of the vigilance parameter. The corresponding results for this case are shown in Fig. R.2.

The waveform shown in Fig. R.2 has the following behaviour according to the zones identified in the dataset containing DDoS attacks: (i) The OG zone spans for values of ρ in the interval $[0, 0.578]$; (ii) the COI zone occupies the rest of the values for ρ denoted by the interval $(0.578, 1]$; and (iii) the OS zone is non-existent for the class describing the occurrences of the alphabet pattern.

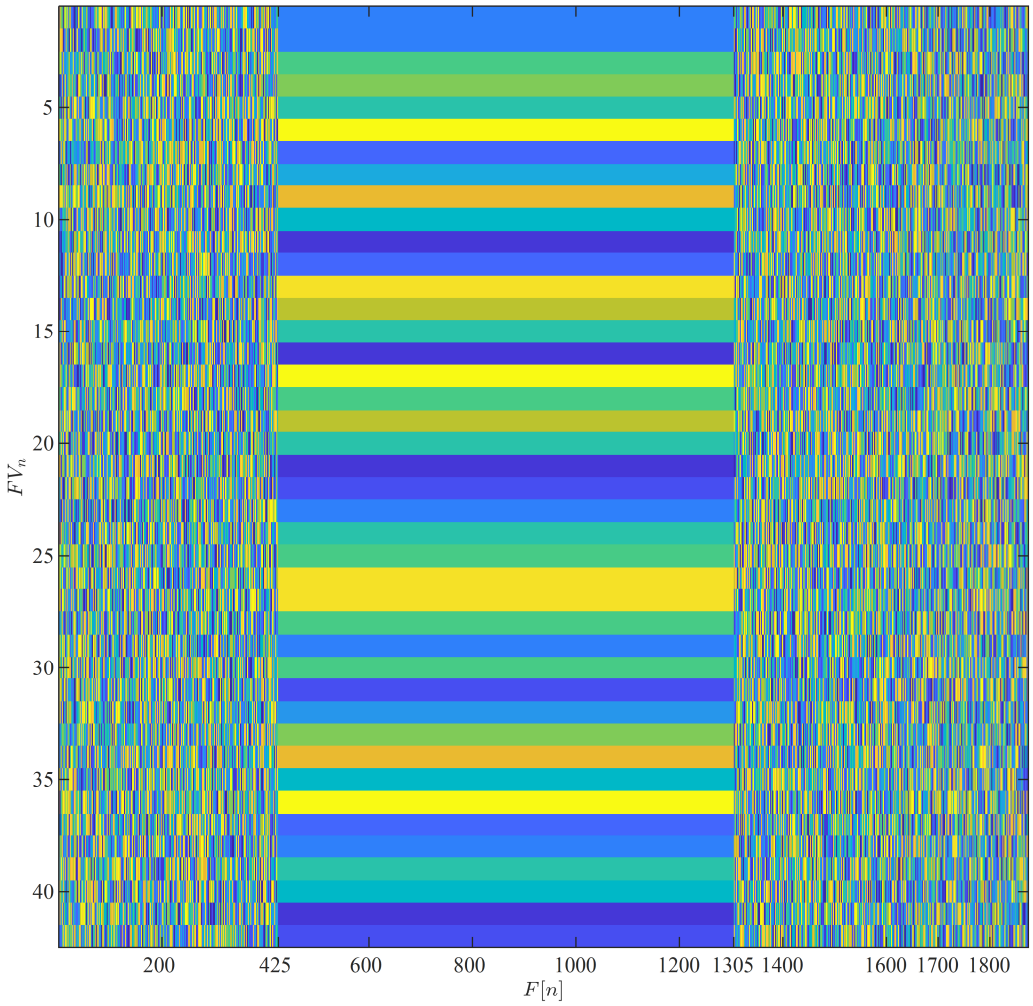


Fig. R.1. Synthetic dataset with occurrences of the alphabet first five letters (represented in matrices 7x5 reshaped into a 42 real valued vector) replacing the DNS DDoS attack. The rest of the occurrences are formed with normalized real valued vectors as part of the analysis applied to the original dataset. The H&R DDoS attack and the normal traffic is also normalized.

Figure R.2 shows interesting behaviours: (i) The OG zone (left to the dash-dotted red line located at $\rho = 0.578$) can be described by three segments almost constant and a decaying exponential. One shall recall that there are 880 occurrences of the same event characterized by exactly the same feature vector values. Hence, when the vigilance parameter ρ is fixed close to zero, occurrences of events that are not related to the alphabet descriptor are also merged with it, which causes the number of occurrences for the majority class to go over 880; (ii) the COI zone (starting at the dash-dotted red line and extending to the left until reaching $\rho = 1$) maintains a constant value of 880 because the feature vector values describing the COI occurrences are

identical; and (iii) the OS zone vanishes because of the descriptors describing the COI occurrences are identical.

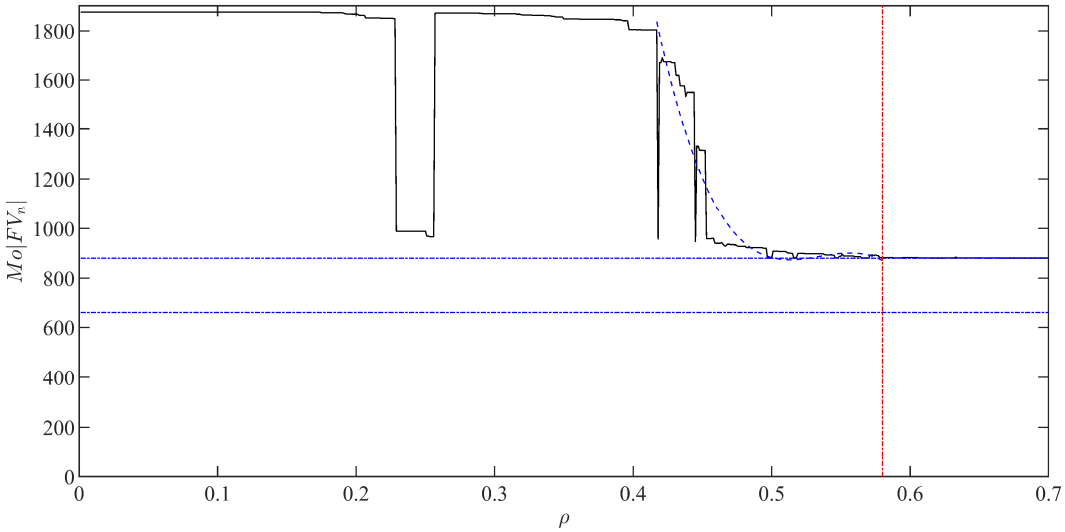


Fig. R.2. Unsupervised classification of feature vector instances FV_n (containing 42 real valued scalars describing the DDoS dataset) through FuzzyART with a vigilance parameter values for ρ spanning in the interval [0, 1].

The OG zone present in Fig. R.2 is generalized with piecewise curves. Three of these curves are approximately constant and one of them is described by a third degree polynomial of the form $a\rho^3 + b\rho^2 + c\rho + d$, where a, b, c, and d are the coefficients. The optimization curve fitting Trust-Region algorithm is used to find the coefficients characterizing this polynomial. The COI zone present in Fig. R.2 is described by a piecewise single section with constant value of approximately 880. Lastly, the overspecialization zone is non-existent. The following equation represents this analysis in a compact form:

$$\mathrm{Mo} \mid FV_n \models \begin{cases} 1860 \\ \text{for } \rho = [0,\ 0.228] & \text{OG} \\ \sim 990 \\ \text{for } \rho = (0.228,\ 0.256] & \text{OG} \\ \sim 1860 \\ \text{for } \rho = (0.256,\ 0.416] & \text{OG} \\ -6.385\times10^5\rho^3 + 1.025\times10^6\rho^2 - 5.475\times10^5\rho + 9.821\times10^4 \\ \text{for } \rho = (0.416,\ 0.578] & \text{OG} \\ \sim 880 \\ \text{for } \rho = (0.578,\ 1] & \text{COI} \end{cases}$$

R.2 A 10 Percent Noisy Real Synthetic Class Condensing the First Five Letters of the Modern English Alphabet

A synthetic real valued feature vector is injected with 10 percent of Uniform random noise. Thenceforward, this noisy feature vector is used to analyse the behaviour of the FuzzyART by changing the value of the vigilance parameter.

The synthetic feature vector instance is shown in Fig. R.3 where three segments are distinguishable: (i) A section from frame 1 to 425 includes noisy real-valued vectors representing normal network/Internet traffic; (ii) a 10 percent noisy real-valued pattern, from the frame 426 to the 1305, corresponding to the concatenated version of the first letters of the alphabet; and (iii) a final section from frame 1305 onwards with noisy real-valued vectors depicting normal network/Internet traffic.

The vigilance parameter ρ changes in the interval [0, 1] in order to investigate its behaviour while utilizing the feature vectors shown in Fig. R.3 for each new value of ρ. The corresponding results for this case are shown in Fig. R.4.

The waveform shown in Fig. R.4 has the following behaviour according to the zones identified in the dataset containing DDoS instances: (i) The OG zone spans for values of ρ in the interval $[0,\ 0.178]$; (ii) the COI zone occupies the rest of the values for ρ denoted by the interval $(0.178,\ 0.682]$; and (iii) the OS zone is placed in the interval $(0.682,\ 1]$ for the class describing the occurrences of the alphabet pattern.

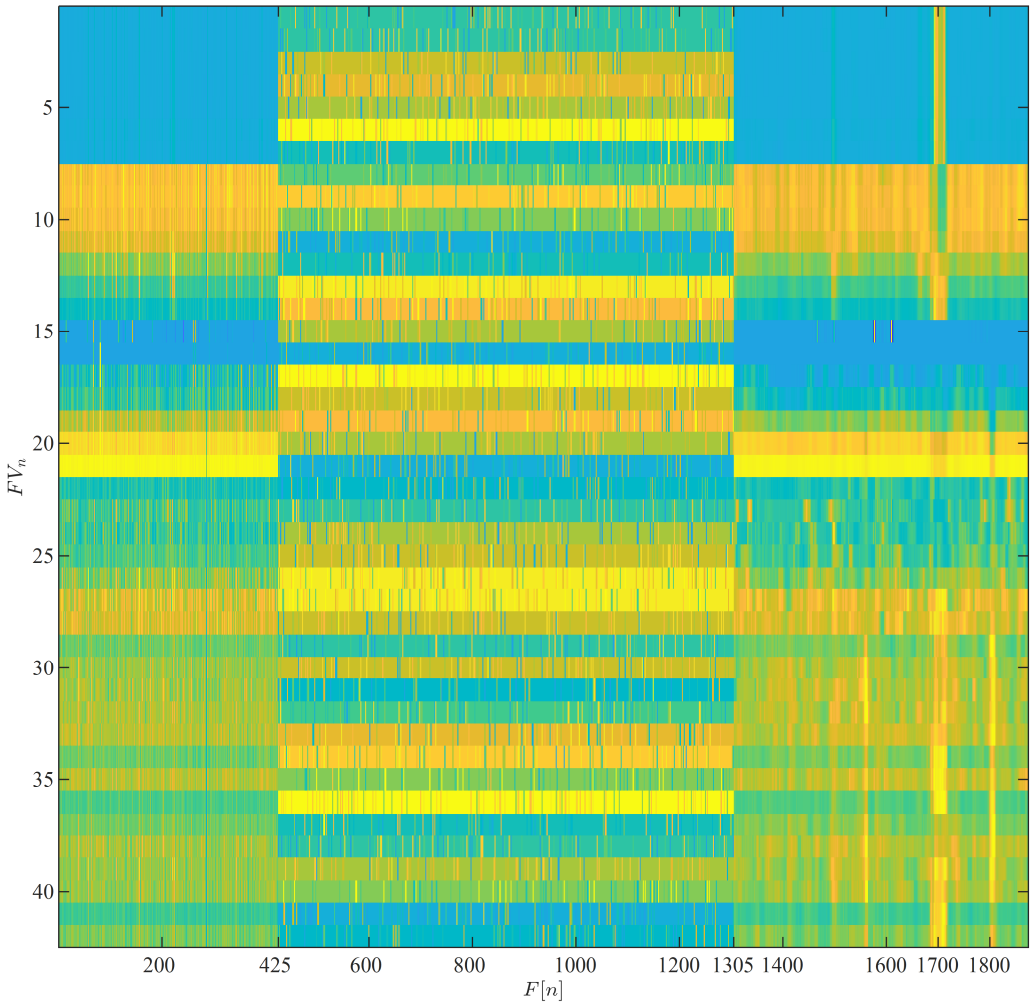


Fig. R.3. Synthetic dataset with 10 percent noisy occurrences of the alphabet first five letters (represented in matrices 7x5 reshaped into a 42 real valued vector) replacing the DNS DDoS attack. The rest of the occurrences are formed with normalized real valued vectors as part of the analysis applied to the original dataset. The H&R DDoS attack and the normal traffic is also normalized.

Figure R.4 has presence of three zones with specific behaviours: (i) The OG zone (left to the dash-dotted red line) has a smaller interval $\rho = [0,\ 0.178]$ when compared with the synthetic class without noise, discussed in section 7.16.2, because of the potential uniqueness of each occurrence (caused by the added random noise onto the alphabet pattern) in the feature vector. Hence, FuzzyART has more difficulties for merging non-overlapping classes (overgeneralize), which creates a smaller OG zone. This section has two curve types: A constant value, and a polynomial; (ii) the COI zone (starting at the dash-dotted red line and extending to the left until reaching $\rho = 0.682$ defined by the dash-dotted green line) is described by four curve types: A

polynomial and three exponentials; and (iii) the OS zone is described by a decaying exponential for values of the vigilance parameter $\rho = (0.682, 1]$. The appearance of the OS zone follows from the uniqueness provided by the random noise when added to the alphabet pattern. Hence, FuzzyART disassociates the COI into smaller sets causing in the end that a single occurrence is placed into a single class.

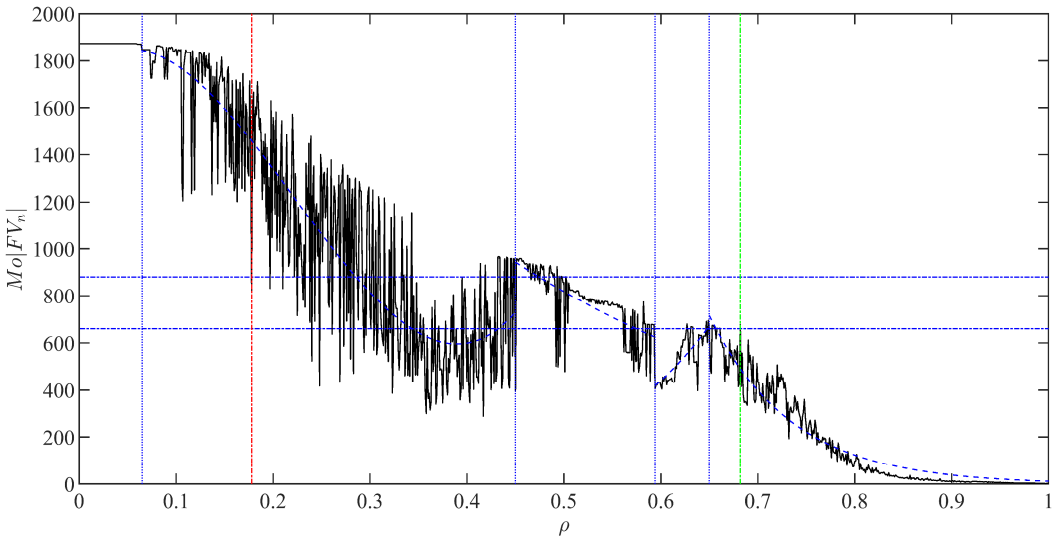


Fig. R.4. Unsupervised classification of feature vector instances FV_n (containing 42 real valued scalars describing the DDoS dataset) through FuzzyART with a vigilance parameter values for ρ spanning in the interval [0, 1].

The waveforms present in Fig. R.4 are generalized through a constant value, a third degree polynomial curve of the form $a\rho^3 + b\rho^2 + c\rho + d$, where a, b, c, and d are the coefficients, and three exponentials of the form $C_1 e^{\rho C_2}$, where C_1 and C_2 are exponential coefficients and e is the natural logarithm. The optimization curve fitting Trust-Region algorithm is used to find the coefficients characterizing each zone. The following equation represents this analysis in a compact form:

$$\text{Mo} \mid FV_n \models \begin{cases} 1860 & \\ \text{for } \rho = [0,\ 0.064] & \text{OG} \\ 6.545\times10^4\rho^3 - 4.342\times10^4\rho^2 + 4.02\times10^3\rho + 1.749\times10^3 & \\ \text{for } \rho = (0.064,\ 0.178] & \text{OG} \\ 6.545\times10^4\rho^3 - 4.342\times10^4\rho^2 + 4.02\times10^3\rho + 1.749\times10^3 & \\ \text{for } \rho = (0.178,\ 0.449] & \text{COI} \\ 3.432\times10^3(e^{-2.869\rho}) & \\ \text{for } \rho = (0.449,\ 0.593] & \text{COI} \\ 2.829(e^{8.417\rho}) & \\ \text{for } \rho = (0.593,\ 0.649] & \text{COI} \\ 1.486\times10^6(e^{-11.75\rho}) & \\ \text{for } \rho = (0.649,\ 0.682] & \text{COI} \\ 1.486\times10^6(e^{-11.75\rho}) & \\ \text{for } \rho = (0.682,\ 1] & \text{OE} \end{cases}$$

R.3 A 20 Percent Noisy Real Synthetic Class Condensing the First Five Letters of the Modern English Alphabet

In this case, the synthetic real valued feature vector is injected with 20 percent of Uniform random noise. Subsequently, this noisy feature vector is used to analyse the FuzzyART behaviour by changing the value of the vigilance parameter.

The synthetic feature vector instance is presented in Fig. R.5 displaying three distinct segments affected by the injection of 20 percent noise: (i) A section from frame 1 to 425 includes noisy real-valued vectors representing normal network/Internet traffic; (ii) a noisy real-valued pattern, from the frame 426 to the 1305, corresponding to the concatenated version of the first letters of the alphabet; and (iii) a final section from frame 1305 onwards with noisy real-valued vectors depicting normal network/Internet traffic.

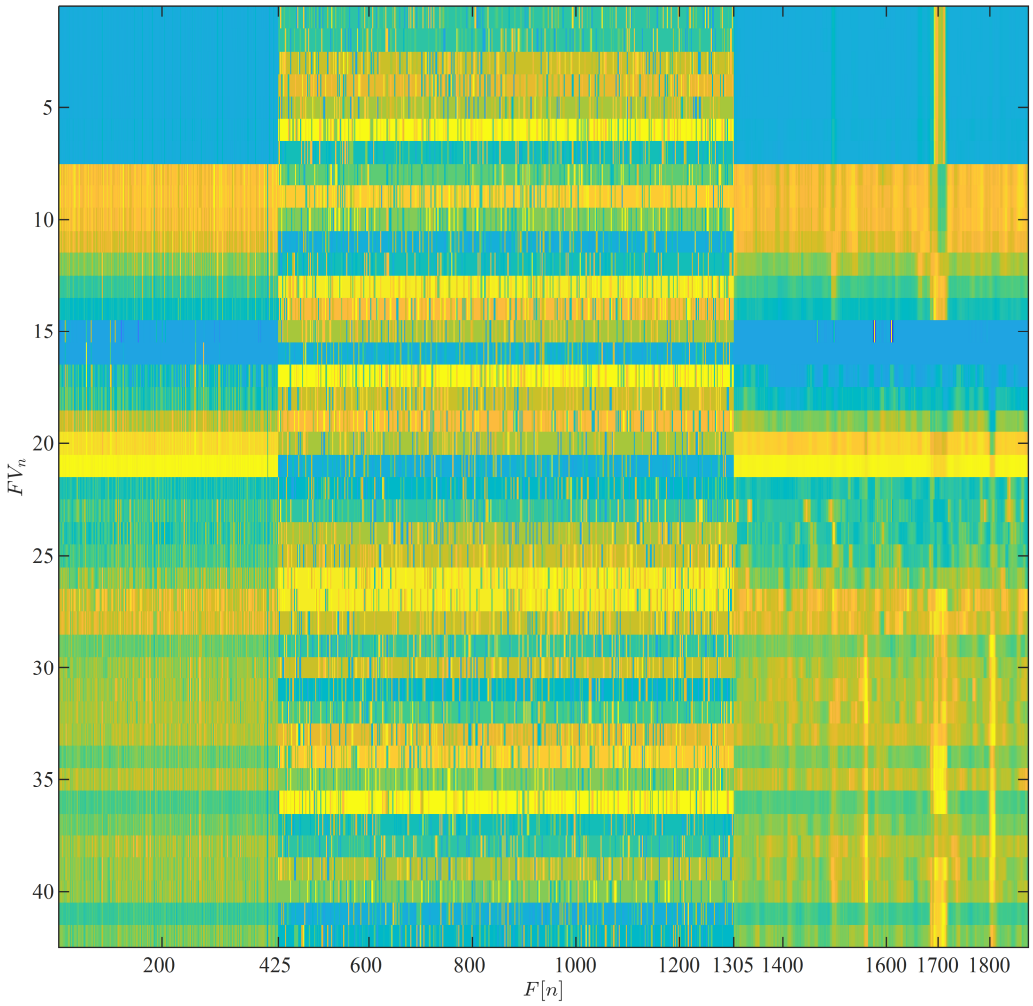


Fig. R.5. Synthetic dataset with 20 percent noisy occurrences of the alphabet first five letters (represented in matrices 7x5 reshaped into a 42 real valued vector) replacing the DNS DDoS attack. The rest of the occurrences are formed with normalized real valued vectors as part of the analysis applied to the original dataset. The H&R DDoS attack and the normal traffic is also normalized.

The values of the vigilance parameter ρ are changed in the interval [0, 1] in order to examine its behaviour when classifying the feature vectors shown in Fig. R.5 for each value of ρ. The matching results for this situation are shown in Fig. R.6.

The curve shown in Fig. R.6 has the subsequent behaviour according to the zones defined in the dataset including DDoS attacks: (i) The OG zone extends for values of ρ in the interval [0, 0.118]; (ii) the COI zone has values for ρ defined in the interval (0.118, 0.687]; and (iii) the OS zone is located in the interval (0.687, 1] for the class describing the occurrences of the

alphabet pattern.

Figure R.6 has three zones with distinct behaviours: (i) The OG zone (to the left of the dash-dotted red line) has a smaller interval $\rho = [0, 0.118]$ when compared with the synthetic class without noise, discussed above in section 7.16.2, because of the potential uniqueness of each occurrence (caused by the added random noise onto the alphabet pattern) in the feature vector. Hence, FuzzyART has more difficulties for merging non-overlapping classes (overgeneralize), which creates a smaller OG zone. This section has two curve types: A constant value, and a polynomial; (ii) the COI zone (starting at the dash-dotted red line and extending to the left until reaching $\rho = 0.687$ defined by the dash-dotted green line) is described by five curve types: A polynomial and four exponentials; and (iii) the OS zone is described by a decaying exponential for values of the vigilance parameter $\rho = (0.687, 1]$. The appearance of the OS zone obeys the uniqueness delivered by the random noise when added to the alphabet pattern. Hence, FuzzyART disassociates the COI into smaller sets causing in the end that a single occurrence is placed into a single class.

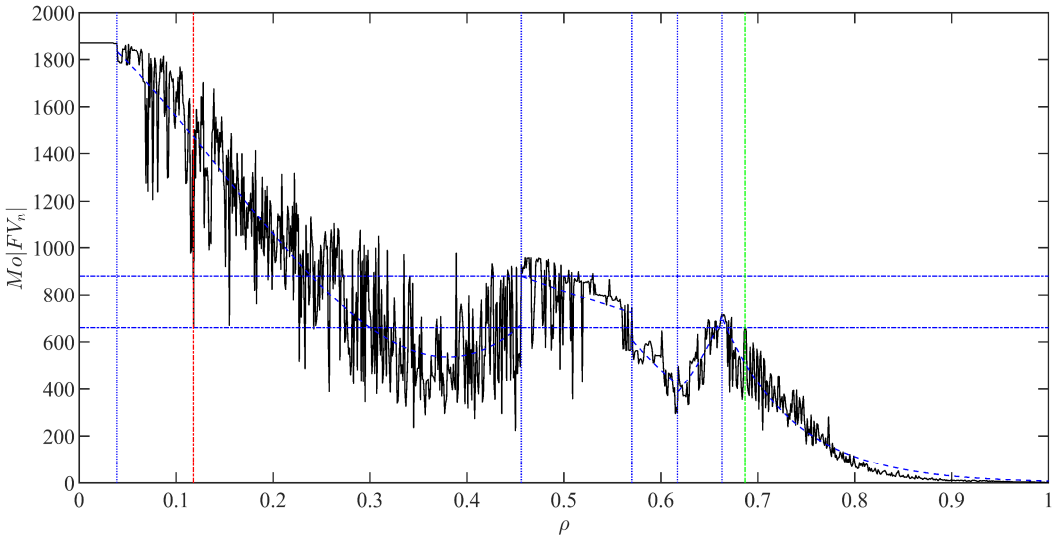


Fig. R.6. Unsupervised classification of feature vector instances FV_n (containing 42 real valued scalars describing the DDoS dataset) through FuzzyART with a vigilance parameter values for ρ spanning in the interval [0, 1].

The waveforms present in Fig. R.6 are generalized through a constant value, a third degree polynomial curve, and three exponentials of the form $C_1 e^{\rho C_2}$, where C_1 and C_2 are exponential coefficients and e is the natural logarithm. The optimization curve fitting Trust-Region algorithm is used to find the coefficients characterizing each zone. The following equation represents this analysis:

$$\text{Mo} \mid FV_n \models \begin{cases} 1875 & \\ \text{for } \rho = [0,\ 0.039] & \text{OG} \\ 3.064\times10^4\rho^3 - 1.337\times10^4\rho^2 - 3.131\times10^3\rho + 1.976\times10^3 & \\ \text{for } \rho = (0.039,\ 0.118] & \text{OG} \\ 3.064\times10^4\rho^3 - 1.337\times10^4\rho^2 - 3.131\times10^3\rho + 1.976\times10^3 & \\ \text{for } \rho = (0.118,\ 0.456] & \text{COI} \\ 1.930\times10^3(e^{-1.721\rho}) & \\ \text{for } \rho = (0.456,\ 0.570] & \text{COI} \\ 5.115\times10^4(e^{-7.773\rho}) & \\ \text{for } \rho = (0.570,\ 0.617] & \text{COI} \\ 0.109(e^{13.24\rho}) & \\ \text{for } \rho = (0.617,\ 0.663] & \text{COI} \\ 5.214\times10^6(e^{-13.14\rho}) & \\ \text{for } \rho = (0.663,\ 0.687] & \text{COI} \\ 5.214\times10^6(e^{-13.14\rho}) & \\ \text{for } \rho = (0.687,\ 1] & \text{OE} \end{cases}$$

R.4 A 30 Percent Noisy Real Synthetic Class Condensing the First Five Letters of the Modern English Alphabet

The synthetic real valued feature vector in this occasion is injected with 30 percent of Uniform random noise. Afterwards, the noisy feature vector supports the analysis of the FuzzyART behaviour through the change of the vigilance parameter value.

The instance of the synthetic feature vector is shown in Fig. R.7 displaying three different segments affected by the injection of 30 percent noise: (i) A section from frame 1 to 425 includes noisy real-valued vectors representing normal network/Internet traffic; (ii) a noisy real-valued pattern, from the frame 426 to the 1305, corresponding to the concatenated version of the first letters of the alphabet; and (iii) a final section from frame 1305 onwards with noisy real-valued vectors depicting normal network/Internet traffic.

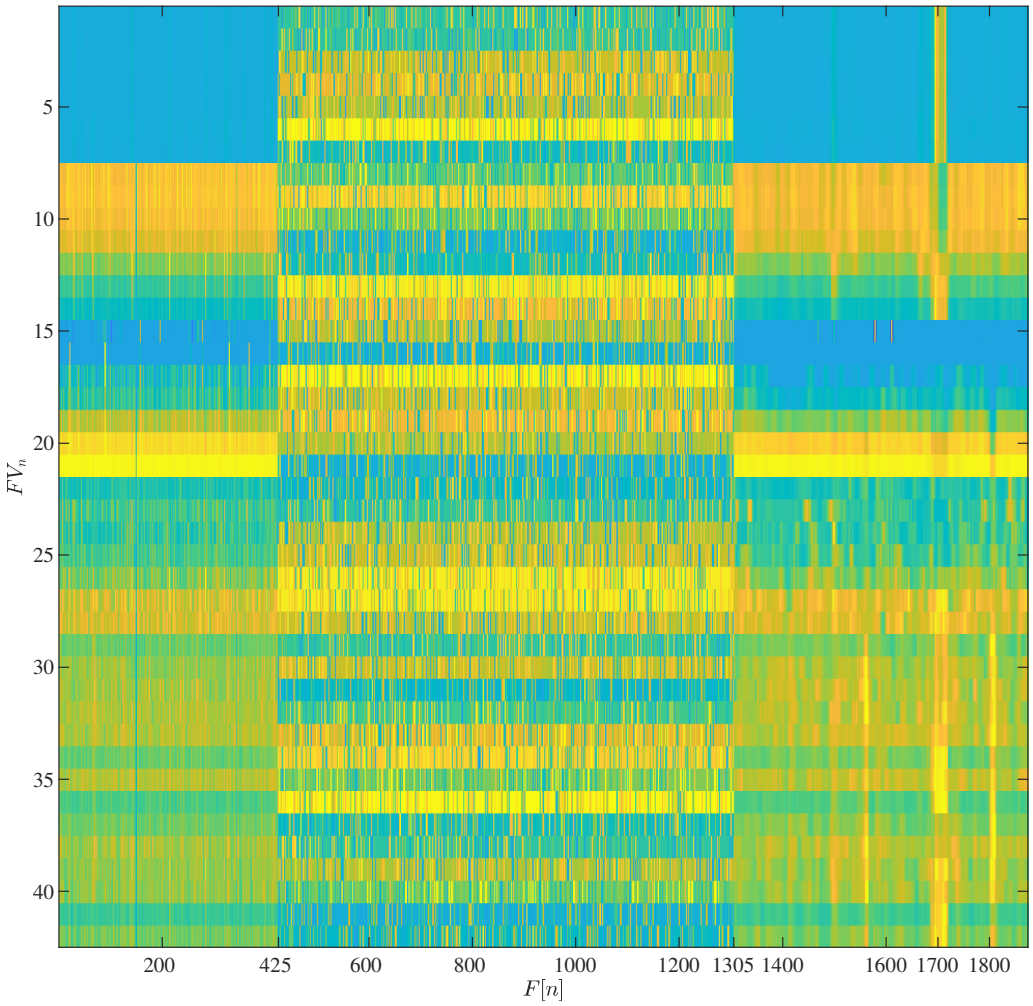


Fig. R.7. Synthetic dataset with 30 percent noisy occurrences of the alphabet first five letters (represented in matrices 7x5 reshaped into a 42 real valued vector) replacing the DNS DDoS attack. The rest of the occurrences are formed with normalized real valued vectors as part of the analysis applied to the original dataset. The H&R DDoS attack and the normal traffic is also normalized.

The vigilance parameter ρ value is changed in the interval $[0, 1]$ for inspecting FuzzyART's behaviour when classifying the feature vectors displayed in Fig. R.7. This is done for each value of ρ. The corresponding results of this process are shown in Fig. R.8.

Corresponding to the zones defined in the dataset involving the DDoS attacks, the waveform exhibited in Fig. R.8 has the subsequent behaviour for the class describing the occurrences of the alphabet pattern: (i) The OG zone spreads for values of ρ in the interval $[0, 0.106]$; (ii) the COI zone is described in the interval $(0.106, 0.679]$ for values of ρ; and

(iii) the OS zone is positioned in the interval $(0.679, 1]$.

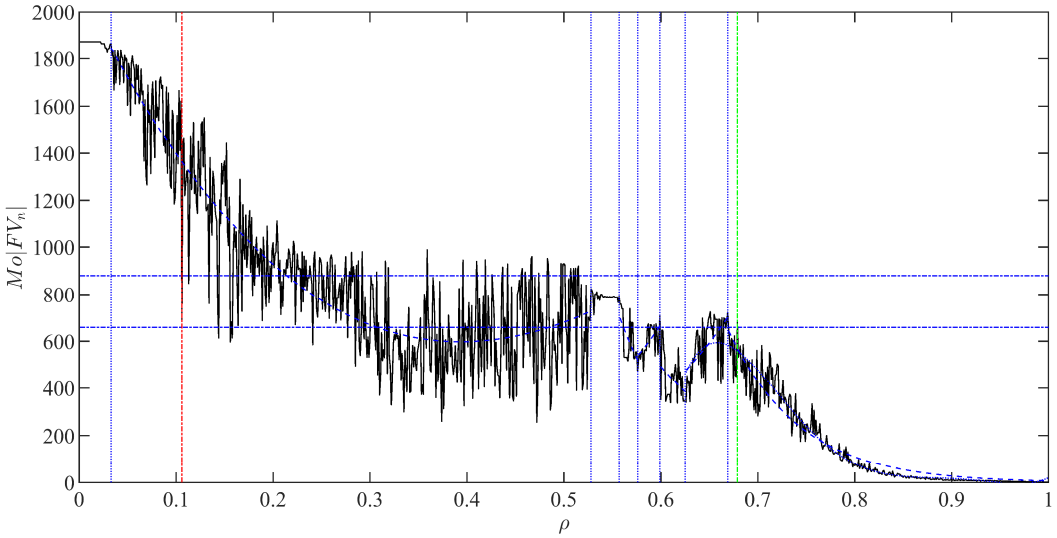


Fig. R.8. Unsupervised classification of feature vector instances FV_n (containing 42 real valued scalars describing the DDoS dataset) through FuzzyART with a vigilance parameter values for ρ spanning in the interval [0, 1].

Figure R.8 shows three zones that perform distinctly: (i) The OG zone, positioned left to the dash-dotted red line, also has a smaller interval $[0, 0.106]$ for this case when compared with the noiseless synthetic class, presented above in section 7.16.2, because of the uniqueness of each occurrence increases. This increase is due to the higher amount of added random noise onto the alphabet pattern in the feature vector when compared to the previous two cases. This creates a higher degree of difficulty for FuzzyART to merge non-overlapping classes, overgeneralize, which generates a smaller OG zone. This zone has two curve types: A constant value, and a polynomial; (ii) the COI zone, beginning at the dash-dotted red line and spanning to the left until reaching $\rho = 0.679$ defined by the dash-dotted green line, is described by five curve types: A polynomial, a constant value, and five exponentials; and (iii) the OS zone is defined by a decaying exponential for the interval $(0.106, 0.679]$ in the vigilance parameter. The size of the OS zone follows from the uniqueness that the random noise incorporates to the alphabet pattern. Hence, FuzzyART granulates the COI into smaller sets causing that a single occurrence is classified into a single class when $\rho \approx 1$.

Figure R.8 shows distinct waveforms: Two constant values, a third degree polynomial curve, and five exponentials of the form $ae^{\rho b}$, where a and b are exponential coefficients and e is the natural logarithm. The Trust-Region algorithm is used as optimization curve fitting to find the coefficients characterizing each nonlinear waveforms. The following equation encloses the scrutiny just described:

$$\mathrm{Mo}\,|\,FV_n\,|= \begin{cases} \sim 1874 \quad \text{for } \rho = [0,\ 0.033] & \text{OG} \\ -4.796\times10^{3}\rho^{3} + 1.354\times10^{4}\rho^{2} - 8.428\times10^{3}\rho + 2.11\times10^{3} \quad \text{for } \rho = (0.033,\ 0.106] & \text{OG} \\ -4.796\times10^{3}\rho^{3} + 1.354\times10^{4}\rho^{2} - 8.428\times10^{3}\rho + 2.11\times10^{3} \quad \text{for } \rho = (0.106,\ 0.528] & \text{COI} \\ \sim 790 \quad \text{for } \rho = (0.528,\ 0.557] & \text{COI} \\ 6.74\times10^{6}(e^{-16.44\rho}) \quad \text{for } \rho = (0.557,\ 0.576] & \text{COI} \\ 2.475(e^{9.34\rho}) \quad \text{for } \rho = (0.576,\ 0.599] & \text{COI} \\ 1.777\times10^{5}(e^{9.019\rho}) \quad \text{for } \rho = (0.599,\ 0.625] & \text{COI} \\ 1.687(e^{-0.1912\rho}) \quad \text{for } \rho = (0.625,\ 0.669] & \text{COI} \\ 5.85\times10^{6}(e^{-13.6\rho}) \quad \text{for } \rho = (0.669,\ 0.679] & \text{COI} \\ 5.85\times10^{6}(e^{-13.6\rho}) \quad \text{for } \rho = (0.679,\ 1] & \text{OE} \end{cases}$$