Cognitive Discriminative Feature Selection Using Variance Fractal Dimension for the Detection of Cyber Attacks

By

Samilat Kaiser

The thesis submitted to the faculty of graduate studies

of

The University of Manitoba

In partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

University of Manitoba, Winnipeg, MB, Canada

Abstract

The dimensions of data have increased significantly with the abundance of the data that we share. This high-dimensional data results in redundant and irrelevant features that creates challenges to existing machine learning algorithms. Redundant and irrelevant features slow down the training and testing process as a result affecting the performance and run time of a learning algorithm. Conventional feature selection methods showcase great potential to select important features. However, machine enabled feature selection models are unable to solve complex analysis in absence of human cognitive aspect to it.

This thesis addresses the challenge concerned with reducing the data dimensions for machine learning algorithms from cognitive aspect. The reduction is carried out by a variance fractal-based complexity analysis to select a reduced set of features of a cyber attack dataset for each attack type that are discriminative in nature. Furthermore, integrated artificial neural network were created with inputs that comprised of the reduced features selected through the complexity analysis. A performance comparison is also provided using our proposed methodology with resulting minimized dataset features of each attack types with non-minimized dataset. The comparative analysis shows that the resultant discriminative features derived from our proposed method not only consume less resource but also speed up the training and testing process while maintaining good detection rates.

Acknowledgements

My sincere gratitude to everyone who has encouraged and supported me directly or indirectly during the entire journey of my MSc. program.

My utmost gratitude goes to my thesis advisor, Professor Dr. Ken Ferens. His spontaneous encouragement, guidance and invaluable time has been instrumental for the completion of my graduate studies. Even in the most distressed times, the patience and support that he has shown towards me, I shall remain forever grateful for that.

I am thankful to my thesis committee members Prof. Robert D. McLeod and Prof. Noman Mohammed for their interests and valuable time for evaluating my research work.

I am extremely grateful to my husband Kaiser Nahiyan for always encouraging me, believing in me and standing strong by my side through every thick and thin. I truly appreciate his sincere support during the entire path of my research work which gave me the confidence to never give up.

I could not have accomplished this without the prayers and support of my beloved family and my in-laws. I am eternally indebted for the unconditional love, sacrifices and support of my beloved parents and my loving sister throughout my life. I am thankful for the guidance from my sister Tashniba Kaiser to pursue my graduate studies from this esteemed university.

I am thankful to the University of Manitoba, Faculty of Graduate Studies and my Electrical and Computer Engineering Department for the opportunity and support to pursue my graduate program.

I am forever obliged to almighty Allah (SWT) for bestowing me with the strength to accomplish my thesis.

Dedication

“I have not failed. I've just found 10,000 ways that won't work.”

— Thomas A. Edison

“Strive not to be a success, but rather to be of value.”

— Albert Einstein

I dedicate this work to everyone who struggle in life to achieve something. No matter what happens, one should never stop believing in oneself. I would also like to dedicate this work to my loving parents, my mother- Nilufar Farooqui and my father- Mahmud Matin Md. Kaiser.

Table of Contents

List of Acronyms

IDS	Intrusion Detection System
ML	Machine Learning
ANN	Artificial Neural Network
FD	Fractal Dimension
VFD	Variance Fractal Dimension
TPR	True Positive Rate
TNR	True Negative Rate
FPR	False Positive Rate
FNR	False Negative Rate
ICMP	Internet Control Message Protocol
SYN	Synchronize
GDP	Growth Driven Profit
IoT	Internet of Things
fBm	Fractional Brownian Motion
MLP	Multi-Layer Perception

List of Figures

List of Tables

Chapter 1: Introduction

Technology is an integral aspect of the everyday life which has progressed significantly over the past few decades. Any mode of technology be it email correspondence, monetary transaction, networking or personal and private information of an individual or an organization, we rely on technology. Hence, any compromise to these means of communication can impose enormous threat to individual, government or business. However, with more advancement of accessibility and usage of technology comes with more challenge of securing and preventing the cyber platform. According to an analysis carried out by McAfee and experts from the Center for Strategic and International Studies, the global cost of cybercrime for 2017 added up to around $600 billion which is equivalent to a 0.8 percent of global GDP [1]. This number mounts up every year. Experts are suggesting this number to be $6 trillion per year by 2021 [2]. Cyber attacks are not only the fastest growing crime, but they are also increasing in terms of threat patterns, dynamism and sophistication. Although Intrusion detection systems (IDS) or machine learning (ML) strategies have much evolved for threat detection nonetheless the task remains challenging in due to dynamic evolution of high dimensional data.

1.1 Motivation

This section describes the motivation behind doing the thesis. At first the problem with high dimensional data is described and followed by, the importance of reducing number of features used in ML algorithm is described. Furthermore, a discussion is provided on how conventional feature selection models are not effective in terms of cognitive analysis. Finally, this section concludes by stating the motivation to carry out the research.

Over the past decade data posed perhaps the single greatest challenge in the field of cybersecurity. The global internet population growth went more than double from the year 2012 to 2018, from 2 billion to 4.3 billion. For every minute in 2019, 188 million emails were sent, 4 million google searches were conducted. The world uses a staggering 4,416,720 gigabytes of internet data per minute [3]. On the other hand, the frequency and damage caused by novel cyber-attacks are increasing with time. Since the cyber-attacks occur simultaneously with usual cyber operations, this adds onto the high volume of data created per day. Such large-scale data poses a challenge in terms of the four data quality dimensions namely; volume, variety, velocity and veracity [4]. Correspondingly, the data patterns are also evolving due to its dynamic nature, distinct characteristics of zero-day attack and high complexity.

Traditionally dataset contains many features that carries information of the system activities. However, it also contains redundant and irrelevant features that are two primary factors which result in large volume and high dimensional data. High dimensional data poses several downsides. To begin with, redundant and irrelevant features consumes more computational resource. Then, such volume of data negatively affects the performance of machine learning algorithm. Also, they increase the computational time for the learning algorithm. Moreover, such large volume of data will require large storage capacity. Finally, high dimensional sparse data affects the accuracy of machine learning algorithm due to curse of dimensionality [5]. All these reasons validate the need of dimensionality reduction which can be attained by removing the features that possess no discriminating power. If we can select the discriminative features from the dataset, the state-of-art machine learning (ML) algorithm is expected to achieve optimal result in machine learning model.

Conventionally, machine learning, mathematical and statistical analysis are applied for feature selection. However, in an ever-evolving threat landscape, these models still pose a threat when it comes to zero-day attack. Hence processing of features or attributes still require manual human involvement to identify the desired features that are discriminative, independent and informative.

The primary intension of this research is to incorporate cognition in feature selection for ML model to detect cyber-attacks. That is to say, the work was carried out as an experimental investigation to apply cognitive feature selection method that will mimic the human intelligence and cognitive analysis approach to identify the discriminative features of cyber attacks.

1.2 Thesis Statement

Typically, experienced human cybersecurity administrators can look at raw features of packet trace in a network and determine which features are more important than others. Ideally, they select the features based on their human intelligence, domain knowledge and expertise which corresponds to the cognitive ability. However, it is believed that this human cognitive process is in part due to complexity analysis. This work hence focuses on depicting this cognitive aspect by using variance fractal dimension (VFD) as a tool to complexity analysis to identify the discriminative features of an internet dataset. The resultant discriminative features enhance the classification performance of the artificial neural network in terms of detection accuracy and computational time. Formally our research statement can be stated as follows:

Can we design or devise a computer algorithm that would simulate the cognitive process of selecting important features?

For designing computer algorithms to simulate human cognitive process of looking at the data and determining which features are more important, we drill down our research problem to below research questions:

1. Can complexity analysis be used to determine discriminative features for the detection of cyber attacks?
2. Can the resulting minimized data set perform as well as non-minimized data set?

1.3 Outline of the thesis

This thesis is divided into the following chapters:

Chapter 1 introduces the concept of this thesis, providing the motivation behind research in the field of cybersecurity, specifically in the area of feature selection. This chapter is concluded with the formal thesis statement and associated research questions.

Chapter 2 provides a literature review and background studies on two major areas; a. contemporary feature selection techniques in dimensionality reduction and b. the concept of cognition, cognitive intelligence. A comparison on different feature selection approaches is also provided in this chapter. The concept of complexity in the field of cognitive analysis, how fractals and variance fractal can be used as a tool for complexity measures are also discussed in this chapter. This chapter provides the idea how our research is directed towards cognition through complexity analysis.

Chapter 3 describes our proposed discriminative feature selection algorithm. A formal flow chart for the proposed workflow is provided in this chapter and each phase of the

proposed algorithm is discussed elaborately. This chapter also includes an overview on Variance Fractal Dimension theory and Artificial Neural Network architecture.

Chapter 4 provides detailed description of the dataset that was used for the experiment. This chapter also provides brief discussion on each type of attacks that are present in the dataset.

Chapter 5 presents our research experiment and results for the proposed cognitive discriminative feature selection algorithm. This chapter also provides information of experimental setup and verified result and discussion on the result for the proposed algorithm for the dataset described in chapter 4.

Chapter 6 gives a summary of the goal of the research, the findings and concluding remarks. We also describe the scope of future works and improvements that can be done in the field of cognitive feature selection

Chapter 2: Background Studies and Literature Search

This chapter reviews the advancement that has been made in the field of feature selections in intrusion detection system (IDS) and other field of studies. Also, existing feature selection approaches, their challenges are also discussed. Furthermore, discussion on application of variance fractal dimension and other cognitive feature selection approaches are also provided in this chapter.

2.1 Dimensionality reduction

The purpose of any security mechanism of an Intrusion Detection System (IDS) is to inspect the traffic to detect any anomaly or suspicious behavior. However, a data stream with high and large dimensionality makes the task a challenging one for any IDS. The complexity and volume of the datasets are ever growing which only makes the machine learning techniques even more complex to extract meaningful information from a dataset containing large number of meaningless data. The high dimensional data that caries redundant and irrelevant features degrade the overall performance of a machine learning algorithm by exhausting and misdirecting the learning mechanism. High dimensional data creates challenge for classification algorithms with its high computational cost and memory usage. Hence, researchers have adopted the dimensionality reduction as a pre-processing step to help the detection model to improve their performance that is more comprehensive considering time, complexity and detection accuracy. Dimensionality reduction can be defined as mapping high dimensional data points to a lower dimension while preserving certain core properties. Simply put, Dimensionality reduction is basically the process of reducing the dimension of a feature set. Dimensionality reduction can be done by two

techniques: 1. Feature extraction and 2. Feature selection [6]. However, both are ubiquitous to dimensionality reduction. Which technique to be performed is largely context dependent. Our scope of work for this research is however targeted to feature selection hence only feature selection methods are being considered.

2.1.1 Feature Extraction

Feature extraction can be described as the technique of finding informative and compact set of features in a data driven way [7]. It is generally linked to generation of new features from raw dataset [8]. One of the advantages of feature extraction is, it can reduce the feature space without having to lose a lot of data of the primary feature space. Out of many feature extractions approaches, Principal Component Analysis (PCA) is one of the most widely used approach.

2.1.2 Feature Selection

A "feature" or "attribute" or "variable" refers to an aspect of the data. In machine learning, a feature is a measurable property of an object being observed. Feature selection (also known as subset selection) refers to the process of selecting subset of features from the data for machine learning algorithm that can describe the specified problem without dropping of performance [9]. It is achieved by removing redundant and irrelevant features. It is basically a data pre-processing step that helps the ML algorithm to reduce computational complexity and improve its performance by improves accuracy and reduces computational complexity by removing redundant and irrelevant features [10].

Ideally, there are three desired characteristics of a feature. A feature should be i. Independent, ii. Discriminative and iii. Informative.

- The feature should be independent or uncorrelated. Even though this is an ideal scenario but in real scenario, features do show some weak independence.
- Feature should be discriminative; it should be unique that represent its distinct nature and does not overlap with.
- Feature should be informative; it should adequately describe and represent the information of the system in consideration.

2.1.2.1 Advantages of Feature Selection

Even though selecting relevant and informative features are the primary focus of feature selection however, it offers other advantages that motivates to apply feature selection techniques:

- It helps reduce the complexity of ML model that is easier to interpret.
- It reduces the dimensionality of the feature space that helps limit the storage requirements
- It helps machine learning algorithm to train faster.
- It removes the redundant, irrelevant or noisy data.
- It helps improve the data quality.
- It improves the performance of machine learning model.

2.1.2.2 Feature Selection Methods

Feature selection methods can be generally divided in two methods: individual evaluation or feature ranking and feature subset selection. Feature ranking is the method of ranking individual features according to its relevance. Fast and effective performance is some of the benefits of feature ranking method. However, it is considered being effective for the scenario where the number of features is large number of training samples are small in comparison. Also individually relevant features may have redundancies because of the underlying feature independence [7]. Feature subset selection on the other hand utilize search strategy to produce subset of features to evaluate the produced subset of features [11]. Unlike feature ranking method, feature subset selection method can overcome the redundancy of feature relevance [12].

There are other methods of feature selection which are evaluation-based feature selection models namely; i. Filter model, ii. Wrapper model, iii. Embedded model [13].

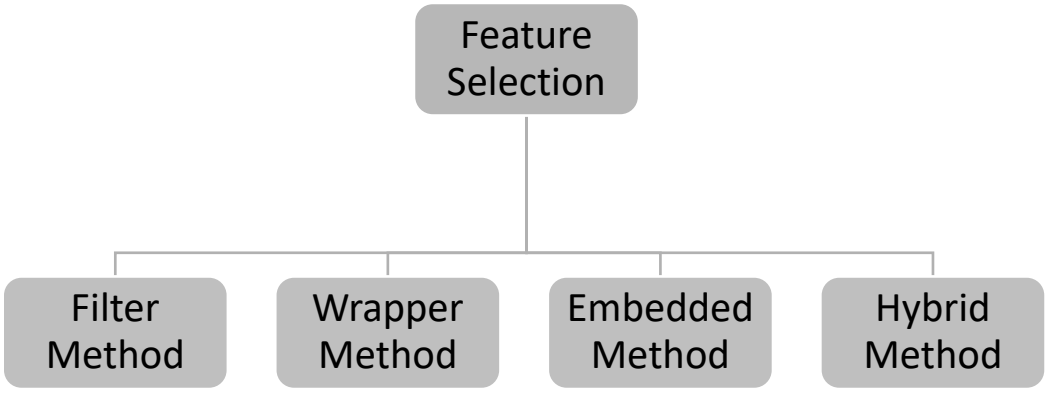

Figure 1: Evaluation -based feature selection methods

I. Filter Model:

The filter model performs independently of classifier algorithm while relying on the general characteristics of the training data [14]. The filter method applies statistical characteristics of the data. The primary advantage of filter method is its unbiased nature as it is independent of classification algorithm. Moreover, the simple design structure helps the filter method to perform faster. Furthermore, compared to wrapper and embedded model, filter models are computationally cost effective. However, filter method often fail to provide best feature subset compared to wrapper method.

In [15] authors have introduced pair-wise correlation analysis based filter method for feature selection. The correlation between continuous and discrete feature was measured by removing weakly relevant, irrelevant and relevant but redundant features. In [16], authors developed a novel multivariate filter method as an extension to the relative discrimination criterion (RDC) for text classification. The authors were able to remove redundant and irrelevant features by using correlation metric that gives maximal relevancy and minimal redundancy. In [17], authors have utilized mutual information based algorithm for selecting best feature set for the classification. The features are selected analytically and further were evaluated for intrusion detection of Kyoto 2006+, KDD Cup 99, and NSL-KDD datasets.

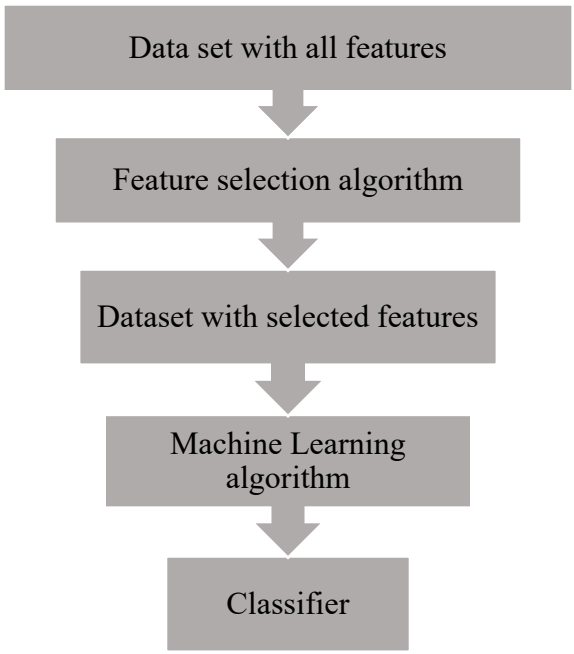

Figure 2: Feature selection- Filter Method

Filter model can be further classified into two methods; univariate and multivariate. Univariate filter method can identify feature relevance independently. In other words, they can identify feature irrelevance using specific criterion however they are unable to eliminate redundant features.

II. Wrapper Model:

The algorithm of a wrapper method uses feature subset that train a learning model. The decision of adding or removing feature from the subset is inferred from the previous model. Wrapper algorithm uses classifiers as a selection process to assess the quality of a given feature subset [18].

Authors in [19] proposed an algorithm that alternates between filter ranking construction and wrapper model iteratively. The model seems promising as it analyzes few blocks of variables that decrease the number of wrapper evaluations drastically. In [20], authors have utilized genetic algorithm- wrapper feature selection to develop an ensemble of classifier for real time scheduling. The model was applied on a case study of flexible manufacturing system. Some researchers tried to help minimize the computational time for wrapper method. In [21], authors have experimented the wrapper model for 8 microarray dataset by applying an embedded K-Nearest-Neighbor (KNN) classifier. The classification performance and cost effectiveness were demonstrated by utilizing both theoretical and experimental results.

In [22], researchers have used wrapper-based feature selection technique based on Improved Ant Colony Optimization. The experiment was able to discriminate bipolar disorder and major depressive disorder subjects by reducing the feature from 48 to 22 while achieving an overall classification accuracy of 80.19%. In [23] the authors have applied two Deep Convolutional Neural Networks to extract the features and later used wrapper model as a feature selection tool to identify discriminative and important features. The model used support vector machine as a classifier. Performance of the proposed deep learning- based malware detection model was compared alongside different classifier model.

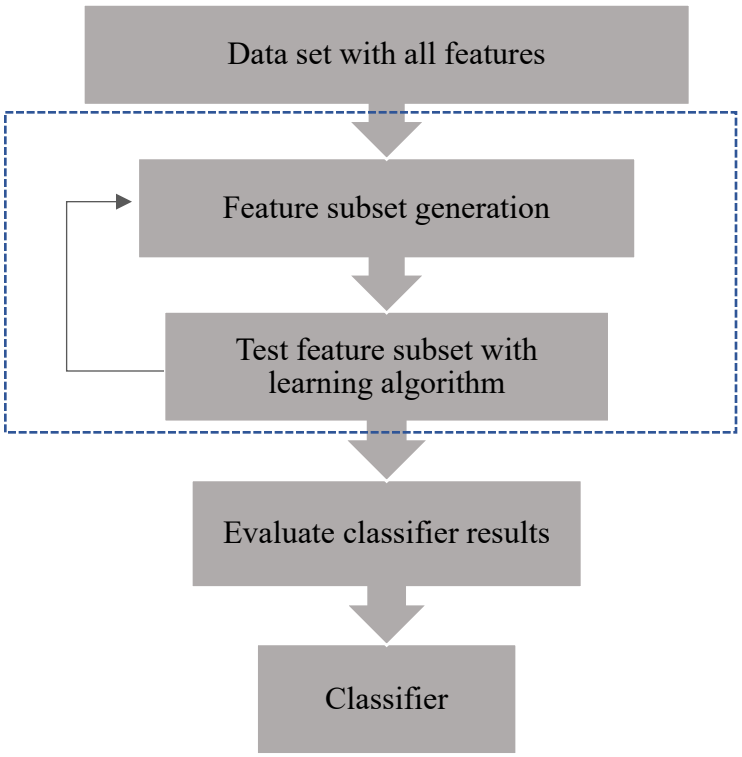

Figure 3: Feature selection- Wrapper Method

While wrapper methods are classification algorithm dependent, it can reach better recognition rate compared to filter methods hence more accurate. However, wrapper model is known to be computationally complex due to employing supervised learning algorithm and expensive compared to filter or embedded model.

III. Embedded Model:

The embedded model for feature selection is directly linked to classification stage training process [24]. The feature selection in embedded model is done in single step.

An embedded model was used for selection method to a variational relevance based classification model construction in [25]. The comparative analysis of the experiment was done using real-world benchmark datasets and industrial dataset. The authors used relevance vector machine model based on an automatic relevance determination kernel using variational inference for regression in hierarchical prior over the kernel parameters setting. In [26], the researchers presented a novel embedded feature selection method based on probabilistic modeling, defining the multivariate Bernoulli distribution of the binary vector and using the vector for feature selection. The approach optimized the parameters of the distribution during the model training introduced initially in [27], where the researchers proposed a general framework for dynamically optimizing the network structures and the connection weights simultaneously.

Two very popular methods used in embedded feature selection are RIGDE regression and LASSO. These methods use penalization functions that are inbuilt and can reduce the overfitting problem. The application of L1-norm regularization can be seen in some embedded feature selection based research [28]. The researcher in [28] have used Regularized Generalized Eigenvalue Classifier or ReGEC on support vector machine in a dataset from Diagnostic Wisconsin Breast Cancer Database.

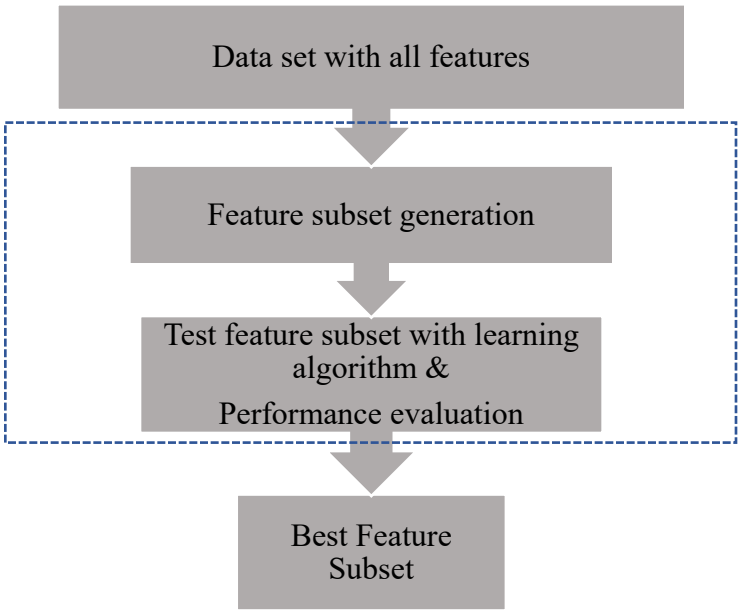

Figure 4: Feature selection- Embedded Method

One of the advantages of embedded method is the feature selection and classification are done in single step. However, the performance of the model can face challenge for dataset with more irrelevant features. Aside, embedded model performs slower compared to filter methods.

IV. Hybrid Model:

Another feature selection model is hybrid model that merges the filter and wrapper-based methods. Authors in [29] have used a hybrid model to select optimized feature set of protein-protein interaction pairs in. The mRMR filter and a KNNs–wrapper was used in the hybrid feature selection model. In [30] the authors have used filter-wrapper based hybrid approach where the filter part was used to select a reduced subset whereas wrapper

part could adjust and further fine tune the filter. The authors used the model to forecast the price and load for electrical power systems. In [31] the researchers have used Laplacian Score ranking and Calinski-Harabasz index for a spectral feature selection framework that was used for a hybrid feature selection for clustering.

Through our literature search, we could examine that the hybrid methods were chosen by many researchers due to the scope of utilizing more than one selection methods for its unique advantage and the fact that the model is cost effective.

Many researchers have carried out other significant techniques as feature selection models. In [32] [33], the authors have identified and analyzed different features that can distinguish between phishing website and legitimate ones. While [32] have developed a computerized tool to automatically extract features, [33] have used Kaiser-Meyer-Olkin test to select features. Feature clustering was used as a selection process using minimal-relevant-redundancy criterion function in [34]. [35] Used Forward Selection Ranking or Backward Elimination Ranking to determine the feature correlation and support vector machine function was used for feature ranking. The model was able to select the features set independently of the classifier used. In [36]the authors have used auto encoder and principle component analysis (PCA) to select low-dimensional features of the CICIDS2017 network intrusion dataset. Later Linear Discriminant Analysis, Random Forest, Bayesian Network and Quadratic Discriminant Analysis classifiers were used to design an IDS.

From the literature we can observe machine learning, mathematical and statistical analysis are applied for feature selection however, the threat landscape is ever evolving and still poses a threat when it comes to zero-day attack. Hence processing of features or

attributes still require manual human involvement to identify the desired features that are discriminative, independent and informative.

2.2 Cognitive computing and cognitive intelligence in cyber security:

The rapid explosion of unstructured and large volume of data over the past decade have set a challenge for the traditional computer programs. Computer systems not only require fast calculation on large volume of data but also require exploring the data to find correlation and new context in it which rely on human analyzing capabilities. This implies that to analyze and explore the data, it is impartial that the existing computer systems need to incorporate the perspective of human intelligence. The idea of cognitive computing or cognitive system aims at replicating the human analytical capability [37].

The word cognition refers to the cerebral process of acquiring knowledge or understanding by the means of thoughts, experiences and analytical thinking. Cognitive skills are essential for every task carried out by human be it tasks related to general senses or other tasks like, motor skills, learning ability, speaking or language skills or even social skills. Cognitive systems are machines that are inspired by the human brain and it can be said that cognitive analysis replicates the human's analysis approach. It is important for the machines to cognize and understand the human-like ability to understand who to world around humans' function [38]. However, in today's era cognitive computing is not only about replicating the way human brain works but also to exceed the capabilities or limitations of human analytics [39]. Hence, cognitive analysis or cognitive computing is the combination of artificial/ machine intelligence and human intelligence.

Cognitive computing now far exceeds the conventional machine learning. A cognitive system embedded into IoT can assist in critical recommendations and decision making [40]. Cognitive radio network can be established by integrating cognitive technologies in information communication systems [41]. The application of this dynamic field of cognitive computing ranges from healthcare, automated transportation, neuroscience to speech recognition, image processing, IoTs, information security and so on.

In the era of big data the computational efficiency of machines are irreversible obvious with the constant increase of information [42]. The big data features are represented by 5v, namely, volume, velocity, variety and veracity. The strength of human analytics lies on the ability to utilize deep thinking, reason with facts, and solving complex problems. However, human capabilities are limited when it comes to big volume of unstructured data. It is believed that cognitive systems can bring meaningful information from 80% of world's data that are considered unstructured by human scientists [43]. This opens the door to the visibility of unpredictable information by giving us the opportunity to harness machine and human intelligence collaboratively.

The primary goal of this research is to apply cognition to the traditional machine learning algorithms in the field of intrusion detection. This thesis utilizes cognitive context along with machine learning model and mathematical tools for cyber security dataset. Like any other field, the application of cognitive analysis in cyber security domain is a much aspired ask of time. Hence, we applied complexity-based feature selection to incorporate cognition in this research work. The following sub section discusses in detail the complexity measure that we used as a tool for complexity analysis.

2.2.1 Complexity Analysis

One of the philosophies used in cognitive analysis is the concept of complexity where complexity can be interpreted as components that are dependent in such a way that it is impossible to distinct the influence of one from another where decomposition of the system into independent components would destroy the whole system [44]. The cognitive complexity measure hence should be based on information theoretic, or computational approaches. Application of fractal dimensions can model the complexity of an object or component where higher fractal dimension would refer to more complexity. Hence, in this experiment we have chosen variance fractal dimension (VFD) to measure the feature complexity.

2.2.1.1 Fractals and Variance Fractal Dimension

Fractals are objects that showcase a phenomenon of normal or irregular behaviour of the object that is scaled down to a measurement that is achieved by fragmenting it to smaller scale. Fractal dimensions have been proven to be an effective computational method [45]. Fractal dimensions are significant in the context of fractals since it measures the complexity of fractals. There are different fractal dimension approaches namely, the box counting method, the information fractal dimension, variance fractal dimension and others. This study uses variance fractal dimension, one of the many fractal dimensions to identify discriminative features from cyber attack data. Researches have shown that, attributes can be selected cognitively using fractal based multiscale analysis [45]. This work used box counting fractal dimension algorithm and was able to get discriminatory sensitivity value.

2.2.1.2 Advantages of using Variance Fractal Dimension

Researches show some promising result with variance fractal as a tool for complexity measure. In [46], variance fractal dimension was used in signal processing for node localization. In [47], the researchers utilized variance fractal dimension as a complexity measure for classifying network intrusion. They applied cognitive clustering where variance fractal dimension was used to accurately identify the cluster containing majority of attack traffic.

This work focuses on identifying features that show significant difference in terms of complexity in normal and attack data. Here, the study uses variance fractal analysis (VFD) as a tool to measure the complexity of normal and attack samples with respect to each feature. One of the benefits of variance-based fractal analysis is, they are unaffected by noise. Additionally, VFD is it can be used for real-time application. Further, it helps to provide unique identification for each class that can be adopted in both data segmentation and feature extraction as it is able to emphasize the underlying complexity.

Chapter 3: Description of Proposed Feature Selection Algorithm

Our approach to variance fractal dimension (VFD) feature selection for the detection of cyberattacks consists of seven consecutive steps. A. Data preprocessing B. Obtaining feature vector, C. Creating feature vector subset, D. VFD calculation, E. Identifying discriminative features, F. Testing the discriminative features, G. Results. The workflow for proposed methodology is depicted in figure 5.

3.1 Dataset preprocessing:

To make the data suitable for the machine learning model, data needs to be preprocessed.

i. Removing missing data:

Missing data refers to an attribute's field value that is missing due to the data was not captured or improperly captured. These fields were removed as they do not contribute much.

i. Encoding of data:

Data contains both numeric and non-numeric attributes. The non-numeric attributes were converted to numbers so that mathematical calculations can be performed for the machine learning model.

ii. Normalization of Data:

Data samples were normalized following the feature scaling process. During data scaling we have transformed the data so that it fits within a specific scale

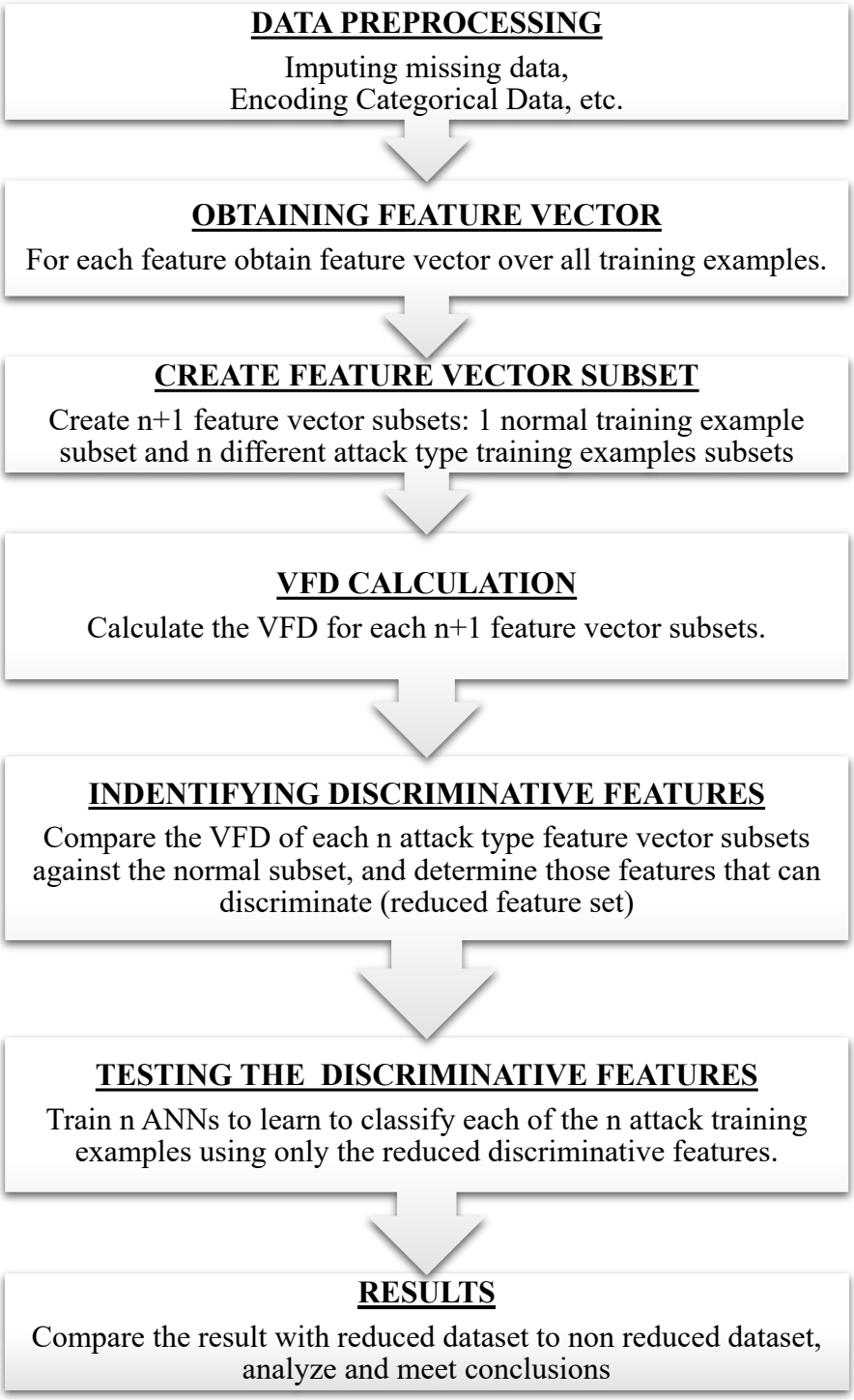

Figure 5: Proposed framework for variance fractal dimension feature selection method.

3.2 Obtaining feature vector, creating feature vector subset and VFD calculation:

In this step VFD value of all 47 features were calculated for normal and individual attack types. First, the normal samples and 9 individual attack samples were separated from the dataset and 10 separate subsets of datasets were created. We have calculated variance fractal dimension of each feature for n=8 different attack dataset [Attack_1, Attack_2, ….. Attack_n]. Here we excluded the attack type 'worm' due to its' insufficient number of data samples (only 174 samples). After that, we have calculated variance fractal dimension of each feature for the normal dataset. The resultant VFD values of features for individual attacks and normal data will be the basis for the complexity analysis of our proposed feature selection method.

3.2.1 Variance Fractal Dimension:

Variance Fractal Dimension (VFD) is a type of fractal dimension that extracts the variance feature on an object. Variance measures the spread of samples around its mean. VFD calculation is done with Hurst exponent which is denoted with H. The variance that is denoted with σ^2 is measured by its amplitude increments over a time series and expressed through below power law [48],

$$Var[x(t_2) - x(t_1)] \sim |t_2 - t_1|^{2H} \quad (1)$$

So,
$$Var[x(t_2) - x(t_1)] = Var[\Delta x_{\Delta t}] \quad (2)$$

Let us assume,

$$x(t_2) - x(t_1) = \Delta x_{\Delta t} \tag{3}$$

Then the equation 1 becomes,

$$\log[Var(\Delta x_{\Delta t})] \sim 2H log[\Delta t] \tag{4}$$

From equation 4, the Hurst exponent H will be,

$$H = \frac{1}{2} \lim_{\Delta t \to 0} \frac{\log[Var(\Delta x_{\Delta t})]}{log[\Delta t]} \tag{5}$$

The variance dimension $D\sigma$ can be then obtained from,

$$D_\sigma = E + 1 - H \tag{6}$$

Here E is embedded Euclidean dimension and E=1 for a single variable time series.

And therefore,

$$D_\sigma = 2 - H \tag{7}$$

The Hurst exponent (H) ranges between 0 to 1[49]. Therefore, for 0<H<1, variance dimension $D\sigma$ must be limited for $1 < D_\sigma < 2$. Hurst exponent is characterized by fractional Brownian motion or fBm. Fractional Brownian motion can be described as a self-similar stochastic process. When $D\sigma$=1.5, it represents the Fractional Brownian

Motion. For mono scale, it will be $D\sigma = 1$. When $D\sigma = 2$ it represents the white noise. Fractional Brownian motion has stationary increment.

For our VFD calculation, the algorithm was verified by calculating the variance dimension for the Brownian motion. For a Brownian motion the Hurst exponent H= $\frac{1}{2}$, hence the variance dimension was verified at $D\sigma = 1.5$.

3.3 Feature Identification: Finding discriminative features:

For each attack type, a comparative analysis was done for the VFD values of each features to the VFD values of each features for normal dataset. This was denoted with Vdiff in the below equation. To find out the Vdiff value for individual attacks we plot a Feature v/s VFD graph and further proceed for the complexity analysis. For each attack type, the features that are more significant will have greater Vdiff value hence discriminative in nature. For each attack type we established a threshold to identify up to which Vdiff value we will consider the feature as discriminant.

$$V_{diff} = | V_{attack\_n} - V_{normal} | \quad (8)$$

Now that we have calculated the Vdiff values, i.e. the difference between VFDs of features in attack samples with the normal samples, the next question is how to define the threshold for selection of significant features. We have already signified in the previous sections that the higher the difference between VFDs for normal and attack, the greater is the discriminative nature of a feature. However, for the algorithm to be effective we need

an objective way to find a threshold to determine which features are significant. In other words, we need to set a threshold for the Vdiff, any Vdiff greater that the threshold will be considered important, the features with Vdiff lower than this threshold will be considered insignificant. We can utilize the classic midline approach, in this case, we can find an optimum range by finding the midline between the two extrema. Below is the equation for the threshold. The $Vdiff_{th}$ is the threshold, which is the midline between the maximum Vdiff and the minimum Vdiff. Any feature having a Vdiff value higher than the midline, i.e. features with Vdiff > $Vdiff_{th}$ value, will be considered as significant. In the example provided in the below figure, the highest Vdiff, is .864 and the lowest value is .123, hence our threshold is (.864 + .123)/2 = 23423.

Feature	VFD attack 1	VFD Normal	Vdiff (VFD attack1-VFD normal)	
1	.987	.123	.864	→ Vdiff_max
2	.	.	.	
3	.	.	.	
.	.	.	.	
.	.567	.444	.123	→ Vdiff_min
47	.	.	.	

Figure 6: Sample of Vdiff threshold selection

The threshold for Vdiff was considered at,

$$\mathbf{Vdiff_{th}} = \frac{\mathbf{Vdiff_{max} - Vdiff_{min}}}{\mathbf{2}} \quad (9)$$

The discriminative features for each type of attack are maintained in a different list and this list of significant features are used in later phases to train a neural network.

3.4 Testing the Discriminative Features: Applying Artificial Neural Network as Classification Model

After the algorithm has autonomously filtered the significant features based on complexity and discriminativeness, at this portion these features should be tested in a classification model. For the comparison, for each attack type, we use an artificial neural network (ANN) to detect the attack using a dataset with all the features and then use the same model and dataset with only the selected features. We capture the performance metrics and then compare the outputs.

Artificial neural network is a simple, easy to use, well established method with great learning capabilities and effectiveness in capturing anomalies. ANNs are known for their abilities to learn, classify, process information faster and as well as their ability of self-organization. These are the reasons why ANNs can increase the accuracy and efficiency of intrusion detection model.

3.4.1 Artificial Neural Network (ANN):

An Artificial neural network can be defined as a nonlinear function approximation that can learn, classify and predict. There are several applications of neural network for instance; input-output mapping, classification problem, prediction problems. There are various types of neural networks that are being used in diverse field of studies like- pattern

recognition, speech recognition, medical studies and so on. However, the most commonly used ANN in the field of cyber security is the Multi-Layer Perception (MLP) which is a supervised learning method. The basic functionality of a Multi-Layer Perception model is to work as a feed forward ANN model that can map input data into suitable output [50].

The application of using artificial neural networks is advantageous as it uses an automated and intelligent learning technique which can classify normal and attack samples. Further, ANN does not require extensive knowledge of system architecture or software and hardware knowledge is not required.

3.4.2 Supervised and Unsupervised Neural Networks:

The key difference in supervised and unsupervised learning is supervised learning maps the input to output label while unsupervised learning is used to learn the inherent structure of the data using their statistical properties and without the label. Supervised learning typically done in the context of classification, or regression problem while unsupervised learning is often used in clustering, representation learning. The well-known algorithms used in supervised learning include logistic regression, support vector machine, artificial neural network, random forest and naive bayes. In our experiment we have used supervised learning model for the classification.

3.4.3 Overview of the ANN architecture:

An ANN is model is made of several neurons that mimics the neuron of a human brain which is the central processing unit of the brain. The idea is to make the model learn the

complexity of input data. In an MLP model, there are usually three layers the input, hidden, and output layer where each layer is has at least m number of neurons (where m>1). The hidden layer neurons are linked with the neurons of adjacent layers. Each neuron is also linked to a constant number called bias. The two types of connections between neurons- the first layer connections and the second layer connections have its distinct weight value that is multiplied to the neuron value of the of the connection of preceding layer. The required number of input-output neurons is determined by the dimension of input and output data.

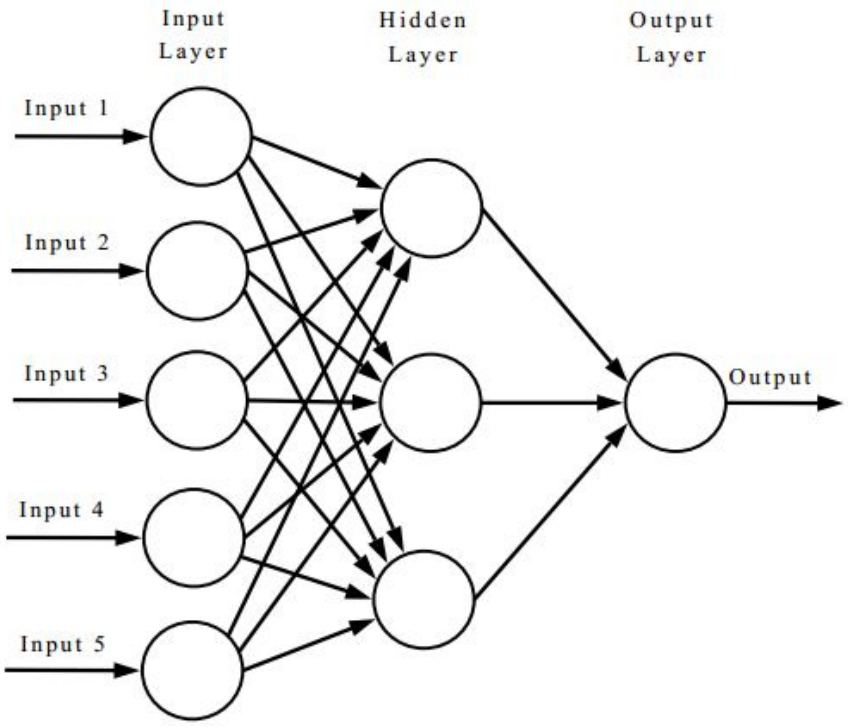

Figure 7: Structure of a basic Artificial Neural Network (picture source:[51])

After determining the values of the inputs, weights, activation function, bias, and the output vector, the ANN is then trained. The data set in the training phase is fed to the ANN with a specific learning rate which later adjusts the weights to minimize the error. The error information is then fed back to the network. The training of the ANN continues until a targeted output is achieved. Once the network is trained, the ANN is then setup for validation and testing. During the validation phase, a partial input data set is used while a new set of unknown data is used in the testing phase.

3.4.3 Forward Propagation:

The forward propagation corresponds to computing the output of each neuron in each layer, except for the input layer. The inputs are associated with a weight value which the inputs pass forward to an activation function and then the output from the activation function are passed on as inputs to the next layer to learn a training set of input output pair. Figure 8 depicts the forward propagation of activations. Each layer is connected by wight values. There is also a wight called bias that is connected with individual neurons of the hidden and output layer. Usually this bias value is constant.

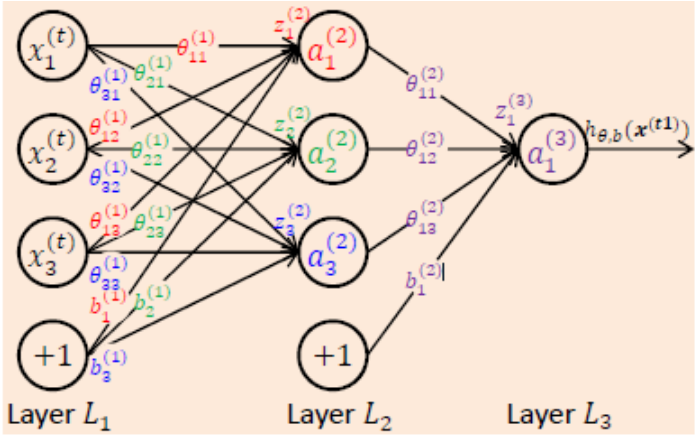

Figure 8: ANN Architecture (Copied by permission from[52])

The activations of each layer can be calculated with below equations,

$$a_1^{(2)} = g\left(z_1^{(2)}\right) = g(\theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3 + b_1^{(1)}) \quad (10)$$

$$a_2^{(2)} = g\left(z_2^{(2)}\right) = g(\theta_{21}^{(1)} x_1 + \theta_{22}^{(1)} x_2 + \theta_{23}^{(1)} x_3 + b_2^{(1)}) \quad (11)$$

$$a_3^{(2)} = g\left(z_3^{(2)}\right) = g(\theta_{31}^{(1)} x_1 + \theta_{32}^{(1)} x_2 + \theta_{33}^{(1)} x_3 + b_3^{(1)}) \quad (12)$$

$$a_1^{(3)} = g\left(z_1^{(3)}\right) = g(\theta_{11}^{(2)} a_1 + \theta_{12}^{(2)} a_2 + \theta_{13}^{(2)} a_{\,3} + b_1^{(2)}) \quad (13)$$

Where the output,

$$h_{\theta,b}(x) = a_1^{(3)} \quad (14)$$

The cost function,

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} cost(h_\theta(x^i), y_i) \quad (15)$$

Our experiment ANN used logistic cost function.

3.4.4 Back Propagation:

Back propagation is the basically the backward propagation of the errors. In a supervised learning model given an error function, back propagation calculates the gradient descent of the error function to update weights. The calculation of the gradient descent proceeds towards the backwards in an ANN. The gradient computation from one layer is used for

the computation of the gradient of preceding layer. This allows the efficient computing of gradient in each layer.

For each wight in the layer L_2 (in figure 8) is calculated with,

$$\theta_{1j}^{(2)} = \theta_{1j}^{(2)} - \alpha\, \delta_1^{(3)} a_j^{(2)} \tag{16}$$

$$b_1^{(2)} = b_1^{(2)} - \alpha\, \delta_1^{(3)} \tag{17}$$

For each wight in layer L_1,

$$\theta_{ij}^{(1)} = \theta_{ij}^{(1)} - \alpha\, \delta_1^{(3)}.\theta_{1i}^{(2)}.a_i^{(2)}(1 - a_i^{(2)}).x_j^{(t)} \tag{18}$$

$$b_i^{(1)} = b_i^{(1)} - \alpha\, \delta_1^{(3)}.\theta_{1i}^{(2)}.a_i^{(2)}(1 - a_i^{(2)}) \tag{19}$$

$$\delta_1^{(3)} = \left(a_1^{(3)} - y^{(t)}\right) a_1^{(3)}(1 - a_1^{(2)}) \tag{20}$$

The rate of change of error is calculated with,

$$\frac{\partial J}{\partial a_1^{(3)}} \tag{21}$$

So, the local gradient can be written as,

$$\delta_1^{(3)} = \frac{\partial J}{\partial a_1^{(3)}}.\frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} \tag{22}$$

For logistic regression cost function, the equation 15 will then be,

$$\delta_1^{(3)} = \frac{\partial}{\partial a_1^{(3)}} \left(-y^i \log\left(a_1^{(3)}\right) - \left(1 - y^i\right) \log(1 - a_1^{(3)})\right).a_1^{(3)}(1 - a_1^{(3)}) \tag{23}$$

An appropriate learning rate is set for satisfactory training of the ANN which is usually based on the architecture of the network. After the system is trained, we set the ANN to work on the testing dataset.

3.4.5 Dataset Creation:

In the 8 different attack types, most of the attacks have attack rows of 10 thousand or higher. For these attack types we construct our dataset by randomly selecting 30,000 normal rows and 10,000 attack rows. And for the attack types like Analysis, Shellcode and Backdoor, with smaller attack samples in the original dataset, in the range of 10,000 or less; we construct the dataset with 1000 randomly chosen attack rows and 3000 randomly chosen normal rows. Out of this dataset, we choose 70% for training and 30% for testing where 5% of the training data were used as validation data (table 1). Hence, for the larger datasets we have 12,000 rows for testing and for the smaller data sets we have 1,200 rows for testing.

Table 1 : Dataset Partition

Data Sets	**Percentage of Data Set**
Training Data	70%
Validation Data	5%
Testing Data	30%

3.7 Results evaluation:

The intension of the proposed method of discriminative feature selection as a preprocessing step is to achieve enhanced performance of machine learning classification algorithm. This also gives an efficient solution to the best response that should be taken in consideration with regards to the individual intrusion type. Hence, the ANN classifier performance for each attack type with its reduced number of feature set were fairly evaluated and later compared with the performance of all features of each individual attack. Below performance metrics were considered to evaluate the performance.

$$\boldsymbol{True\ negative\ rate} = \frac{\boldsymbol{TN}}{\boldsymbol{TN+FP}} \tag{24}$$

$$\boldsymbol{True\ positive\ rate} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{25}$$

$$\boldsymbol{False\ positive\ rate} = \frac{\boldsymbol{FP}}{\boldsymbol{FP+TN}} \tag{26}$$

$$\boldsymbol{False\ negative\ rate} = \frac{\boldsymbol{FN}}{\boldsymbol{FN+TP}} \tag{27}$$

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \tag{28}$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{29}$$

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TN+TP+FN+FP}} \tag{30}$$

Chapter 4: Dataset

In this chapter we provide detailed information of the dataset that was used for our research. This chapter also provides a brief overview on the different attack types that are present in the dataset.

4.1 The UNSW-NB15 Dataset

In our experiment we have used the publicly available UNSW-NB15 dataset. The dataset was created utilizing the IXIA PerfectStorm tool in the cyber range lab of the Australian Centre for Cyber Security (ACCS) at UNSW in Canberra. A hybrid of normal and abnormal data can be found in the network traffic dataset [53]. Around 2.5 million data samples are distributed in four CSV files and in consist 49 different features including class label. Further, detail description of the features can be found at [53][54]. This dataset contains nine different types of attacks as illustrated in Figure 9 Type of Attacks.

Among 49 captured attributes, 47 attributes are of different data forms like binary, float, integer and nominal. Rest 2 attributes are labels and attack type where in label field attack data samples are denoted as ‘1,’ and normal data samples are denoted as ‘0’. The attack type field reflects the type of data class, i.e.; attack type or normal.

Figure 9: Type of Attacks in UNSW-NB15 dataset

4.1.1 UNSW NB15 Dataset record distribution:

A total of 2,537,715 data samples contain both normal and attack samples. The total number of normal data samples are 22,18,764 where the total number of attack samples are 321,283. Table 1 shows the distribution of each attack type and normal samples in the UNSW-NB15 data files.

Table 2: Dataset Record Distribution

Type of data samples	No. of records or samples
Normal	2,218,761
Fuzzers	24,246
Analysis	2,677
Backdoors	2,329
DoS	16,353
Exploits	44,525
Generic	215,481
Reconnaissance	13,987
Shellcode	1,511
Worms	174

4.1.2 UNSW NB15 Dataset attack description:

Fuzzers:

In this attack, the attacker targets for security loopholes of the network, the operating system or a program and cause the program, operating system or the network get crashed or suspended by feeding it massive number of randomly generated data.

Analysis:

This variety of intrusion occurs by penetrating the web application through ports, emails and web script. For example, port scans, spam, HTML files.

Backdoors:

Backdoor attack is a technique of bypassing a stealthy normal authentication, securing unauthorized remote access to a device, network or software application through which the attacker can access personal, financial data or even install additional malware [55].

DoS:

In a Denial-of Service (DoS) attack, the attacker makes a malicious attempt to make a machine or server inaccessible to its intended user by shutting down a machine or network by flooding it with traffic or crashing it with sent information that triggers the crash. Even though this attack typically does not result in theft or loss of information, it can cost an organization in terms of resources, time and money as the expected services and resources get inaccessible [56]. DoS attack generally occurs in two methods: flooding or crashing services. Flooding refers to the scenario where the system receives traffic too large for its server to buffer which eventually cause the system to slowdown and stop. The flooding attack can occur through buffer overflow attack, ICMP flood or smurf attack and SYN flood attack. In crashing method, the attack takes advantage of the vulnerabilities of the target system which subsequently make the system destabilize or crash so that it becomes inaccessible to use. Distributed Denial of Service or DDoS is an additional type of DoS attack where synchronized DoS attack is orchestrated by multiple systems that attacks a single target [57].

Exploits:

Exploits are programs or codes that finds a security gaps within an operating system or a piece of software that an intruder can use to deliver malware. Exploits can be generally classified into categories: known and unknown (or zero-day exploits). In known exploits intruder takes advantage of the known vulnerabilities in system or software. The more severe exploits are the unknown or zero day exploits where the system vulnerability is unknown and only apparent when attacker exploiting the vulnerability [58].

Generic:

Generic attack is a technique that works against all block-ciphers irrespective of the structure of the block-cipher. A cipher in cryptography refers to the algorithm that is performed in encryption or decryption. A block-cipher is the input data type which encrypt block of data of fixed size [59].

Reconnaissance:

In reconnaissance attack is aimed at gathering information of its target via either active or passive reconnaissance. In an active reconnaissance, the interaction with the target occurs directly whereas in a passive reconnaissance there is no direct interaction rather use information available on the web [60].

Shellcode:

Shell code refers to a small piece of code or a series of instruction that is used as the payload to exploit a vulnerability. Shellcode can be local or remote. Local shellcode is used

to take control over the machine in runs on while remote shellcode is used to access the target machine through a network [61].

Worms:

A computer worm is a type of attack where it replicates itself from computer to computer. It does not require human interaction to replicate itself instead, it uses a computer network to spread itself. Worms can transmit using software vulnerabilities or could spread though spam email attachments or instant messages in a computer.

Chapter 5: Experiments and Results

This chapter focuses on the experimentation, the workspace setup in which the proposed algorithm was put to test, and the results that was achieved to test the hypothesis of the proposed algorithm. Here we discuss in detail, what approaches were followed to make sure that the specific key attributes of the proposed algorithm are well-tested, and how the results are measured.

5.1 Experiment Setup

The experiment was performed on the university's distributed infrastructure systems that are shared amongst the researchers for performing research experiments requiring high computational resources. The system utilizes a backbone of multi-core CPUs and 256 GBs of RAM, shared amongst all its users, from which this simulation only uses a small portion. A python engine was installed, the experiment used various popular python libraries like pandas, scikit-learn, and several other libraries, some of which needed additional installation.

For testing the proposed variance fractal-based feature selection algorithm, we utilized the UNSW-NB15 dataset. The dataset has two versions, a smaller version and a larger version. For this study, the larger version was used, which comprised of four text files containing the raw data. The dataset comprised of 49 attributes including label and attack category. Each row in the dataset is a network event, and the attributes in the rows, determine the values captured during the event. The two attributes, attack_cat and label,

contains the label information, which identifies the row, i.e, the network event as normal or attack, and the attack_cat attribute specifies the type of attack. Before proceeding any further, these two attributes are discarded from our dataset, hence now the dataset contains in total 47 features.

5. 2 Test Procedure

The process of testing the algorithm needs to happen in three steps. The steps are elaborated in detail in the below sections. The purpose is to filter out the attributes that are discriminative in nature- discriminative as in, for normal traffic they will have certain character and for attack traffic they will have a significantly different character. As our proposed algorithm suggests, in order to do that we need to compute the variance fractal dimension trajectory for attack traffic and normal traffic and determine which attributes show the most difference in VFD values for attack traffic and normal traffic.

Henceforth, there are four logical steps for this process, firstly we need to compute the VFD values of each attribute for the normal traffic only, discarding all the attack traffic, and store the VFD values into a table. Secondly, we must load the dataset containing only the attack traffic and discard the normal traffic, and then compute the VFD of each of the feature. This process should be performed for each of the attack type and then the VFD values of the attributes for each attack type should be stored.

Thirdly, we should perform a comparison of the difference to compute which attributes show maximum difference and isolate them as the most discriminative attributes.

This is best depicted with a diagram comparing the VFD values of attributes for attack traffic and normal traffic.

Finally, once we have the entire list of significant discriminative features, we will be performing a test using an Artificial Neural Network to compute the we will be performing a simple test to measure the accuracy of these selected attributes in an artificial neural network and compare them with the accuracy of all the attributes.

Step 1

In order to understand the discriminative nature of the attributes, and the difference of the VFD values of the attributes for normal traffic and attack traffic, let us first calculate the VFD values of each of the attribute only for normal data. In order to do that, we construct a data set only with the normal data and get rid of all the attack data. Once we have a dataset with only the sequential normal traffic, we traverse through all the attributes, one after another, until we have processed all the attributes. The process that we perform for each of the attributes are as follows. For each of the attribute we focus on the column and start from the first row till the very last row of that column and calculate the VFD for that attribute. In this experiment we find out the variance fractal dimension trajectory for all the attributes for the normal data only.

Step 2

In the next step we perform the similar process for the eight attack types in our dataset. For each of the attack types, we remove all rows from the dataset excepting the

ones that belong to the specific attack traffic. In this manner we calculate the VFD values for each of the attributes one after another and store the VFD value in a table.

After completing this process for one attack, we do the same for another attack. We start with the full dataset, filter out all the events / rows that do not pertain to traffic from this specific attack, and start calculating the VFD values for each of the attribute, record the VFD values in a table and move on to the next attack type, so and so forth.

Lastly, we will have the list of VFD values for the 8 attack types and the normal traffic, in total 9 set of VFD values for every attribute.

Step 3

Now that we have calculated the VFD values for each attribute for the normal traffic and the attack traffic, we may proceed to the next phase, which is to identify the discriminative features that show major differences of VFD values with normal traffic and attack traffic. In order to do that we have calculated the difference between the VFD values of attack and the VFD values of normal. We sort the VFD value difference in ascending order and arrange the attributes in that order. The plots are presented in the result section below.

5. 3 Result & Discussion

In the below figures (Figure 10- Figure 17) we have plotted the VFD values for normal traffic and the VFD values for attack traffic. The VFD values for normal traffic are depicted with the blue line, and the orange line represents the VFD values for the attack traffic. The features in the x-axis are organized in increasing order of the difference of VFD values of

attack and normal, i.e. with the features with the smallest VFD differences of normal and attack traffic appears on the left and the ones having the most differences appears on the right in the x-axis [Vdiff = | $V_{attack} - V_{normal}$ | in Equation 1]. That is why the blue line and the orange line appears closer in the left and grows apart towards the right. Therefore, the most significant features with the most discriminative characteristics are the rightmost features in the graphs, where the value difference for VFD are the greatest between normal traffic and attack traffic. Since we are considering the absolute difference, we can see the lines intersecting, sometimes the V_{attack} is higher than the V_{normal} and sometimes for the immediate next attribute on the x-axis the scenario is reversed. That is why the blue line may be on top of the orange line, but for the adjacent feature on the right, the orange line can be on top of the blue line, however, each attribute in the right has an increased difference in VFD values between the attack and normal than the value difference for the attribute in its adjacent left, so and so forth.

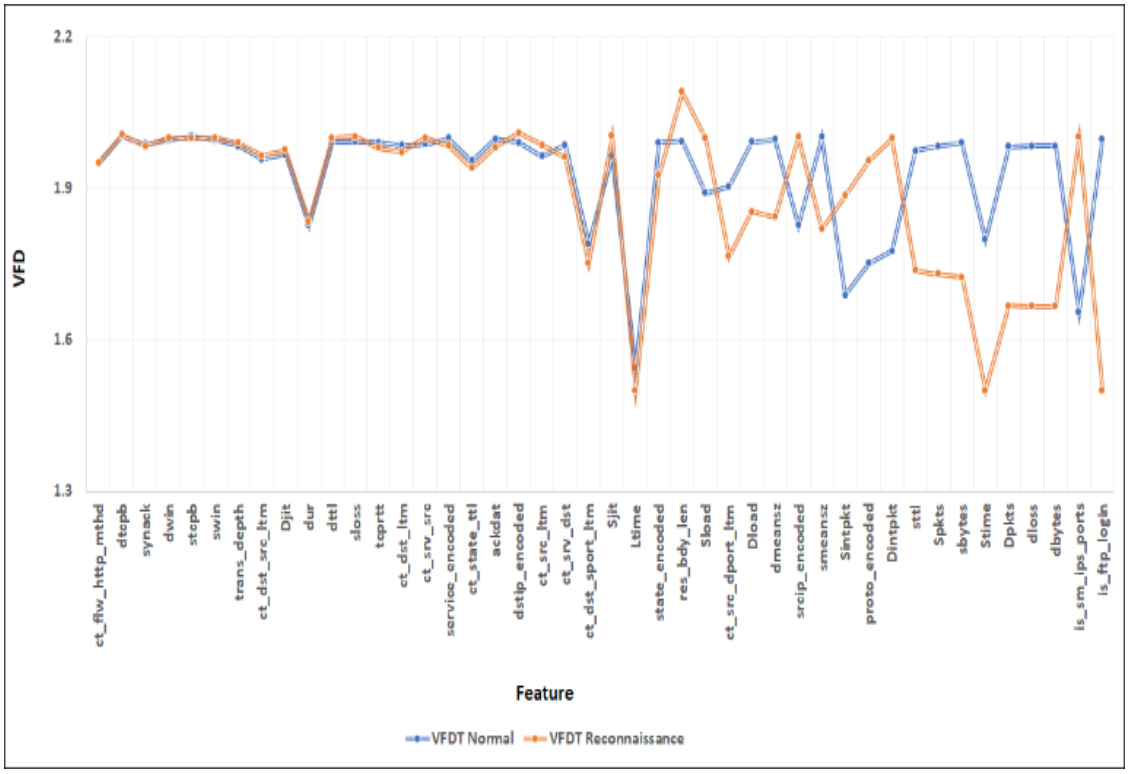

Figure 10: VFD v/s Feature graph for normal and Reconnaissance

Figure 10 represents the attack type reconnaissance. Towards the right, we can observe the significant difference from the sharp peaks.

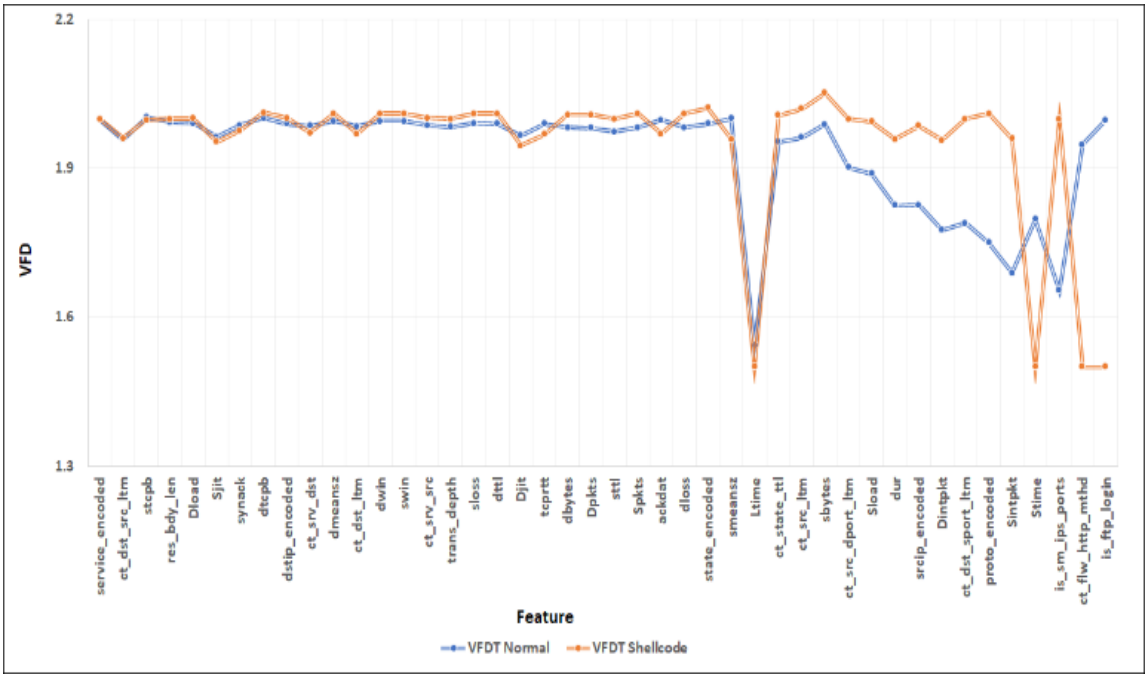

Figure 11: VFD v/s Feature graph for normal and Shellcode

For attack type Shellcode, only a few features show discriminative nature with respect to the normal VFD value.

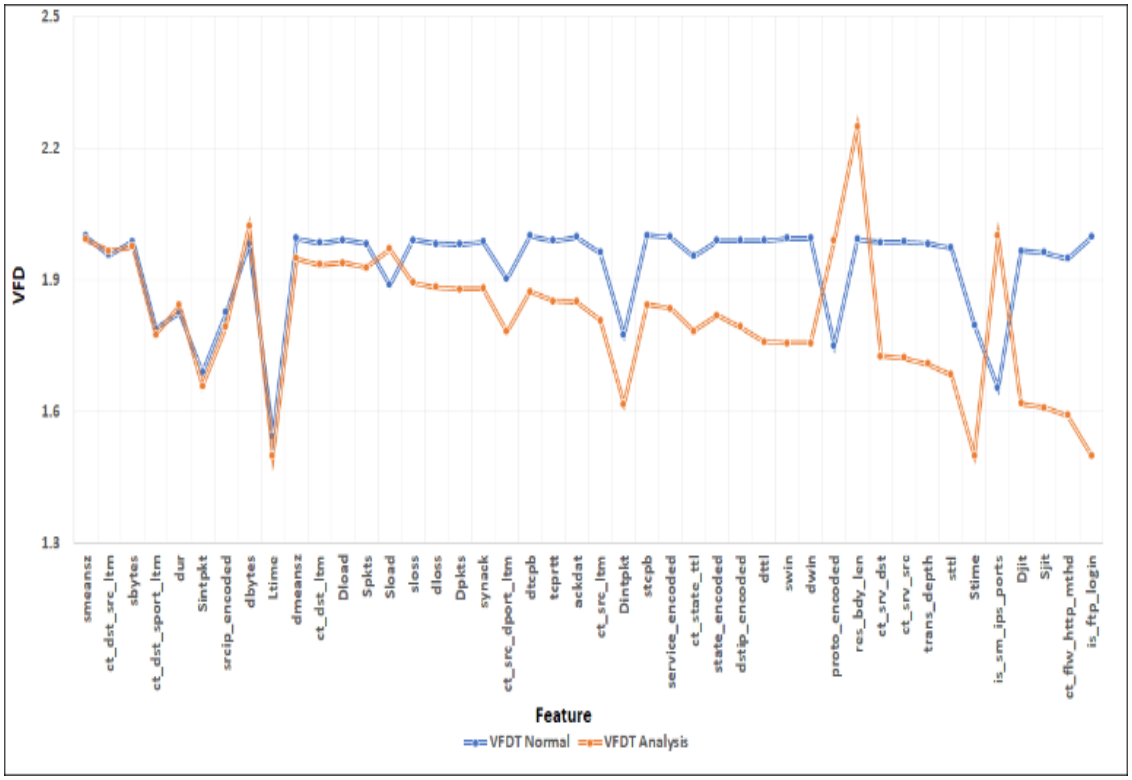

Figure 12: VFD v/s Feature graph for normal and Analysis

In figure 12 for attack type Analysis, we can see that there is a significant VFD value difference in in majority of the features forwards the right.

Figure 13: VFD v/s Feature graph for normal and Backdoor

In figure 13, we can also see that there is significant difference in the features where the attack graph and the normal graph are significantly apart in many of the features towards the right.

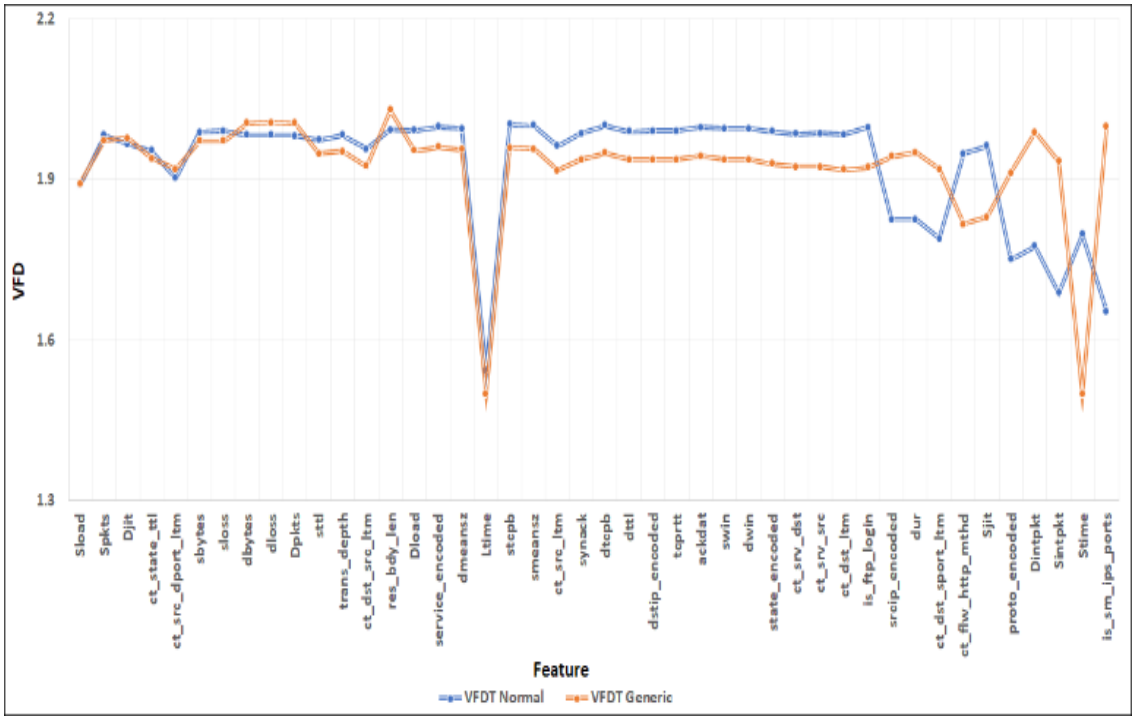

Figure 14: VFD v/s Feature graph for normal and Generic

The graph in the figure 14 shows a very little difference in terms of VFD value for the attack Generic to Normal.

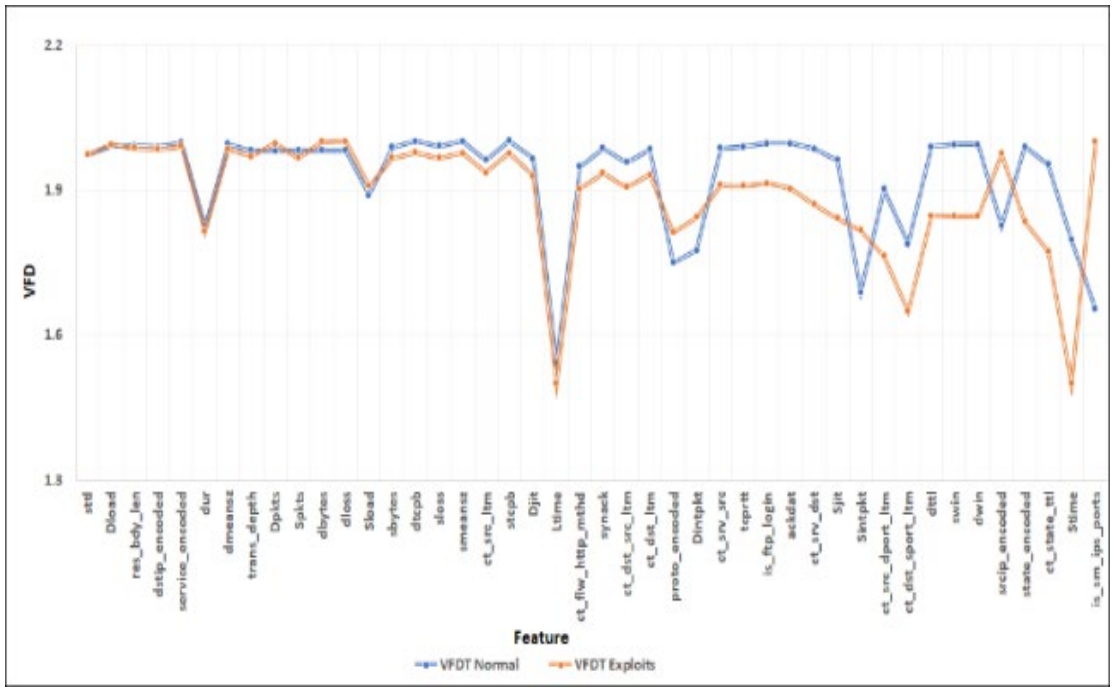

Figure 15: VFD v/s Feature graph for normal and Exploits

For the attack type Exploits in figure 15, we see the least number of features that can be identified as significant.

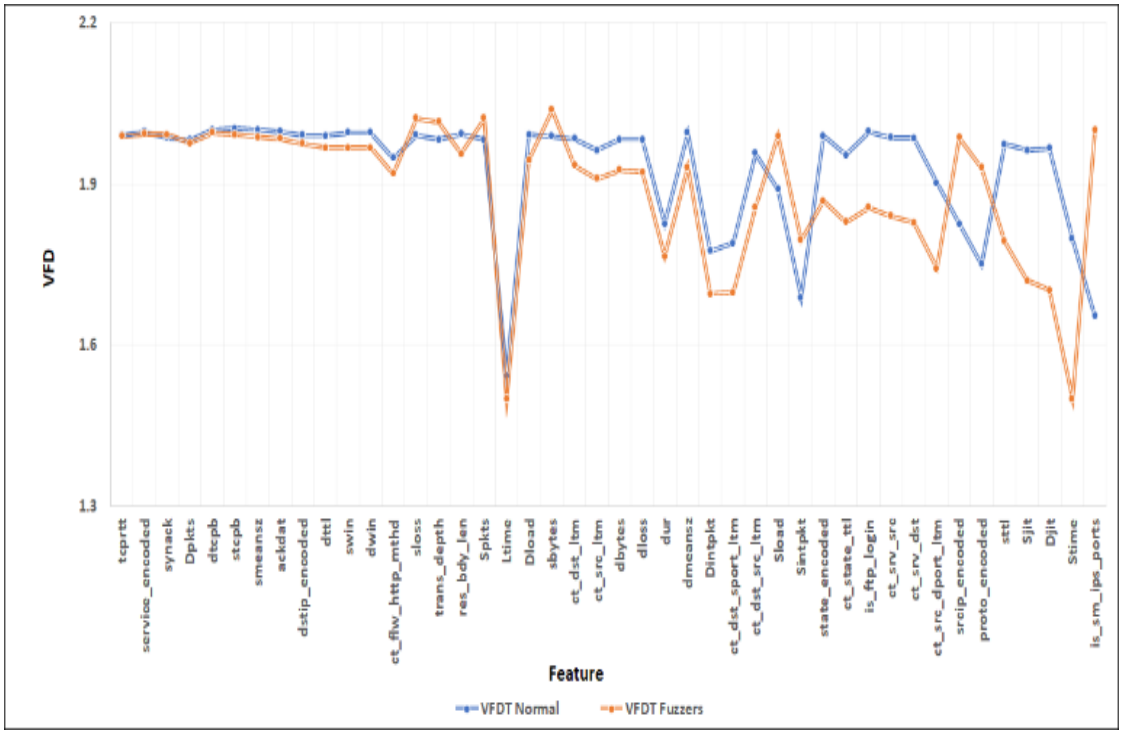

Figure 16: VFD v/s Feature graph for normal and Fuzzers

Form the graph for the attack type Fuzzers in the figure 16, we can infer about 14% of the features show distinct difference in VFD value with respect to normal.

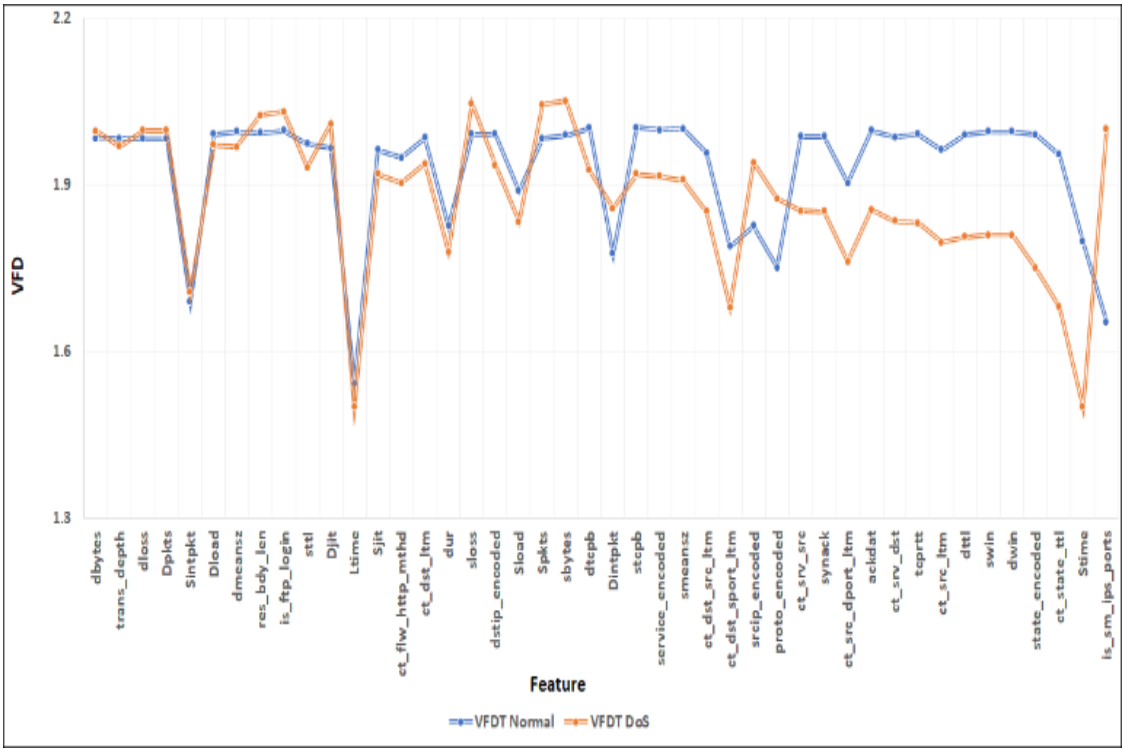

Figure 17: VFD v/s Feature graph for normal and DoS

DoS attacks in figure 17 also show similar nature as Fuzzers and show discriminative nature for around 15% of the features.

One interesting way to look at the discriminative features would be to check if we can infer any relationship among the discriminative features of all attacks. In figure 18 plotted the VFD values for normal traffic and the VFD values for all attack traffic. The VFD values for normal traffic are depicted with the blue line, for which we have plotted the values in ascending order. The pink, grey, green and light blue represent the attack Generic, Exploits, Reconnaissance and Shellcode. The orange, red, purple and yellow represent the Analysis, Backdoor, Fuzzers and DoS respectively. For the feature stime almost all the attacks show a significant difference from normal in terms of complexity. We also see similar pattern for the feature is_ftp_login. However only 4 attacks show significant difference and other 4 attacks reveal complexity that are neither close to the value of normal nor the values of preceding attacks. The features in the x-axis are organized in increasing order of the difference of VFD values of attack and normal,

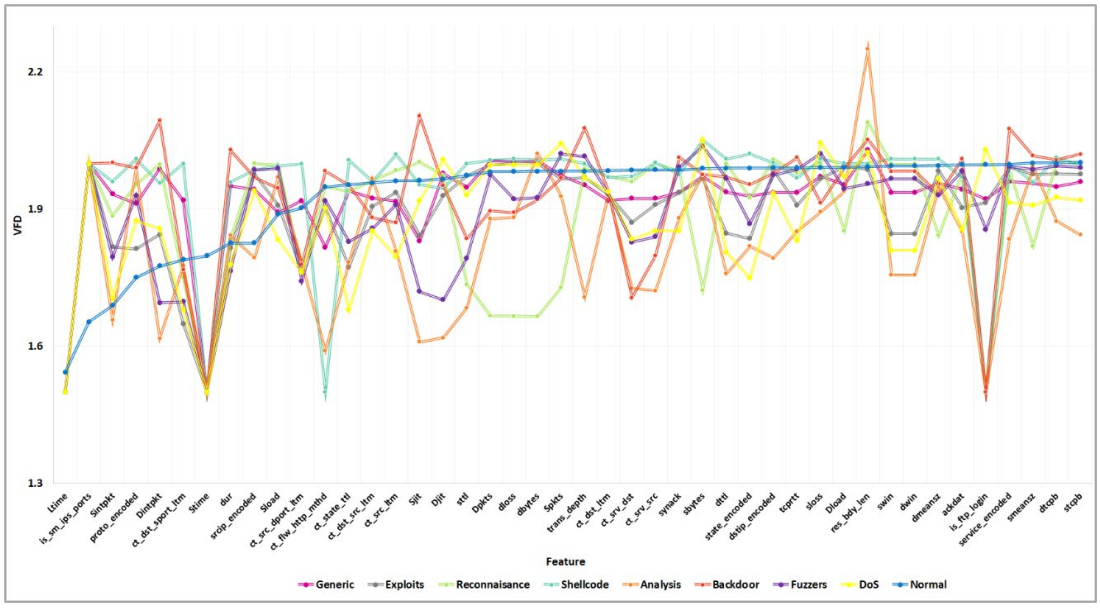

Figure 18: VFD v/s feature graph for Normal and all attacks

For all other features the attacks do not show significant nature that can depict a similar complexity that is discriminative with respect to the normal. However, for some small group of attacks we can see the VFD values are closer and well apart from the VFD values of normal. Hence, we can state that the analysis can be done for group of attacks considering the similar VFD values. However, this do not provide a conclusion for finding discriminative features that are applicable for all attack types. Furthermore, applying ANN with all features that are discriminative for all attacks do not provide a solution to identify individual attack type and their discriminant features.

Once the significantly discriminative attributes are filtered out considering the midline criteria set in the proposed algorithm, in Equation 2, they are ready to be tested in the subsequent sections of the algorithm. In Table 1, we have summarized the number of attributes that have fulfilled the midline criteria, and has been classified as significantly discriminative attributes.

As depicted in the table 2, for Analysis attacks, 11 features have a difference of VFD values greater than the threshold, and for Shellcode, there are only 6 features that have VFD difference greater than the midline threshold. These features are filtered out and passed on to the later sections, where the algorithm trains and tests a neural network with a dataset created only with theses filtered out attributes. The model is trained and tested, and the accuracy and other evaluation parameters are captured and compared with the model tested with the entire dataset. In the paragraphs below we describe the dataset creation process.

Table 3: Reduced features for each attack

Attack Name	Reduced no of discriminative features
Fuzzers	6
Analysis	11
Backdoors	6
DoS	7
Exploits	3
Generic	4
Reconnaissance	8
Shellcode	6

In order to evaluate our proposed algorithm, we then train and test an ANN using only the reduced number of features. Among the 8 attack types that were considered in the experiment, most of the attacks have attack rows of 10,000 or higher. For these attack types we construct our data subset by randomly selecting 30,000 normal rows and 10,000 attack rows. And for the attack types like Analysis, Shellcode and Backdoor, with smaller attack samples in the original dataset, in the range of 10,000 or less; we construct the dataset with 1000 randomly chosen attack rows and 3000 randomly chosen normal rows. Out of each data subsets, we choose 70% for training and 30% for testing. Hence, for the larger datasets we have 12,000 rows for testing and for the smaller datasets we have 1,200 rows for testing. Further, the detection performance of minimized dataset was compared with non-minimized dataset.

As shown in the below Table 3, the accuracy for supervised learning has been on the high 90s. Scikit-learn's neural network-based learning models are known to provide high accuracy when it comes to supervised learning. When the number of features in the training dataset are reduced to only the selected few significant features, the accuracy drops. The rate of the decrease remains sporadic in nature; however, it is witnessed that despite the significant data loss the accuracy doesn't suffer too much.

Accuracy for Shellcode suffers the most at 91.83% and the accuracy of Analysis remains quite high at 99%. For all the other attack types, the accuracy decreases around 1-7%. The results show that losing the attributes were worthwhile, it is because those attributes were non-discriminatory in nature, which is why losing them did not decrease the accuracy substantially.

Table 4: Testing Accuracy for All Features vs the Selected Discriminatory Features

Attack Type	Accuracy (%)	
	All Features	Reduced features
Reconnaissance	99.92%	96.05%
Shellcode	99.72%	91.83%
Analysis	99.79%	99.08%
Backdoor	99.99%	92.92%
Fuzzers	99.59%	96.10%
Generic	99.97%	95.22%
Exploits	99.39%	97.94%
DoS	99.93%	98.04%

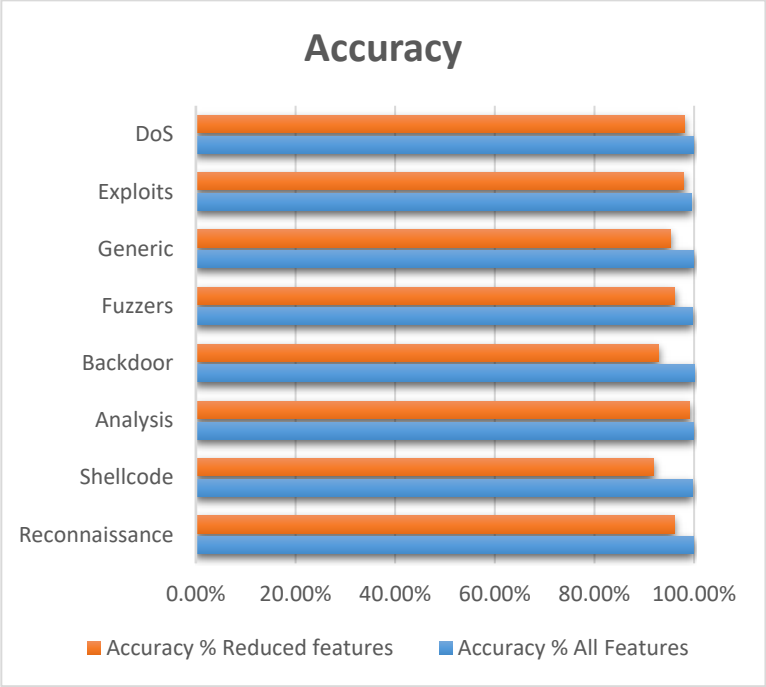

Figure 19 Testing Accuracy for All Features vs the Selected Discriminatory Features

Our proposed method was also experimented for the testing accuracy when we randomly select same number of features as reduced features. Table 4 shows the comparison of testing accuracy for randomly selected same number of features as reduced features vs the selected reduced discriminatory features. Even though it's a supervised model we see a significant drop in the testing accuracy for randomly selected features for all the attack types. Accuracy for Backdoors and Exploits suffer most at 66.56% and 63.40% respectively. In the case of Fazzers, DoS and Shellcode number of randomly selected features were 6,7 and 6 respectively for which we see the accuracy at 71.96%, 72.03% and 72.95% correspondingly. Generic gives 76.18% accuracy for 4 randomly picked feature.

Table 5: Testing Accuracy for randomly selected same number of features as reduced features vs the Selected Discriminatory Features

Attack Type	Accuracy (%)	
	Randomly selected same number of features as reduced features	Reduced features
Reconnaissance	84.79%	96.05%
Shellcode	72.95%	91.83%
Analysis	82.53 %	99.08%
Backdoor	66.56%	92.92%
Fuzzers	71.96%	96.10%
Generic	76.18%	95.22%
Exploits	63.40%	97.94%
DoS	72.03%	98.04%

Reconnaissance and Analysis give more than 80% accuracy. Since the number of features selected for both attacks are higher than the rest, the model gets higher probability of selecting features that can contribute to better accuracy. The highest accuracy that was achieved by selecting random number of features was 84.79% for attack Reconnaissance. The results depict some interesting numbers while comparing with accuracy result of the reduced discriminatory features. However, these randomly selected features do not have any selection criteria and do not comprehend discriminative characteristics.

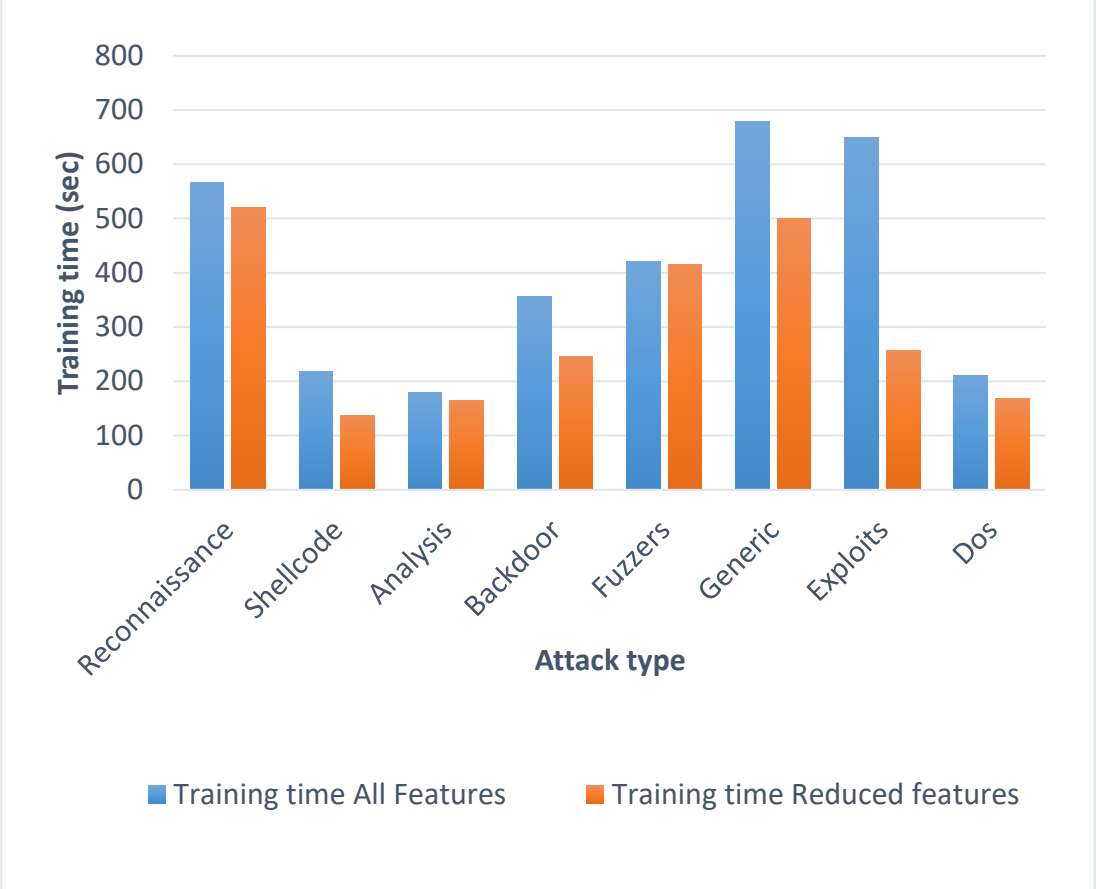

Figure 20: Training time for attacks with reduced feature and all features

The proposed method reduces the training time taken for reduced features in comparison to the time taken for all features (Figure 20). The reduction seems greater for the attack types like Generic and Exploits, and less for attack types like Fuzzers and Analysis. The reduction is training time suggests that the reduction of the features that were not discriminatory enough was justified and is a meaningful improvement.

Figure 21: Confusion Matrix - Reconnaissance

In the above confusion matrix, we can see that the results were sufficiently accurate, with 8556 normal accurately identified as normal, and 2970 attacks were accurately identified as attacks. Some 130 attacks were misidentified as normal and 344 normal were falsely identified as attacks. As we can see the misidentification is significantly less than the portion of accurately identified samples. Further accurate comparison will be presented in the below section.

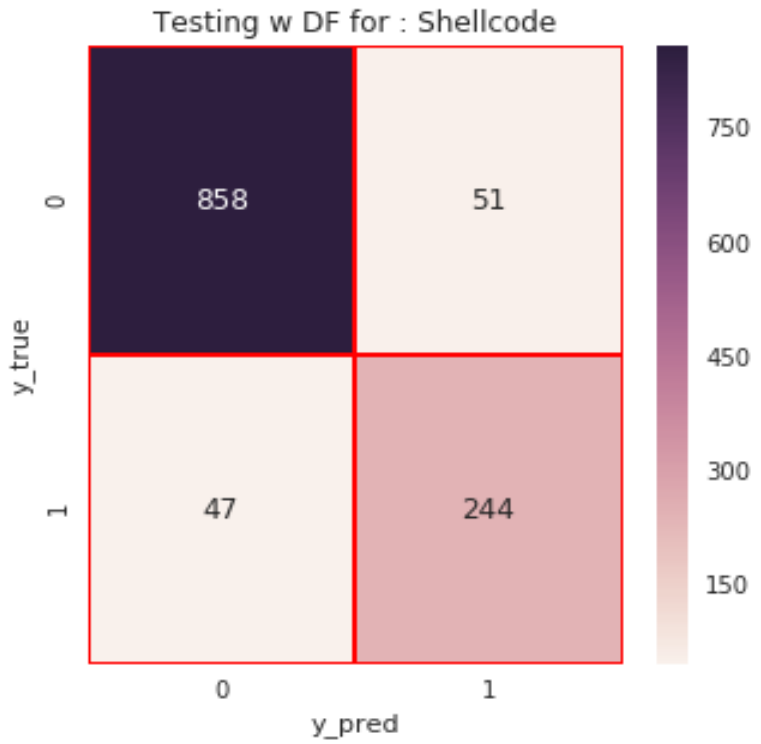

Figure 22: Confusion Matrix - Shellcode

In the above confusion matrix, we can see that the results were sufficiently accurate, with 858 normal accurately identified as normal, and 244 attacks were accurately identified as attacks. Some 47 attacks were misidentified as normal and 51 normal were falsely identified as attacks. As we can see the misidentification is significantly less than the portion of accurately identified samples, we will be able to focus on the accurate comparisons like accuracy and recall in the later sections.

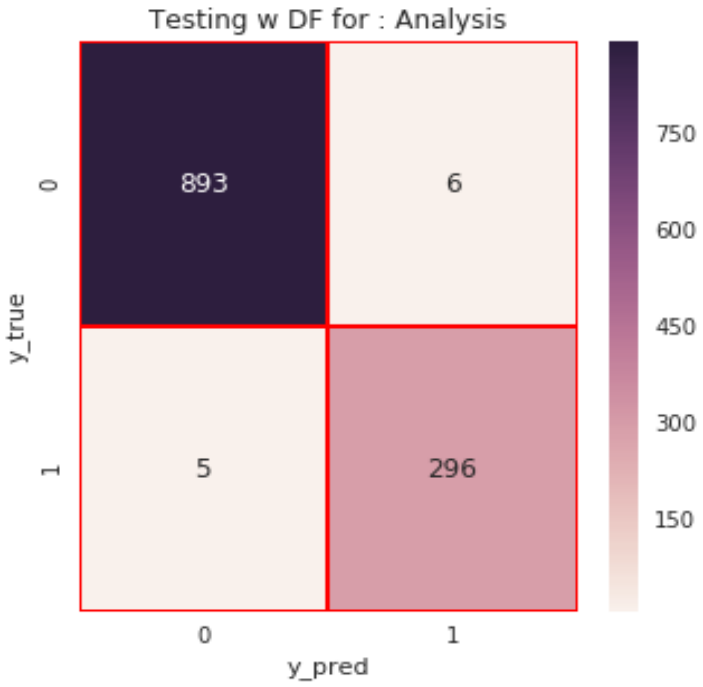

Figure 23: Confusion Matrix - Analysis

In the above confusion matrix, we can see that the results were sufficiently accurate, with 893 normal accurately identified as normal, and 296 attacks were accurately identified as attacks. Some 5 attacks were misidentified as normal and 6 normal were falsely identified as attacks. As we can see the misidentification is significantly less than the portion of samples accurately identified samples, we will be able to focus on the accurate comparisons like accuracy and recall in the later sections.

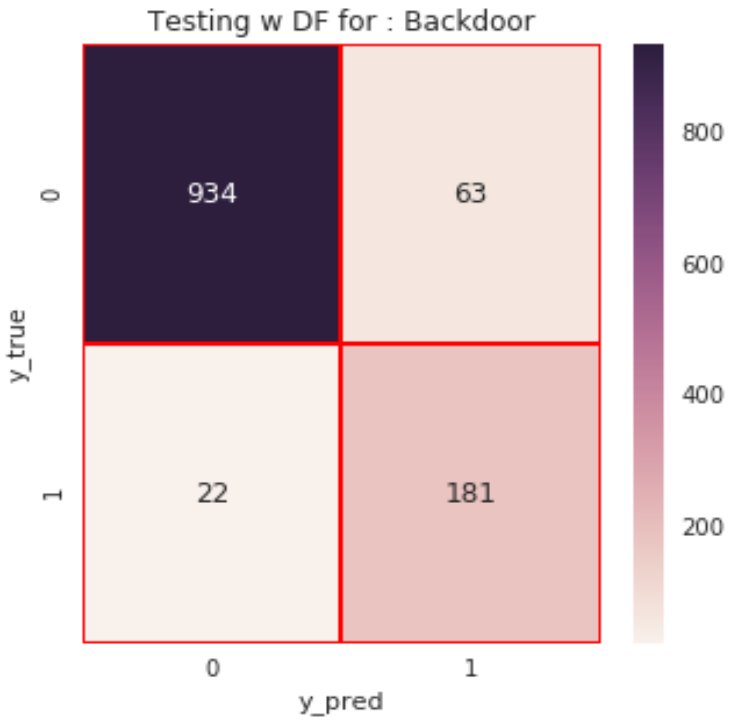

Figure 24: Confusion Matrix - Backdoor

In the above confusion matrix, we can see that the results were sufficiently accurate, with 934 normal accurately identified as normal, and 181 attacks were accurately identified as attacks. As can be seen, 22 attacks were misidentified as normal and 63 normal were falsely identified as attacks.

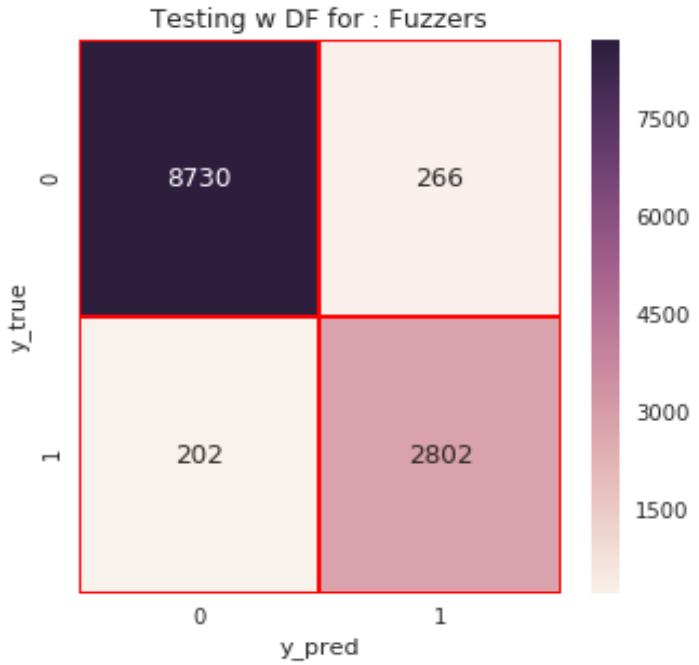

Figure 25: Confusion Matrix - Fuzzers

In the above confusion matrix, we can see that the results were sufficiently accurate, with 8730 normal accurately identified as normal, and 2802 attacks were accurately identified as attacks. 202 attacks were misidentified as normal and 266 normal were falsely identified as attacks. In later sections we will able to see how the values equate to items of the evaluation criteria.

Figure 26: Confusion Matrix - Generic

In the above confusion matrix, we can see that the results were sufficiently accurate, with 8920 normal accurately identified as normal, and 2506 attacks were accurately identified as attacks. There are some 474 attacks were misidentified as normal and 100 normal were falsely identified as attacks. The number of times the model as performed accurately seems much higher than the times the model has performed inaccurately.

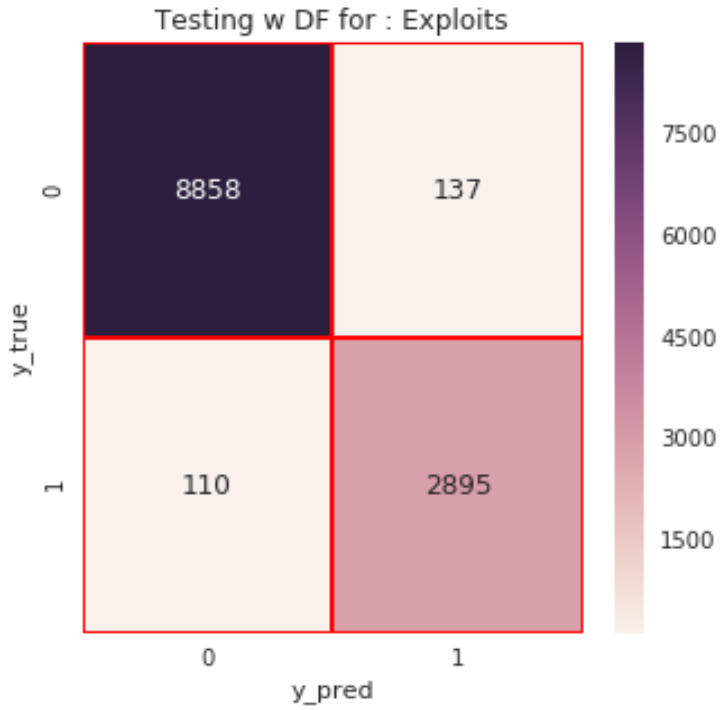

Figure 27: Confusion Matrix - Exploits

In the above confusion matrix, we can see that the results were quite good, with 8858 normal accurately identified as normal, and 2895 attacks were accurately identified as attacks. There are some 110 attacks were misidentified as normal and 137 normal were falsely identified as attacks.

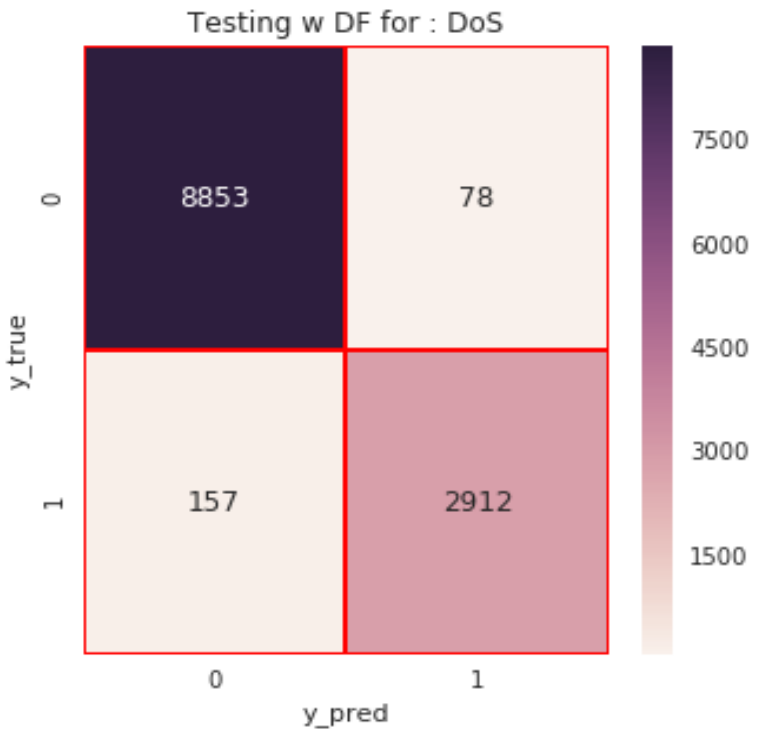

Figure 28: Confusion matrix - DoS

In the above confusion matrix, we can see that the results were sufficiently accurate, with 8853 normal accurately identified as normal, and 2912 attacks were accurately identified as attacks. There are some 157 attacks were misidentified as normal and 78 normal were falsely identified as attacks. These results show that the evaluation metrices for the model will have great results.

It should be noted that, not only the proposed method performs well for the classification problem at hand, but it also reduces computational complexity while identifying features that are important to individual distinct attack types instead of an indefinite or collective attack types.

Chapter 6: Conclusions and Recommendation for Future Work

This chapter provides concluding remarks on the thesis. A review on the results and the contribution of this thesis is discussed in this section. An overview of possible future work is also outlined in this section.

6.1 Conclusions

This thesis proposes a cognitive feature selection technique of a machine learning classifier model for the purpose of finding the discriminative features of each attack type of a network dataset. The presented method uses variance fractal-based complexity analysis as a feature selection technique for an artificial neural network. The experiment was validated for 8 different attack types of the UNSW-NB15 dataset namely; Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode. For each of the attack types, complexity analysis was performed in the form of variance fractal dimension calculations, for every feature within the dataset, separately for attack traffic and normal traffic. The features showing maximum difference in complexity were identified as significant features, and they were tested in subsequent sections. The primary goal of this proposed method was to identify discriminative features of each attack types while reducing the computational complexity. The overall computational complexity was reduced by removing features that do not possess any significance in attack detection and hence could help achieve reduction in data dimension. Further, an artificial neural network was used to test the effectiveness of the identified significant features. The experiment was

carried out with minimized dataset for each attack types to compare with the non-minimized dataset.

One of the key findings for this thesis was the application of complexity analysis in the field of feature selection. The experiment with variance fractal dimension as a tool for complexity-based analysis shows promising result in selecting cognitive discriminative features. Further, this research gives an insight on features that are significant to the distinct attack type. The resultant reduced feature set improves the detection performance of the classification algorithm in terms of computational complexity and detection time while giving substantial accuracy rate.

6.2 Recommendations for Future Work

Fractal-based method involves computation time for fractal analysis which motivates the authors to further work in future to assess a work around to optimize the computation time in feature selection stage. Moreover, a substantial improvement can be achieved to improve the accuracy performance. The thesis utilizes the variance fractal dimension for the proposed algorithm. In future, other fractal dimensions like, information fractal dimension, box counting fractal dimension can be used to experiment and validate the performance of the model. Also, the current work was done using the UNSW-NB15 dataset which can be further expanded and compared by applying the model for other internet dataset with a range of different attack types. This research can also be expanded in the direction of comparison in classification algorithm and detection method. The model can be compared for other machine learning classification algorithms like- support vector machine, naive Bayes model or decision tree. Also, the classification model uses

supervised learning algorithm. In future, this feature selection technique can also be experimented by using unsupervised learning algorithms. Another direction to continue this work in future can be comparing the proposed algorithm with other conventional feature selection methods.

Reference:

[1] McAfee, “Economic Impact of Cybercrime—No Slowing Down Report,” 2018.

[2] Steve, “2017 Cybercrime Report.”

[3] “Data Never Sleeps 7.0 Infographic | Domo.” [Online]. Available: https://www.domo.com/learn/data-never-sleeps-7. [Accessed: 05-May-2020].

[4] “Infographic: The Four V’s of Big Data | IBM Big Data & Analytics Hub.” [Online]. Available: https://www.ibmbigdatahub.com/infographic/four-vs-big-data. [Accessed: 03-Sep-2020].

[5] H. Bahsi, S. Nomm, and F. B. La Torre, “Dimensionality Reduction for Machine Learning Based IoT Botnet Detection,” in *2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV 2018*, 2018, pp. 1857–1862.

[6] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, “Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset,” *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5947–5957, 2011.

[7] I. Guyon, *Feature Extraction Foundations and Applications*, vol. 207, no. 10. 2006.

[8] Y.-H. Taguchi, “Unsupervised Feature Extraction Applied to Bioinformatics A PCA Based and TD Based Approach Unsupervised and Semi-Supervised Learning Series Editor: M. Emre Celebi.”

[9] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Network anomaly detection: Methods, systems and tools,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[10] A. Fahad, Z. Tari, I. Khalil, I. Habib, and H. Alnuweiri, “Toward an efficient and scalable feature selection approach for internet traffic classification,” 2013.

[11] M. S. Srivastava, M. N. Joshi, and M. M. Gaur, “A Review Paper on Feature Selection Methodologies and Their Applications,” vol. 7, no. 6, pp. 57–61, 2013.

[12] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, “Foundations of Feature Selection,” Springer, Cham, 2015, pp. 13–28.

[13] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, “Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset,” *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5947–5957, May 2011.

[14] G. Doquire and M. Verleysen, “Feature selection with missing data using mutual information estimators,” 2012.

[15] S. Y. Jiang and L. X. Wang, “Efficient feature selection based on correlation measure between continuous and discrete features,” *Inf. Process. Lett.*, vol. 116, no. 2, pp. 203–215, Feb. 2016.

[16] M. Labani, P. Moradi, F. Ahmadizar, and M. Jalili, “A novel multivariate filter method for feature selection in text classification problems,” *Eng. Appl. Artif. Intell.*, vol. 70, pp. 25–37, 2018.

[17] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, “Building an intrusion detection system using a filter-based feature selection algorithm,” *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 2986–2998, 2016.

[18] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, Dec. 1997.

[19] P. Bermejo, L. De La Ossa, J. A. Gámez, and J. M. Puerta, “Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking,” *Knowledge-Based Syst.*, vol. 25, no. 1, pp. 35–44, Feb. 2012.

[20] Y. R. Shiue, R. Guh, and K. Lee, “Development of machine learningbased real time scheduling systems: Using ensemble based on wrapper feature selection approach,” *Int. J. Prod. Res.*, vol. 50, no. 20, pp. 5887–5905, 2012.

[21] A. Wang, N. An, G. Chen, L. Li, and G. Alterovitz, “Accelerating wrapper-based feature selection with K-nearest-neighbor,” 2015.

[22] T. Tekin Erguzel, C. Tas, and M. Cebi, “A wrapper-based approach for feature selection and classification of major depressive disorder-bipolar disorders.”

[23] M. F. Rafique, M. Ali, A. S. Qureshi, A. Khan, J. Y. Kim, and A. M. Mirza, “Malware Classification using Deep Learning based Feature Extraction and Wrapper based Feature Selection Technique.”

[24] H. Fu, Z. Xiao, E. Dellandréa, W. Dou, and L. Chen, “Image categorization using ESFS: A new embedded feature selection method based on SFS,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5807 LNCS, no. January, pp. 288–299, 2009.

[25] J. Zhao, L. Chen, W. Pedrycz, and W. Wang, “Variational Inference-Based Automatic Relevance Determination Kernel for Embedded Feature Selection of Noisy Industrial Data,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 416–428, 2019.

[26] S. Saito, S. Shirakawa, and Y. Akimoto, “Embedded Feature Selection Using Probabilistic Model-Based Optimization.”

[27] S. Shirakawa, Y. Iwata, and Y. Akimoto, “Dynamic Optimization of Neural Network Structures Using Probabilistic Modeling.”

[28] M. Viola, M. Sangiovanni, G. Toraldo, and M. R. Guarracino, “A generalized

eigenvalues classifier with embedded feature selection," *Optim. Lett.*, vol. 11, pp. 299–311, 2017.

[29] L. Liu, Y. Cai, W. Lu, K. Feng, C. Peng, and B. Niu, "Prediction of protein-protein interactions based on PseAA composition and hybrid feature selection," *Biochem. Biophys. Res. Commun.*, vol. 380, no. 2, pp. 318–322, Mar. 2009.

[30] O. Abedinia, N. Amjady, and H. Zareipour, "A New Feature Selection Technique for Load and Price Forecast of Electrical Power Systems," *IEEE Trans. Power Syst.*, vol. 32, no. 1, pp. 62–74, Jan. 2017.

[31] S. Solorio-Fernández, J. Ariel Carrasco-Ochoa, and J. Fco Martínez-Trinidad, "A new hybrid filter-wrapper feature selection method for clustering based on ranking," 2016.

[32] R. M. Mohammad, and F. Thabtah, and L. McCluskey, "An assessment of features related to phishing websites using an automated technique," *2012 Int. Conf. Internet Technol. Secur. Trans.*, pp. 492–497, 2012.

[33] W. Fadheel, M. Abusharkh, and I. Abdel-Qader, "On feature selection for the prediction of phishing websites," *Proc. - 2017 IEEE 15th Int. Conf. Dependable, Auton. Secur. Comput. 2017 IEEE 15th Int. Conf. Pervasive Intell. Comput. 2017 IEEE 3rd Int. Conf. Big Data Intell. Compu*, vol. 2018-Janua, pp. 871–876, 2018.

[34] J. Martínez Sotoca and F. Pla, "Supervised feature selection by clustering using conditional mutual information-based distances," *Pattern Recognit.*, vol. 43, no. 6, pp. 2068–2081, 2010.

[35] S. Zaman and F. Karray, "Features selection for intrusion detection systems based on support vector machines," *2009 6th IEEE Consum. Commun. Netw. Conf. CCNC*

2009, pp. 1–8, 2009.

[36] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electron.*, vol. 8, no. 3, 2019.

[37] A. P. Appel, H. Candello, and F. L. Gandour, "Cognitive computing: Where big data is driving us," in *Handbook of Big Data Technologies*, Springer International Publishing, 2017, pp. 807–850.

[38] M. Chen, Y. Tian, G. Fortino, J. Zhang, and I. Humar, "Cognitive Internet of Vehicles," 2018.

[39] A. Gliozzo, C. Ackerson, and R. Bhattarcharya, "Building Cognitive Applications with IBM Watson Services: Volume 1 Getting Started," vol. 1, pp. 1–130, 2017.

[40] K. Hwang and M. Chen E-Book, *Big-Data Analytics for Cloud, IoT and Cognitive Computing The definitive guide to successfully integrating social, mobile, Big-Data analytics, cloud and IoT principles and technologies*. 2017.

[41] Y. Hao, M. Chen, L. Hu, J. Song, M. Volk, and I. Humar, "Wireless Fractal Ultra-Dense Cellular Networks," *Sensors*, vol. 17, no. 4, p. 841, Apr. 2017.

[42] A. Fernández *et al.*, "Big Data with Cloud Computing: An insight on the computing environment, MapReduce, and programming frameworks," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 4, no. 5, pp. 380–409, Sep. 2014.

[43] J. E. Kelly, "Computing, cognition and the future of knowing.," *IBM White Pap.*, p. 7, 2015.

[44] W. Kinsner, "System complexity and its measures: How complex is complex," in *Studies in Computational Intelligence*, 2010, vol. 323, pp. 265–295.

[45] S. M. Prigarin, K. Hahn, and G. Winkler, “Estimation of Fractal Dimension of Random Fields on the Basis of Variance Analysis of Increments,” vol. 4, no. 1, pp. 91–102, 2011.

[46] T. Kaiser, “Node Localization using Fractal Signal Preprocessing and Artificial Neural Network,” 2013.

[47] K. Nahiyan, “Cognitive Unsupervised Clustering for Detecting Cyber Attacks,” 2020.

[48] P. Zhang, H. Barad, and A. Martinez, “Fractal dimension estimation of fractional Brownian motion,” in *Conference Proceedings - IEEE SOUTHEASTCON*, 1990, vol. 3, pp. 934–939.

[49] S. C. Liu and S. Chang, “Dimension estimation of discrete-time fractional Brownian motion with applications to image texture classification,” *IEEE Trans. Image Process.*, vol. 6, no. 8, pp. 1176–1184, 1997.

[50] “Multilayer perceptron - Wikipedia.” [Online]. Available: https://en.wikipedia.org/wiki/Multilayer_perceptron. [Accessed: 01-Sep-2020].

[51] T. Datta Chaudhuri and I. Ghosh, “Artificial Neural Network and Time Series Modeling Based Approach to Forecasting the Exchange Rate in a Multivariate Framework,” *Corresp. Author indranilg@calcuttabusinessschool.org J. Insur. Financ. Manag.*, vol. 1, pp. 92–123, 2016.

[52] K. Ferens, “Applied Computational Intelligence - Lecture: ANN_BP_v19,” in *Class notes*, Winnipeg: Dept. of Electrical and Computer Engineering, University of Manitoba, 2020, p. 34.

[53] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network

intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*, 2015.

[54] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J.*, vol. 25, no. 1–3, pp. 18–31, Apr. 2016.

[55] "Backdoor Attacks – Definition & Examples | Malwarebytes." [Online]. Available: https://www.malwarebytes.com/backdoor/. [Accessed: 19-May-2020].

[56] "Understanding Denial-of-Service Attacks | CISA." [Online]. Available: https://www.us-cert.gov/ncas/tips/ST04-015. [Accessed: 18-May-2020].

[57] "What is a denial of service attack (DoS)? - Palo Alto Networks." [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos. [Accessed: 18-May-2020].

[58] "Computer Exploit – What is a Zero-Day Exploit | Malwarebytes." [Online]. Available: https://www.malwarebytes.com/exploits/. [Accessed: 19-May-2020].

[59] "Cipher - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Cipher. [Accessed: 18-May-2020].

[60] P. Engebretson, "Chapter 2 - Reconnaissance," in *The Basics of Hacking and Penetration Testing*, P. Engebretson, Ed. Boston: Syngress, 2011, pp. 15–41.

[61] "Shellcode - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Shellcode. [Accessed: 19-May-2020].