Cognitive Machine Learning -- An Intelligent Approach for Dimensionality Reduction of Internet Datasets

by

Danish Kaleem

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

University of Manitoba, Winnipeg, MB, Canada

Abstract

High-dimensional data has always been a serious problem especially when the dataset has many irrelevant attributes. With the advancement of internet and cloud computing platforms, an exceptional rise has been recorded in the complexity of internet data attributes. Furthermore, in the domain of cyber security, modern data sets are highly disorganized and carry massive information to define a single event. Nonetheless, inspection of such dispersed high-dimensional data sets requires terrific human expertise and time. The contemporary machine learning techniques have great potential to deduce the relevant information from data sets, however, human cognition is always needed as an input to learning algorithms before training phase. Therefore, conventional models collapse in pruning redundant information from data sets due to the absence of a cognitive point of view.

This thesis proposes a novel fractal based cognitive model to reduce the dimensionality of two different internet data sets. The aim of the proposed research is to automate raw data attributes selection using ANN and cognitive aspects. Furthermore, the overall computational complexity of the proposed model has been reduced by pruning redundant hidden neurons of ANN. Hence, experimental results demonstrate that fractal based cognitive model selects only 7 relevant attributes from a dataset of 155, and shortlists 17 attributes from another dataset of 49 attributes. Moreover, hidden neurons pruning mechanism eliminates 108 useless neurons from a single hidden layer of 154 neurons while maintaining the maximum classification accuracy of 99.2%.

Acknowledgements

First and foremost, I wish to express my deepest gratitude to Prof. Dr. Ken Ferens for his supervision and guidance during my M.Sc. Study at the University of Manitoba. Without his valuable assistance, this work would not have been completed. Thank you so much! For being a wonderful teacher, advisor and a good friend.

I would like to thank my committee members for their interest in my research. In particular, I would like to appreciate Prof. Dr. Bob McLeod and Prof. Dr. Yang Wang for their precious time and interest in evaluating my thesis.

I am also thankful to University of Manitoba and Canada for providing me the opportunity to continue my higher studies in Canada.

Finally, I am thankful to Almighty ALLAH for His Countless Blessings, and giving me the opportunity, determination and strength to do my research.

Dedication

Every breath of my life and every drop of my blood is dedicated to my loving mother Nasreen Anjum. Thank you, mom, for all your love and support throughout my life.

Table of Content

List of Figures

List of Tables

Chapter 1 Introduction

In the domain of cyber security, high-dimensional data has always been a serious dilemma especially when the dataset has numerous irrelevant attributes. With the expeditious growth of internet and cloud technologies, an exceptional rise has been recorded in the complexity of data attributes. Furthermore, the number of attributes or feature space of advanced cyber threats is also changing with the seemingly unlimited growth of data logging technologies. Data engineers therefore have two major challenges to deal with. The first is thousands of raw data attributes and the second is rapid variations in the feature space of zero day attacks. Also, due to evolutionary growth of the internet, modern datasets are highly disorganized and carry massive information to specify an event only. For example, an event of email message can be described as an association of source to destination or destination to source IPs and addresses, routing and protocol information. To obtain this data, a number of mechanisms are used to record this information, e.g. packet firewall, client and server, operating system logs. Adding more complexity, cloud and host-based email software needs to be fully aligned for secure communication and hence, epochs, timestamps and other associated attributes are shortlisted to preprocess data. Moreover, artificial intelligence, autonomous computing algorithms, knowledge-based processors and other intelligent methodologies are implemented by existing technologies for different applications.

Today, data logging platforms are competent enough to collect every possible data attribute associated with a single event. All these attributes carry unprocessed data and mostly consist of redundant information regarding an event. Moreover, this raw data can be utilized to obtain discriminatory information. In addition, eliminating redundant raw data fields would not miss any event related information. For instance, in internet data sets, the attributes of packet count between source to destination and bytes transferred between source to destination are actually the same.

Both fields contain same information and eliminating one of them would not result in the loss of any information. In the domain of cyber security, for internet data sets, it is always a good practice to shortlist a subset of those data fields which carry maximum information about the malicious threats. To select this subset, one needs to have knowledge about the association or dependency of various data attributes to the particular attack or event. In the industry, data experts or engineers extract a list of relevant features based on data processing experience and, most importantly, their domain knowledge. At this stage, another question can be raised that why selecting subset of fields? Why not consider all of them? Notwithstanding, considering every attribute is problematic for a number of reasons. The first reason is that training all attributes or fields will append noise of unnecessary or useless fields to the shortlisted relevant subset of fields. Secondly, an increase of dimensionality of data attributes will not only lower the learning accuracy but also adversely effects comprehensibility. And finally, the impact of important fields will be reduced by the presence of irrelevant attributes which inject the factors of biasness and higher order statistical variance into the training phase. Thus, precise treatment and training of attributes is required to extract primarily high information containing variables to achieve greater accuracy in the classification of threats.

Notwithstanding, the time for big data has come; the enormous collection of contemporary data sets is so complex that it has challenged the traditional statistical and machine learning techniques [1], [2]. For example, a few specific features of big data - enormous, non-homogeneous, high-dimensional, disorganized, complex, irregular, noisy, imprecise etc. have severely argued conventional statistical analysis techniques, which were ideally introduced for inspecting standard or relatively small data. Further, in the domain of cyber security, the feature space of zero-day threats is so complex that they can't be properly analyzed by conventional statistical methods (e.g. correlation based statistical analysis or classical linear regression analysis) originated on the basis

of presumed or assigned distribution of data. Contrarily, contemporary cognitive learning techniques are quite data oriented and produce more reasonable or practical solutions [3]. In addition, cognitive analysis of data sets is needed before feeding it to learning algorithms which is also known as the preprocessing of a data set. A cognitive inspection is composed of three major components; field knowledge, experiential learning, and complexity analysis [4]. More precisely, it is a combination of various techniques to analyze different challenges in the same way as humans do. For instance, in a cricket match a batsman always carries three cognitive aspects, which are to analyze the ground pitch, weather (e.g. moisture), and opponent bowlers pace or strike rate. This is also known as complexity analysis. Then, based on his cricket knowledge and experience as a batsman (i.e. domain or field knowledge), the player would continue his turn. So, in the processing of enormous data sets, the incorporation of three cognitive ingredients is required to feed it to learning algorithms. As a result, conventional models fail due to absence of cognitive aspect.

In recent years 'Cognitive Machine Learning' as a developing multidisciplinary domain of artificial intelligence has been extensively acknowledged in various complex and enormous data-oriented fields such as astronomy, engineering, marketing, and cyber security in order to extract hidden information of interest in data. Today, with assistance from advanced computational techniques, complex data analysis using artificial neural networks has received great attention, especially in the cyber domain where dimensionality is known to be a curse [5]. Furthermore, the basic ingredient of artificial intelligence research is neural networks [6]. "Cognitive neural networks + big data" is becoming a driving force of invention, communal advancement, and contemporary research establishment. An ideal and mutually supportive chemistry of neural networks and big data can be evidently elaborated by inspecting the primary concepts and engineering principles in big data along with the structure of neural networks. On one side, artificial

neural networks have enough potential to choose the most relevant attributes from data sets. To capture quick changes in the feature space, neural networks incorporate multiple techniques to transform and preprocess non-homogeneous data. On the other side, 'high-dimensional + large volume' data provide sufficient samples to train neural networks properly. 'Cognitive neural networks + big data' is also not free from technical complications. From the point of view of neural networks, the architecture of hidden neurons and hidden layers requires further inspection and advancement [7]. The solution for network volume requires more cognition and logical guidelines to fix some inherent problems in the learning algorithms. From the data point of view [8], there are also three main technical problems; 1) how to maintain consistency in rapidly changing feature space; 2) how to extract relevant information as per domain knowledge and 3) how to portray temporal dependency. Far more research and inspection in this domain is still required from every possible intellectual and factual aspect. More collaboration with cognitive informatics, natural intelligence and computational intelligence is required to fix the core technical complications in neural networks research and big data, in order to improve the research of big data analysis using neural networks.

1.1 Thesis statements

In this thesis, we introduce a fractal based cognitive model to extract important attributes from two different internet data sets. The proposed algorithm is called" Sensitivity based pruning algorithm (SBP) which selects the most relevant attributes based on the sensitivity value of each field. The core idea of the SBP scheme is to train an artificial neural network by feeding preprocessed internet data set and calculate the minimum mean squared error (MSE), which is essentially the local minima of the error curve of the data set. Then, we replace each attribute with

the mean value of all the samples in that attribute and again calculate the MSE by testing the earlier trained network. This process is repeated for all the attributes in the data set and the MSE is found. Sensitivity is then calculated for each field by subtracting the MSE of the actual data set from the MSEs of all the replaced attributes. Based on the magnitude and sign of these values, we rank each attribute as important or redundant. Similarly, we calculate the sensitivity of each attribute by replacing ‘standard deviation’, ‘random numbers’ and most importantly the ‘box counting fractal dimension. Hence, experimental results demonstrate that the proposed algorithm prunes redundant attributes cognitively.

After pruning raw internet data sets, we propose another methodology called the Hidden neurons pruning mechanism (HNPM) to prune hidden layer of neural network. The proposed scheme not only reduces computational complexity significantly, but also determines the required number of hidden neurons while maintaining the maximum classification accuracy. The primary idea of the HNPM scheme is to train a neural network with a maximal number of neurons in the hidden layer. The first step is to record the higher-order features by feeding each training example one after another. Once the new data set is recorded, feed it to another 2-layers network by keeping the same weights of the network trained earlier and apply SBP algorithm again to prune higher-order features. Using this cognitive approach, we prune redundant hidden neurons by up to 2/3 of the maximum neurons while maintaining maximum classification accuracy.

1.2 Contributions of thesis

a) In this thesis, we have proposed the application of a box counting fractal dimension algorithm to extract important attributes of two different internet data sets. The experimental results confirm that cognitive selection of attributes using domain knowledge, experiential learning and complexity analysis can be achieved using the proposed fractal based cognitive model.

b) In this thesis, we found that a single scale analysis using mean, standard deviation and random numbers doesn't provide significant value of sensitivity measured to select relevant fields while multi-scale analysis, (i.e. a fractal-based approach), produces discriminatory for selection which conforms to the human analytical method.

c) In this thesis, the proposed algorithm not only prunes raw data attributes and hidden neurons (higher-order features) but is also capable of determining the actual size of the hidden layer needed for a neural network application while maintaining the maximum classification accuracy.

d) In this thesis, the SBP algorithm has been experimented on two different internet data sets (wireless and IP-based). The performance of the SBP algorithm demonstrates that this methodology is potentially capable of pruning any data set regardless of the nature of data.

e) We have studied literature in detail and found that the fractal based cognitive model is more precise and can achieve higher classification accuracy when it comes to detecting malicious threats as compared to other techniques.

1.3 Outline of the thesis

The organization of this research thesis is arranged as follows: Chapter 2 presents a literature survey of available dimensionality reduction techniques including feature selection and extraction strategies and their potential advantages or disadvantages. Chapter 3 provides the background of cognitive machine learning such as biological neural networks and artificial neural networks and also provides a brief description of fractal dimensions and multiscale analysis. Chapter 4 demonstrates two diverse internet datasets used in the experimentation of the proposed research and also discusses the statistical and feature space analysis techniques applied on both datasets. Chapter 5 presents the proposed fractal-based cognitive model for dimensionality reduction of internet datasets. Further, it presents the HNPM technique to eliminate redundant neurons in the hidden layer of artificial neural networks. Chapters 6 elaborates the results from the experimental works. It explains the cognitive aspect of the proposed methodologies using feature space diagrams. It also elaborates the classification accuracy and generalization capability of HNPM technique. Furthermore, this chapter compares the results with respect to first and second order statistical moments and multiscale fractal dimensions. Finally, Chapter 7 concludes the thesis and provides the scope of works which can be done to reduce the redundant dimensions of convoluted complex internet datasets in the future.

Chapter 2 Related works

In the modern age, dealing with high-dimensional data is the most painful job for scientists and analysts since the data is generated by different collection devices, methods and techniques. The curse of dimensionality downgrades the performance of ML algorithms by causing under fitting, overfitting, time delay in developing ML modes and reducing the classification rate since the massive and unprocessed data has more noise and carries redundant information categorically [9]. The set of irrelevant features doesn't take part in the learning process and the set of redundant attributes carries same the information, so both misdirect the learning mechanism which consequently degrades the overall performance. Therefore, dimensionality reduction as a preprocessing stage to ML is an efficient approach to improve generalization capability, learning accuracy, and result comprehensibility.

Feature Selection and Feature Extraction are the most practical approaches to reduce the dimensionality of massive and raw datasets. In the domain of cyber security, feature selection is a process of selecting only those input variables that contain compatible information with respect to happening attacks [10]. Feature extraction, however, is the process of transforming input space onto a low-dimensional subspace to retain significant information with respect to particular attack [11]. Both techniques are used to enhance the potential of ML algorithms with reference to authenticity and time to develop patterns. Usually, attributes are of three types as: 1) important, 2) useless and 3) redundant. In the mechanism of feature collection, the best subset of relevant attributes is shortlisted from the available set of features [12] (see Figure 2.1 for more details). Fewer dimensions and more contribution towards learning accuracy are the most salient features of the best subset. The main benefit of a selection mechanism is that crucial information of any

variable is accessible or available. However, if any subset of attributes is needed and all available variables are distinct then there is always the possibility of data being lost since a few of them must be eliminated. In case of extraction, however, transformation of high space onto low subspace can be made without losing a single piece of information. The main disadvantage of the feature extraction technique is that linear combination of available data attributes is mostly not understandable, and the information of relevant/important attribute is lost [13].

Research literature [14], [15], [16], [17], [18] proves that many attempts have been implemented so far to discover the best selection or extraction techniques. Some of the most credible approaches are: PCA, ICA, non-linear PCA, mRmR, BW-ratio, Genetic Algorithms, Neural Networks-based pruning, SVM-REF, RELIEF, Sensitivity-based selection and Correlation-based selection etc. In the aspect of extraordinary number of current feature selection and extraction techniques, it is extremely important to decide the selection criteria of an algorithm before employing it in a certain situation. This chapter provides the detailed literature review of possible feature selection and extraction techniques. In this study, we also compare the performance of the proposed feature selection mechanism with other existing potential techniques with respect to their advantages and disadvantages.

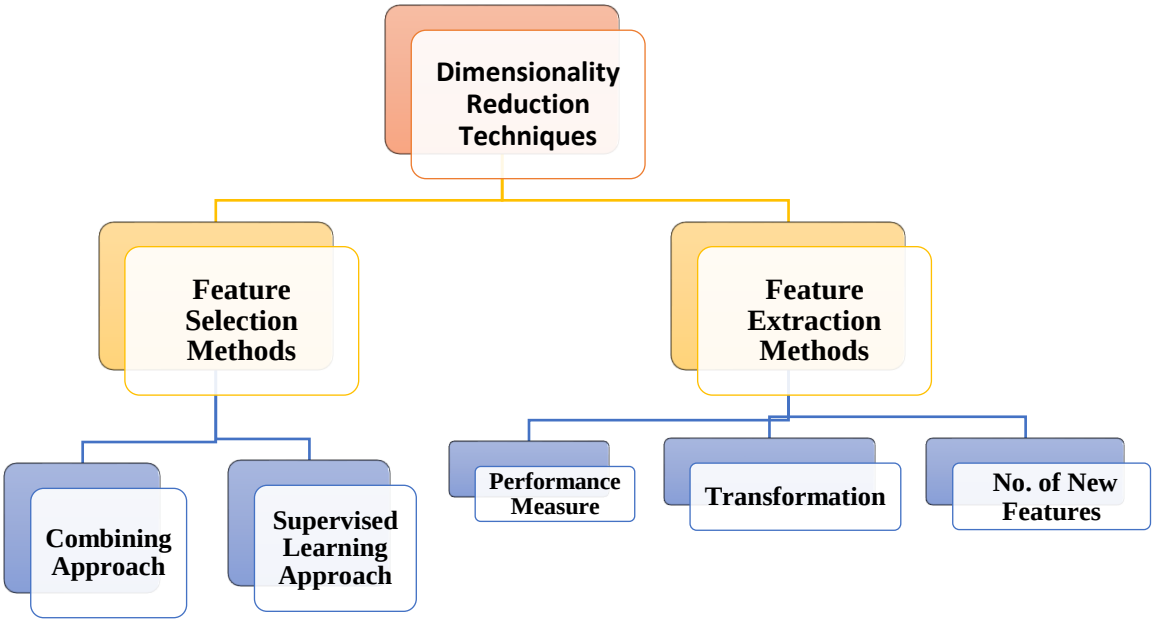

Figure 2.1 Breakdown of dimensionality reduction techniques.

2.1 Feature selection techniques

Feature selection mechanism is usually categorized into two sections: feature subset selection methods and feature ranking methods. The process of feature subset selection produces possible combinations of the subsets by using different searching techniques e.g. GFS and GBE (Greedy Selection and Reduction) etc. to measure due potential of the subsets with respect to variance, correlation, consistency etc. This approach always has a major disadvantage of space and computational complexity because of the generation and inspection of numerous possible subsets. Furthermore, in the process of feature ranking methods each attribute is graded with the help of a selection metric, (for example gain ratio, sensitivity, information gain, symmetry information, and linear dependency) and then a set of high ranked attributes is shortlisted based on the threshold

value (which is already predefined). Logically, the feature ranking approach is better than the feature subset approach with regards to space complexity and computational cost. Moreover, the mechanism of selecting salient attributes is also categorized into 4 types: embedded, filter, wrapper and filter-wrapper (hybrid) approaches depending the supervised learning-based selection of features.

In [12], Ladha highlights the general set of advantages of possible feature selection techniques:

- Feature selection techniques reduce the high-dimensional space to enhance the speed of learning methodologies.
- These techniques improve the quality of data by pruning irrelevant/noisy attributes.
- They reduce computational complexity and increase results comprehensibility.
- The selection methods improve the potential/capability of learning algorithms and increase the classification accuracy.
- These techniques make the high spaced complex data more understandable.

2.1.1 Features combining based selection

This portion provides the details of two novel approaches for combining the features from an estimation point of view. Feature subset and ranking based approaches select the best set of features from a dataset that highly correlate with the recorded attacks.

2.1.1.1 Feature selection based on searching strategies

In the process of searching based selection, all possible number of feature combinations are generated using different searching strategies and then, all possible subset of attributes is evaluated using supervised learning algorithms (SL) or statistical analysis. Finally, the most relevant set is shortlisted for further inspection. It is important to note that the set of features assessed using SL algorithm can also be called as the wrapper-based selection approach.

In [19], Hall has developed the correlation-based feature subset selection mechanism which is so far the best example of selection methods. In this approach, the selection metric to evaluate the potential of the subsets is consisted on two parameters: feature-class correlation and feature-feature correlation. The proposed model generates different subsets of the attributes. Then each combination is assessed by the selection metric. A subset which shows less feature-feature correlation and maximum feature-class correlation is considered to be the most significant subset of features. In [20] authors have proposed consistency-based subset feature selection mechanism. This approach applies statistical measure (consistency) as a selection metric to shortlist the significant subset of features. The proposed approach uses the filter-based methods since supervised learning algorithms are not employed in the COFS method.

An in-depth/exhaustive search usually produces 2 raises to the power of N subsets of features from a list of N number of attributes for assessment. Thus, the in-depth searching mechanism has high cost in terms of computational complexity. However, heuristic searching approaches are quite popular and have been employed by many researchers since these generate a relatively small number of features subsets for evaluation. The best heuristic searching strategies

are: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA), Ant Colony Optimization (ACO), Tabu Searching (TS) etc. In [21], researchers employed the SA searching approach and found that the performance of SA is better than other exhaustive searching strategies. In [22] authors employed SA, and evaluated the selection approach by using supervised learning algorithms (BPN-Back Propagation Network) to obtain better network architecture. Authors in [23], shortlisted the best subset of features using TS (Tabu Search) and then evaluated the mechanism using the ensemble classification algorithm to find the most relevant subset of features. A significant number of feature pruning techniques adopt the Ant Colony Optimization strategy for acquiring the best group of features. In [24], authors use ACO searching technique to find the best subset using nearest neighbor classifier by using customer review dataset. In certain feature selection research work, a Genetic algorithm or a Particle Swarm Optimization algorithm have also been used as searching tools in different applications. For example, in [25], [26] and [27] researchers have employed GA and PSO in the application of handwritten digit recognition, land cover classification, and sleep disorder diagnosis system.

In the literature of searching based selection, we have noticed that both exhaustive and heuristic strategies lead to high computational complexity. An exhaustive search generates 2 raised to the power of N subsets of features which can't be suitable for massive/complex datasets. On the other hand, heuristic approaches need previous information and each produce group of features required to establish a classifier in order to estimate and to achieve the best subset. So, both approaches are expensive and cannot handle a massive raw dataset.

2.1.1.2 Feature selection based on statistical measure

In the process of feature selection based on statistical measure, each attribute is evaluated by some statistical or information-theoretic measures. Once an attribute is evaluated, it is graded based on its sensitivity level. The features with more sensitivity value are then shortlisted using a certain threshold value (which is already predefined). The rank of an attribute is directly proportional to the significance of that attribute. A high rank means greater relevance/importance of an attribute in a dataset. Chi-Square-based feature selection (CSBF) is the best example to demonstrate the ranking-based feature selection approach. In [28] researches used the CQFS mechanism to grade the most significant set of variables for cancer classification. Similarly, in [29], authors used the gain ratio, (or information gain, etc.), as an analysis tool to measure the rank of features.

In the literature of rank-based pruning, we observed that researchers use statistical or information measures to rank individual attributes by evaluating the dependency among variables and target class/attack. Although this approach is significantly cheaper from the perspective of computational cost, the majority of the algorithms fail to prune redundant variables [30]. The ranking-based approach will be known as a filter-based pruning technique if it doesn't employ supervised learning algorithms. We have also observed that some researchers claim that ranking-based methods are independent of the ML algorithms and therefore achieve more comprehensive results.

2.1.2 Learning based selection

This portion provides the details of dimensionality reduction techniques mainly based on the learning algorithms. It is classified into four methods: Wrappers, Filters, Embedded and Hybrid methods.

2.1.2.1 Wrapper-based methods

As demonstrated by Figure 2.2, these methods produce a subset of features by using any of the available searching strategies and measure the potential of each subset using SL algorithms (Supervised learning) with respect to the detection rate [31]. Researchers in [32] have proposed a wrapper-based method to shortlist the best features from the dataset. The proposed methodology is known as 'search engine', which generates possible subsets as well as evaluates their performance. Experimental execution has been compared with hill-climbing and heuristic search strategies using Naïve Byes classifier. The authors have found that the wrapper-based approach has the complications of overfitting, searching overhead and unnecessary time delay.

In [33], researchers have proposed a wrapper-based model by using SVM (Support Vector Machine) with kernel function. They have used the SBFS (Sequential Backward Feature Selection) method for producing combinations of subsets, which are evaluated w.r.t detection accuracy. To reduce the complication of searching overhead, authors at [34] have developed a novel searching methodology called SFS-LW to rank the features w.r.t their replacing errors to classify their redundancy. Researchers have also developed a constructive wrapper-based feature selection model using Artificial Neural Networks (ANN) in [35]. They successfully eliminate the irrelevant noisy data by using a correlation metric to enhance the learning speed of ANN. Furthermore, to

reduce the searching overhead authors developed a wrapper model using GA with SVM to analyze hyper-spectral images [36].

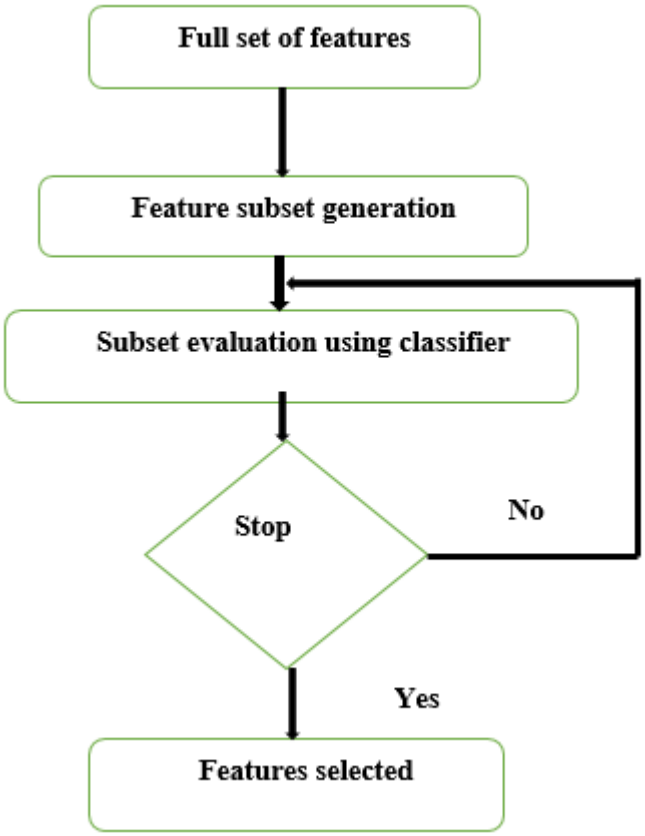

Figure 2.2 Wrapper-based features selection algorithm.

In wrapper-based feature selection methods overfitting is another major problem. We have observed that many researchers overcome this issue by proposing jitter, early stopping, and post pruning methods. In the jitter approach most irrelevant sets of features which make the computational learning growth complicated are eradicated to fit the training data properly. Post pruning is executed by establishing the decision tree. Furthermore, the early stopping criteria is usually achieved by adjusting the iteration parameters of ANN and GA [37], [38].

In the literature of wrapper-based feature selection we have observed that searching overhead, overfitting and unnecessary delays can be minimized only up to a certain level. We also observed that they suffer from greater computational complexity since they employ SL algorithms. Therefore, wrapper-based feature selection methods are not suitable for high-dimensional massive raw datasets.

2.1.2.2 Filter-based methods

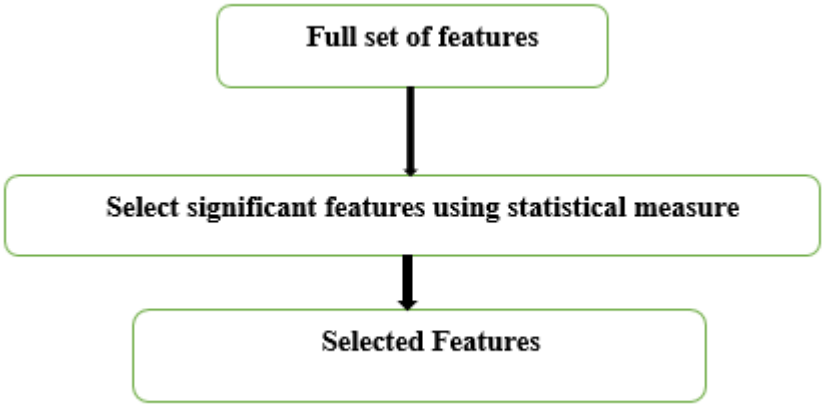

Figure 2.3 Filter-based features selection algorithm.

As per the algorithm demonstrated in Figure 2.3, filter-based methods don't rely on SL algorithms and therefore improve overall generalization capability of the model. In terms of computational work filter approaches cheaper as compared to wrapper or embedded-based feature selection techniques. One of the best examples of filter methods is RELIEF, which was proposed by a research group and is explicitly explained in [39]. The proposed approach calculates the weight of each feature in terms of correlation with the attack class. The major drawbacks of RELIEF are: 1) it can only handle 2-class problems. 2) It is not capable enough to prune redundant

attributes. The researchers later proposed a better version called RELIEF-f [40]. The updated version is intelligent enough to deal with multi-classes as well as irregular/noisy data, however, it is still unable to eliminate useless attributes. In [41],a single rule-based selection model was developed. The model implies a single rule for each variable and selects all those attributes that produce the smallest classification error. In [42], authors developed a mutual-information based model to select a salient subset of features. The proposed approach measures the mutual-information value among each feature and target class. The higher the information, the more significant a feature is. To summarize, each variable/feature is ranked, and the subset of attributes which carries the most information is selected.

Recently, researchers have developed clustering techniques for feature selection. In [43], the proposed clustering scheme determines the similarity or dependency among the variables and removes any useless sets of features. Another group proposed a mechanism which is mainly based on the information theory. This approach is the best example of text classification which clusters the available text information based on the similarity/dependency measure [44]. In [45], the clustering algorithm is merged with the Chi-Square metric to choose variables from an empirical dataset. One group also developed a multi-spectral clustering mechanism in [46] to discriminate important and less important attributes. In [47] authors incorporated the idea of joint mutual information with the supervised clustering approach to evaluate the potential of each attribute in terms of its relevancy measure.

In the literature of filter-based feature selection methods, we have inspected that this approach has enough potential to deal with the high-dimensional massive raw datasets in terms of computational cost and the generality of results.

2.1.2.3 Embedded-based methods

Embedded methods are completely dependent on the learning phase of SL algorithms. These methods employ a major section of the learning course of contemporary SL algorithms to select significant features. Their performance in terms of computational work is much better than wrapper methods [48]. Embedded-based models are usually classified into three categories. In the first category, it feeds all possible features of a given dataset into the learning algorithm to build the classification model. After this step, all the variables with an insufficient correlation coefficient are eliminated using SVM. In the second category, also known as the built-in mechanism, a bit of the learning module of C4.5 and ID3 is used to shortlist best variables. Finally, in the third category, classification errors are reduced and variables with less or close to zero regression coefficients are pruned.

In [49], researchers proposed an embedded-based model to shortlist the relevant set of features from the real-world dataset. In their methodology variables are selected using linear and non-linear Support Vector Machine and real-world benchmark datasets are used to obtain better predictions. Another group in [50] have used an embedded-based method in the application of audio signals. They developed a novel mass function using the principal of evidence theory, and finally selected the most significant variables one after another in each iteration. Furthermore, in [51], authors have reported the performance of embedded-based methods in an unbalanced dataset. They have concluded that multiple objective functions can only enhance the functionality of diverse and massive datasets.

In the literature of embedded-based methods, we have examined that these methods are more computationally efficient than wrapper methods in terms of high-dimensional space, but

don’t perform well in comparison to filter methods. We also observed that the majority of the researchers pay more attention towards filters when it comes to choosing the better option between embedded and filter methods.

2.1.2.4 Hybrid methods

Hybrid-based models merge the filter and wrapper-based methods to select feature from a dataset. It is usually a hard task to deal with the high-dimensional data with wrappers. Therefore, authors in [52] have proposed a technique called the filter-wrapper feature selection model. In filter-wrapper method the authors first statistically analyze the available dataset using a filter model and rank each attribute with respect to their weight. They then feed the list of ranked/weighted attributes to the wrapper model to train the supervised learning algorithm. Another research group applies the hybrid method on a medical dataset. They first shortlist a subset of important attributes using filter methods and then feeds the selected subset into the wrapper model. The combination of both methods successfully discriminates cancerous genes [53].

In [54], authors experimented by using the hybrid approach on a real-world dataset using information gain as the statistical measure and SVM as supervised learning algorithm. In [55], they choose relevant features using a mutual information metric and then apply ANN as the supervised learning algorithm. Another group apply the same combination of filter-wrapper methods for text classification. They experiment information gain to shortlist variables and then feed them to SVM learning algorithm.

In the literature, we deeply examined the performance of hybrid methods and found that the hybrid approach is less expensive than other mentioned techniques. We also found that these methods have less generality as compared to filter models.

2.2 Feature extraction techniques

Feature extraction is a technique which applies transformation on the original set of features in order to produce other high-order significant features. By definition, "it is a technique used to generate discriminatory set of features by constructing linear combinations of the available raw data features" [56]. A critical issue in the various disciplines of Artificial Intelligence (AI) is to find the appropriate description of multivariate data.

Feature extraction techniques can also be considered as complexity reduction approaches since they provide simple representation of data in terms of linear combination of input/low-level attributes. The most popular and best example of the feature extraction technique is Principle Component Analysis (PCA). In this section, we study the brief description of the most popular and common examples of feature extraction techniques such as PCA, ICA, PPCA, KPCA etc.

2.2.1 Principle component analysis

It is an open limit statistical approach used to discriminate the significant data from noisy/raw data. It is also defined as an approach to transform the raw data linearly in order to reduce repetitions and to magnify the relevant information [57]. Another group defines [58] PCA as a statistical analysis tool which employs orthogonal conversion to modify patterns of correlated variables into linearly uncorrelated variables. Hence, new generated variables are known as Principle Components, which are similar to first-order moments. It is important to note that PCA is a pure Un-Supervised extraction technique which doesn't require a labelled dataset. If data follows a normal distribution, then the produced set of new components is independent by nature. It has been observed that PCA prunes the initial number of attributes by removing the last

components which don't play any role in the observed change. It employs covariance and variance metrics to increase or decrease the relevant information.

Generally, Principle Components have two salient characteristics: 1) every new generated component is formed by linearly combining initial attributes 2) all components are independent, and their mutual correlation measure is zero [59]. Data reduction/compression, time series analysis, image analysis, visualization, and regression etc. are the applications of PCA. The major drawbacks/ limitations of this technique are [60], [61]:

- It assumes that Principle Components are a linear combination of original variables.
- It also assumes that all components are orthogonal to each other.
- It employs variance as a statistical measure. This means that high variance axes are considered to be components and the other axes are considered to be noise.
- It doesn't provide sensible results if all the features are not scaled at one numeric level.
- The lack of probabilistic structure is another limitation when it comes to the models Bayesian decision strategy.

2.2.2 Independent component analysis

Another popular example of a feature extraction technique is Independent Component Analysis (ICA). It is a computational approach used to separate multivariate data into additive subcomponents [62]. It assumes that all the involved subcomponents follow non-Gaussian distribution and are statistically independent to each other [63]. ICA is also known as 'Blind Source Separation'. The Cocktail party problem is the best example of the ICA analysis technique [63].

ICA has two main classes: one class of algorithms are dedicated to minimize the mutual information (MI), and the other class is used to maximize the non-gaussian distribution. MI can be interpreted in terms of reduction of irrelevancy of a feature A after observing feature B. Therefore, such algorithms which are dedicated to reducing mutual information are actually searching out the maximum independent components.

Another approach to evaluate the independent components is the non-gaussianity of each feature. Components are usually extracted by forcing them to be a distant from normal distribution as possible. The major limitations of the ICA analysis technique are:

- It assumes that original features are independent.
- The number of original variables must be equal to number of mixed variables and the observed mixture shouldn't depend on a variable.
- The mechanism/environment should be less noisy.
- Collected data should be geo-centric by nature.
- Original signals shouldn't follow Gaussian distribution.

2.2.3 Non-linear component analysis

To address the limitations of principle and independent component analysis techniques, Guttman proposed another model called as the non-linear component analysis technique [64]. It has the same targets and performance as PCA but is appropriate for features scaled at different numeric levels. In this approach all features are classified into various clusters and every single sample of a feature belongs to certain category [65]. The most significant advantages of non-linear

component analysis are that it combines both nominal and ordinal features and it is capable enough to extract the hidden relationship among various features and their components. It can also handle each feature at an adequate inspection metric. The main difference in terms of execution among linear and non-linear models is that in PCA features are directly evaluated, whereas in non-linear models measured features are quantified during analysis. The most powerful advantage of this approach is the usage of optimal quantification (conversion of categorical data to numerical values). The limitations of non-linear component analysis are: 1) non-linear models are instable or non-unique as compared to linear models 2) they are highly dependent on the dataset are therfore ineffective if the dataset is short or noisy 3) non-linear models are unable to reproduce missing values in a high-dimensional space

2.2.4 Probability principle component analysis

This is another advancement which addresses the third limitation by endorsing noise components to form isotropic structure [66]. The PCA is essentially incorporated as a part of the learning stage for the probability Principle Component model using probability estimation function. Another important expectation algorithm is incorporated in this model, which learns parameters sequentially. The main advantages of PPCA are:

- It incorporates multiple PCA models.
- It possesses the application of Bayesian methods.
- It is capable enough to reproduce missing data values.
- It incorporates statistical measure.
- It can be used as a constrained Gaussian density model.

Chapter 3 Background on cognitive neural networks and fractal dimensions

This chapter provides the details on the cognitive nature of Artificial Neural Networks (ANN). The first section describes the biological neurons and their similarities to the ANN model. The second section explains the overall structure of ANN and briefly elaborates on their non-linear relationship in terms of mathematical equations. The following sections demonstrate fractal dimensions, as well as single scale and contemporary multi-scale analysis techniques.

3.1 Cognition and cognitive informatics

As per the literal meaning of cognition, "it is a process of understanding, recognizing or conceiving". In literature, "it is defined as an incorporative research of fundamentals of intelligence through an artificial mechanism renowned as learning by understanding" [67], [68]. More precisely, it is defined as a set of techniques to examine/audit various phenomena in the same way as humans do [69], [70]. In simple words, cognition is an approach for replicating humans' comprehension.

Over the last decade, cognitive informatics has emerged as a mature development in artificial intelligence and has now turned into a most powerful mechanism to improve the potential of traditional ML algorithms. The literature study specifies that an extensive research attempts have been conducted to understand the humans process of thinking and the process of how humans inspect, study, learn and sense. The formal definition is as follows: "cognitive informatics is an integrative class of cognition and information science, which examines the core processing structure of human brain as well as their constructing operations of intelligence" [71].

Cognitive science is an attracting research field directed towards engineering applications by combining the knowledge of cognitive computing and informatics [72]. This involves inspecting different strategies to study human mental processing abilities and using that knowledge to advance/enhance artificial learning models. This includes offering increased understanding, improved recognition and extended comprehensive capabilities in contemporary engineering concepts and mechanisms to discover more reliable and robust solutions. It is accomplished by mimicking strategies and associations which are accelerated in a human processing unit because of an activity. The overall procedure requires the transplant of theories and multi-channel domain knowledge including but not limited to advance deep machine learning, multi-scale fractals and chaos engineering, hybrid statistical signal processing analysis and wavelets analysis.

The core idea of research in this field is to enhance the adaptive learning capabilities of the conventional ML algorithms. In this research work, we first focus on the cognitive analysis to preprocess datasets before feeding it to neural networks and deciding the optimum number of features. Further, to improve the learning ability of ML algorithms, we shortlist three components of cognition: Domain knowledge, Complexity measure and Experiential learning. Artificial neural networks are mostly treated as complexity extraction methods since every hidden layer of neurons approximates high-order features. There are various approaches to define cognition in terms of neural networks but for the sake of this work, we mainly target the cost function of a neural network as a function of input variables and their correlation with one of more attributes with respect to the given target class. We deeply observe that the use of neural networks is equivalent to using experiential knowledge by humans to take certain intelligent decisions. Fractal dimension is another powerful tool employed in this research work which employs multiscale analysis to extract

hidden information of interest among features of a dataset. Categorically, fractal dimensions are considered to be an indicator of long range correlation in an object.

3.2 Crude computational model

The crucial component of the biological neural network is a neuron. A typical human brain is made up of 86 billion neurons and each neuron consists of four parts; Soma (cell body), Dendrites, Axon, and Axon terminal. The cell body is used to execute non-linear convoluted procedures. A bunch of input trails used to collect input signals from contiguous neurons are called dendrites. A comparatively long trail which carries away the output signal from the cell body is known as the axon, and the extreme end of each axon which connects neuron to another neuron is called axon terminal/synapse.

The fundamental intention of a neuron is to clip the incoming signals (in the form of chemical and electrical signals) and to decide whether to transmit the electrical signals to adjacent neurons or not. Generally, the neurons either receive information from the other peripheral environments or from the neighboring neurons. In the cell body, all the incoming signals are combined, and the overall summed value is calculated. The neuron takes a decision of sending an output signal based on the threshold of the summed value. It is important to note that neuron can behave as a transceiver (capable of sending/receiving chemical and electrical signals at the same time). When the summed value of stimuli is greater than the threshold value, an output signal is sent which passes through the axon and reaches to the axon terminal. The extreme end of the axon discharges chemically encrypted information after the output signal reaches it. The chemically encrypted information is also known as neurotransmitter since it release the encrypted chemical signals. These encrypted signals communicate/pass the information to the axon terminals of the

adjacent neurons. Generally, it is almost impossible to measure the size and substantiality of this transmitted chemical, but one can easily determine the feedback of a receiver which is either an excitatory or an inhibitory axon terminal based on the peculiarity of the receptor. It is important to note that if the collected information exceeds a certain threshold it would trigger the receiver neuron and enforce it to release an excitatory signal to the axon terminal.

3.3 ANN architecture overview

By definition, an ANN (Artificial Neural Network) is a hypothetical learning-based model that has developed from the domain knowledge, behavioral attributes, complexity configuration and most importantly decision-making capability of the fundamental unit of a human central processing part known as a neuron. The overall objective of this model is to mimic the core behavior of the human brain. An ANN model is also made up of several hidden neurons to learn the behavioral complexity of the input data samples. So, compared to other traditional learning techniques, an ANN has the capability to model complex non-linear relationships. Furthermore, it is also capable of interpreting complex datasets, can be used to detect/reveal such patterns that are not commonly noticed by data experts. In other words, a trained ANN model can be considered as a data expert, can be used to analyse the given information (raw dataset) intelligently and autonomously.

A multi-layer perceptron is made up of at least three layers; the input, hidden and output layer (as described in Figure 3.1). Every layer is made up of m neurons (where m>1) and each neuron of any layer is linked with all the neurons of the neighbouring layers. A bias or a constant number is also connected to each neuron. There are two types of connections between neurons: first layer connections and second layer connections. Each connection has its individual weight

which is multiplied to the neuron value of the previous connection layer. Number of input and output neurons is determined by input and output data dimensions. Finding the appropriate number of layers and hidden neurons, however is an open-ended research problem which is usually decided heuristically.

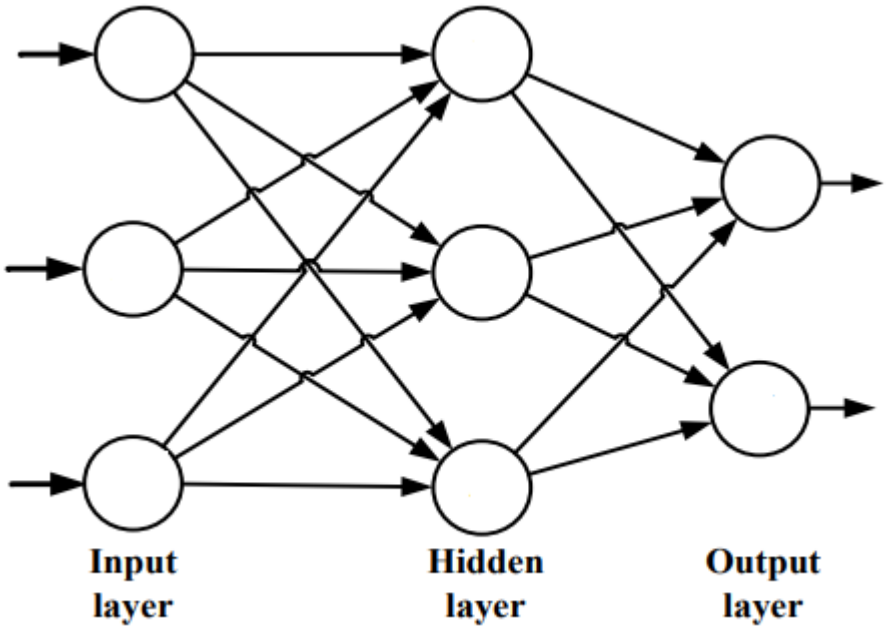

Figure 3.1 Artificial neural network architecture.

Multi-layer perceptrons and ANNs are usually known for their two distinguished features. The first is high generalization capability and the second is their tolerance against faults. In the context of generalization, ideally, the neural networks have the ability to handle unseen patterns from other known patterns that contribute the same distinguishing features. In other words, noisy or irrelevant data will be distinguished based on their differences with processed or smooth data. However insufficient training samples or overfitting can badly affect the generalization capability too. Furthermore, high tolerance against faults is another distinguished feature of neural networks. The adaptive behavior of neural networks helps them to learn the patterns continuously even when

a major part of the network (including neurons and their interconnections) stop learning patterns. Further learning is an encyclopedic term. A learning mechanism modifies itself to accommodate circumstantial changes. An ANN model can learn in any perspective however the question of how to execute it will always arise. Theoretically, an ANN could learn by:

- Organizing new (state-of-the-art) connections.
- Modifying connection weights.
- Erasing/excluding existing connections.
- Modifying threshold value of neurons.
- Adding or pruning the number of hidden layers.
- Modifying activation or propagation function.
- Pruning redundant or irrelevant neurons in the hidden layer.

3.4 Pros and cons of ANNs

Some of the salient advantages of neural networks are [73]:

- Both regression and classification problems can be resolved using artificial neural networks (which are more flexible with regards to their applications).
- Any type of data which exists in numeric form can be employed in an ANN model, since ANN is a mathematical mechanism with certain estimation functions.
- ANNs are capable of extracting a hidden non-linear relationship of given data and actual outputs.

- The predictions made by ANNs are more accurate and faster after they go through the training phase.
- Neural networks are generally more flexible than other learning models. They can be used for supervised, un-supervised and reinforcement learning mechanisms.
- ANNs produce improved results with more data samples.
- An ANN model accommodates any number of inputs and layers.
- An ANN is capable of mimicking convoluted feature space over a certain degree of freedom.

Some of the disadvantages of neural networks are:

- They don't perform well in the case of small datasets.
- They require plenty of guesswork to choose an appropriate topology for any given problem.
- The training speed of an ANN is slow. It can literally take weeks or even months to train.

3.5 Fractal dimensions

The perception of a dimension has many mathematical connotations [74]. Generally, the dimension is computed to estimate the convoluted nature of an object. For instance, 'Euclidean Dimension (ED)' is the integer dimension, which embeds an object of interest on the minimum number of coordinates in a Euclidean space. In addition, 'Topological Dimension (TD)' defines the form of an object under any possible distortion, without the object losing its essential characteristics [75]. It is interesting to note that both popular dimensions are capable of measuring the complexity of an object but are limited to integer-dimension space only e.g. a smooth curve on

a 2-dimensional space has the topological dimension of 1 but its embedding dimension will be 2, Cantor Dust on a 2-dimensional space has the topological dimension of 0 but its embedding dimension will be 2. So, regardless of the principles behind calculations of both dimensions, an interesting fact to note is that complexity analysis of an object is limited to integer-dimension space only. Therefore, the whole idea of integer dimensions loses the capacity to understand the complexity for the multi-scale evaluation of a system that demands non-integer dimensions or scales.

By definition, "A fractal dimension of either a self-similar or self-affine object is the critical exponent in a power-law relation which makes a measure of the object constant" [76]. In other words, it can be interpreted as the degree of roughness, brokenness, irregularity, or singularity of an object [76]. A fractal dimension is a non-integer dimension. For example, despite the fact that the length of Britain's coastline is inestimable/unlimited (it has a certain characteristic degree of roughness), it has a non-integer fractal dimension of 1.24. It has also been observed that outside the critical value, if the exponent is too small, the measure diverges to infinity or if the exponent is too large, it vanishes to zero. There are, however, so called fat fractals whose dimensions are integer, even though their structures may be non-differentiable or even nowhere differentiable just like the regular fractals. In the domain of big data analysis, the non-integer dimension is generally computed to evaluate long range correlation among a given or extracted set of features. Hence, non-integer dimensions extract the hidden relationship of complexity among features. In other words, it is like finding connection across different amplified and convoluted scales.

In contrast to other integer dimensions, mathematically these are expressed by computing the log ratio of minimum number of volume elements with respect to available size to cover an

object at multiscale. The log ratio between volume elements and available elements is also known as an exponent which represents the degree of irregularity present in a certain object. Therefore, fractal dimension is proportional to the complexity of a system. Fractal dimensions can be classified according to the information content considered in the corresponding fractal, method of computing the dimension or may be based on the applicability of the dimension to specific processes e.g. geometry-based fractal dimension (morphological dimension), entropy-based fractal dimension (statistical information dimension), spectrum-based dimension, and polyscale variance-based dimension.

In literature, the concept of fractal dimension is fundamental to the understanding of fractals themselves [77]. It can be used in a number of applications including comparison of complicated non-differentiable static objects, dynamic non-stationary processes which may be inspected using time dependent fractal dimension may be useful in pattern recognition, identifying a limit set and, in particular, a strange attractor in chaos. A strange attractor has a non-integer dimension spectrum, while the dimension of a non-chaotic attractor is always integer. The multifractal dimensions may be useful in discovering the importance of various regions of a fractal which could be used in surface analysis of materials or images through their textures [78].

3.6 Multi-scale analysis using fractals

To compute morphological, entropy-based or variance-based fractal dimension of an object, crucial exponent of various statistical analysis quantities e.g. mutual information, entropy, self-similarity, variance is calculated at different scales. As a rule of thumb, for all ideal shapes or geometries (smooth objects) e.g. a line segment or a rectangle, the exponent is equivalent to the integer (TD) dimension. Whereas, in case of an irregular or rough object, the exponent value is

always greater than the integer (TD) dimension. Further, to compute the morphological (box-counting) dimension, a self-similar fractal object is considered. By definition, "self-similarity is the isotropic (i.e. the same scale along different axes) invariance against changes in scale or size" [80]. In other words, a geometry, figure or a time series is self-similar if a portion of the whole is a scaled down version of the whole. Moreover, we also suppose that self-similar figure is placed on an equally distanced grid and then number of boxes/VELs layering the geometry/figure are computed at first scale (S-1). Then the scale size is incremented by compressing the diameter of a VEL and again counting the number of grid squares layering the geometry at second scale (S-2). The whole procedure is repeated until the diameter of a VEL reaches to zero value (ideally speaking). The mathematical representation of the box counting (morphological) dimension is:

$$D_{box} = \lim_{r \to 0} \frac{\log N_r}{\log \frac{1}{r}} \quad (1)$$

Where N represents min. number of VELS of size r for layering an object X.

The expression to compute fractal the dimension represents the log-log relationship between figure/geometry at multiscale. The best examples of the box counting (morphological-based) fractal dimensions, which represent the complex convoluted nature of a time series with regards to their shape or geometry are Hausdorff , mass, and gyration dimensions.

Chapter 4 Dataset and feature selection

This chapter provides the detailed information of two different internet data sets with respect to their collection, structure, and contents, while highlighting their source of origin. This is followed by a detailed discussion on human analytical approach for important attribute selection. Feature space statistical analysis is then elucidated in detail.

4.1 Aegean WiFi Intrusion Dataset

AWID is a publicly available dataset which belongs to WiFi domain containing a rich blend of normal and abnormal traffic against 802.11 networks [80]. In 2016, this dataset was generated and processed at the University of Aegean, Greece. To gather this data, the overall experimentation was performed in a lab which actually resembles a SOHO infrastructure. 10 different stations (desktop machine, tablets, smart TV and laptops) were used as valid clients of the IEEE 802.11-based network. A single mobile node was introduced as an attacker to release attacks such as impersonation, flooding, and injection attacks. All valid clients were doing web browsing, VoIP and file downloading to produce wireless traffic. The overall network was made protected by a single AP (Netgear N50 WNR1000 v3 device) which supports a transfer rate up to 56 Mbps. Single attacker using Kali Linux 1.0.6 64 bit was fully equipped with a D-link DWA-125 card to inject malicious traffic. Complex and technical tools were employed to unleash various attacks e.g. MDK3 tool, Aircrack-ng suite, Metasploit framework etc. A desktop machine as a monitor node was also introduced inside the network to capture wireless traffic. This machine was using Linux Debian 7.3 operating system and a Samsung 840 series SSD hard drive. A Tshark application was also available on this node to capture pcap files.

The collection of AWID dataset is categorized into two types which differ based on the labelling method. The first type is called "AWID-CLS" and is labelled according to the main target classes of attacks i.e. impersonation, flooding, injection and normal (as demonstrated in Figure 4.1). The second type, "AWID-ATK," is named according to detailed classification based on the actual threats. i.e. Caffe-latte, Honeypot, Evil Twin etc. (see Figure 4.2 for more details) [80]

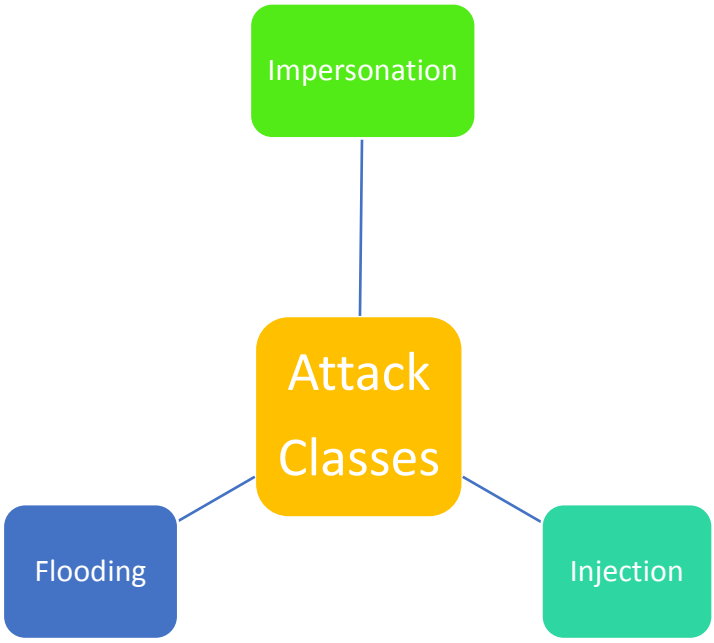

Figure 4.1 Attack classes of AWID

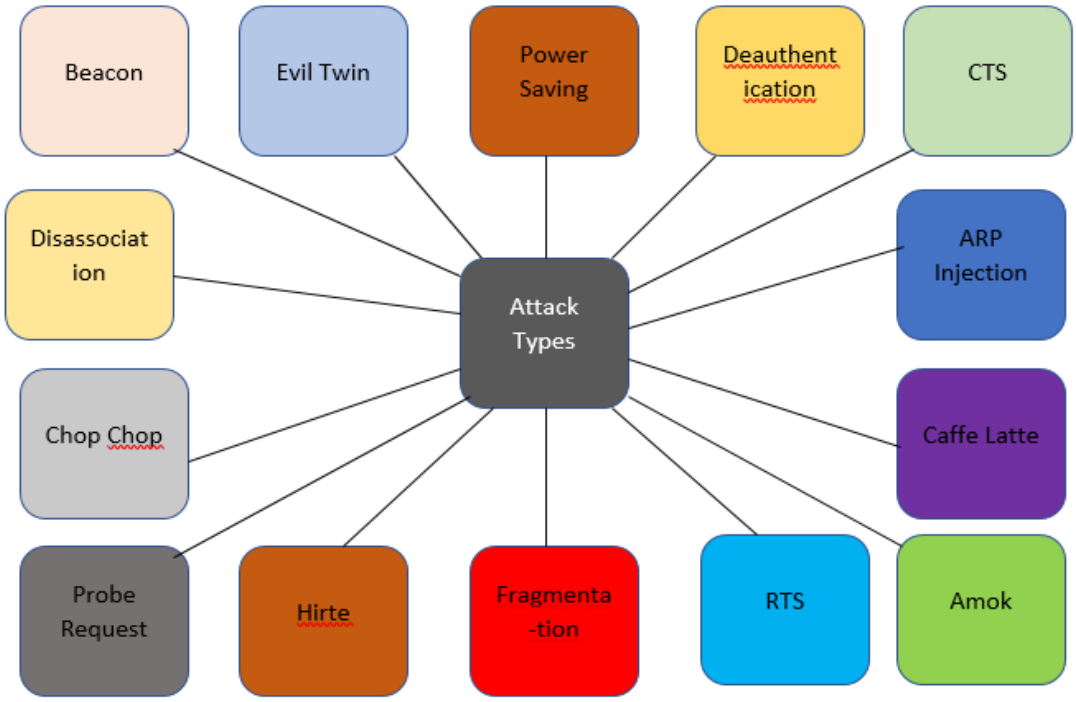

Figure 4.2 Attack types of AWID.

Each type of AWID dataset is also divided based on the size of the file e.g. “AWID-CLS-F” and “AWID-ATK-F” belong to full class and “AWID-CLS-R” and “AWID-ATK-R” belong to the reduced one. The reduced types are not the derivatives of full versions in any ways and don’t carry manufactured or fabricated data. Both subsets were generated from live network operation in two different sessions. The reduced version has better quality due to their small size and potential ability to be easily inspected by single node Tshark application. However, the large volume of full subset reflects the contemporary challenges to the intrusion detection systems to cope with data dimensionality by utilizing advanced analysis techniques.

Finally, each version has two subsets: 1) training “AWID-CLS-F-Trn, AWID-ATK-F-Trn, AWID-CLS-R-Trn, AWID-ATK-R-Trn” and 2) testing “AWID-CLS-F-Tst, AWID-ATK-F-Tst,

AWID-CLS-R-Tst, AWID-ATK-R-Tst". Both AWID-CLS-R-Trn and AWID-ATK-R-Trn carry 1,795,575 samples in total. Out of that volume 162,835 records are malicious, and 1,633,190 samples are normal. Both versions were produced within one hour, with a normal wireless traffic span of 45 minutes and a 15 minutes span of abnormal traffic. The pcap (raw data) files of both versions are of 948 MB on the disk while the Comma Separated Values (CSV) files occupy 935 MB. In this research work, we have used AWID-CLS-R-Trn and AWID-CLS-R-Tst versions to conduct the experimentation of our proposed model. The detailed breakdown of AWID family dataset is described here in Table 4.1;

Table 4.1 Breakdown of AWID Family [80].

File name	**Total Recs**	**Normal Samples**	**Anomalous Samples**	**Ratio**
AWID-CLS-F-Trn	37817835	36732463	1085372	9:1
AWID-CLS-F-Tst	4570463	4373934	196529	9:1
AWID-CLS-R-Trn	1795575	1633190	162385	3:2
AWID-CLS-R-Tst	575643	530785	44858	3:2
AWID-ATK-F-Trn	37817835	36732463	1085372	9:1
AWID-ATK-F-Tst	4570463	4373934	196529	9:1
AWID-ATK-R-Trn	1795575	1633190	162385	3:2
AWID-ATK-R-Tst	575643	530785	44858	3:2

4.2 UNSW-NB15 Dataset

The UNSW-NB15 is a publicly available real-world dataset which was processed and produced at Cyber Range Lab of the Australian Center for Cyber Security. This dataset was recorded back in 2014 and is available now in CSV and PCAP (packet capture) formats [81].

UNSW-NB15 records threats at the third and higher layers of the OSI model. It carries 49 attributes in total. Most of fields were gathered directly from raw data e.g. IPs of source and destination, information of protocol and their states etc. A small number of preprocessed features is also the part of the 49 fields e.g. number of connections that contain the same source and destination addresses in the last 100 connections, number of flows that has same method of GET and POST in http service etc. The IXIA PerfectStorm tool is used in the lab to generate a hybrid of normal and malicious network traffic. Further, IXIA tool is configured with 3 virtual servers. Both servers 1 and 3 are utilized for normal traffic while the second server spreads malicious traffic in the network. This dataset consists of several attacks as illustrated in Figure 4.3.

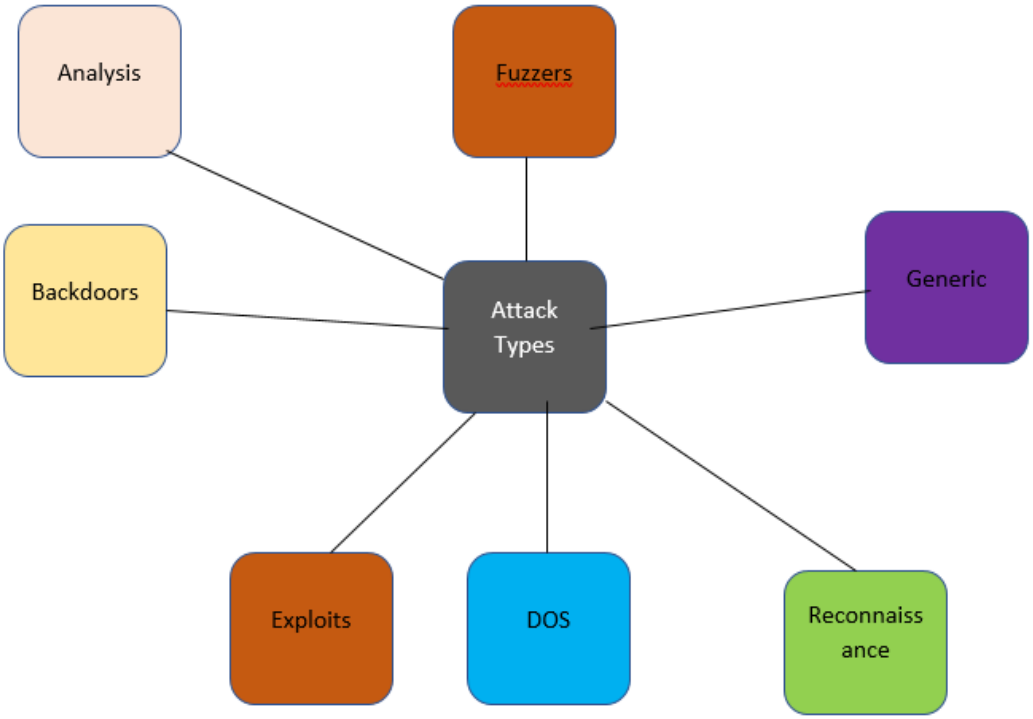

Figure 4.3 Attack types of UNSW-NB15.

The UNSW-NB15 dataset has several internet protocols but this research is primarily focused on the HTTP protocol. As per the statistics, it has been found that more than 90% of the attacks are abusing the HTTP protocol. This is interesting to note that massive exploitation of the HTTP protocol (in many forms of internet dependent communication) usually can't be stopped/filtered by administrator. Today, it has given a great chance to an attacker to utilize the protocol without getting exposed. Furthermore, sometimes the protocol is not misused but still it plays a vital role in the attack success. For instance, HTTP-based redirection is quite common these days to download any virus/malware.

The UNSW-NB15 dataset has also been divided based on training, testing, and file size. It has 6 parts in total. Two of them are labelled as training and testing however, the other four files

are labelled as 1st, 2nd, 3rd and 4th. For this research work, we utilize the UNSW-NB15_1.CSV file which contains 540,044 samples in total. It carries 31 raw data attributes and 18 extracted features. The detailed breakdown of this dataset is illustrated in Table 4.2.

Table 4.2 Breakdown of UNSW-NB15 [81].

UNSW File Number	**Total Samples**	**Normal Samples**	**Anomalous Samples**	**Ratio A/T**
1	55858	53887	1971	3.52%
2	65824	61661	4163	6.32%
3	51274	43408	7866	15.34%
4	33317	28470	4847	14.54%

4.3 Feature selection: human analytical approach

This section provides a detailed human cognitive analysis of both datasets in order to eliminate redundant, noisy and dependent attributes. It is a manual but comprehensive approach to shortlist important/relevant subsets of raw data fields, which certainly requires domain knowledge of both datasets, i.e., WiFi domain and internet protocols. How are attacks represented in a data file? How are attacks correlated with data attributes? What would it mean if data fields are dependent and independent? What is the significance of data preprocessing? To answer all these queries, one needs to have core knowledge of the domain as well as skill sets of big data analysis.

1. To shortlist meaningful fields, we first correlate every single attribute with the recorded threat with respect to wireless and wired networks environment. For example, in AWID dataset, 52 attributes named as ‘signal strength, power

management, complementary code keying, channel frequency, OFDM, data pad' etc. carry radiotap headers of the IEEE 802.11 protocol, which basically provide additional/redundant signal information of wireless network infrastructure. Such radiotap headers can't help to detect launched attacks.

2. We separate the MAC layer attributes from physical layer attributes, because attacks recorded in the AWID dataset (impersonation, injection and flooding) are not launched on the first layer of the OSI model (physical layer), and therefore, all are useless fields which show 0% correlation with attacks. For example, WiFi LAN card, PWR MGT, protected flag, current AP, status code, beacon code, association ID are first layer attributes and have no relation with the attack happening at second layer. Similarly, in UNSW-NB15, we immediately removed 18 fields because they are already preprocessed higher-order features. Since the purpose of this research is to analyze raw data fields, 31 out of 49 are selected for further analysis.
3. We remove all those fields which represent binary relation or constant relation with the launched attacks. For example, in UNSW-NB15, the timestamp field always carries 1.42E+09 value when the network traffic is normal and then the value changes when the network traffic is malicious. Also, the fields titled as 'sloss (source packets retransmitted or dropped)' and 'dloss (destination packets retransmitted or dropped)' contain either 0 or 2 when a shell code attack happens and contain different random values when the network traffic is normal. Similarly, in the AWID dataset, some of the attribute like depth, uncompressed data transfer rate, and advertisement window contains either 0 or 255 w.r.t. launched attacks. In the field of Cyber Security, such attributes are known as signatures for the attack

and this research work is purely focused on the behavioral approach for ML. Thus, all these signatures are also removed from the dataset.

4. We have also removed all those attributes which contain either ‘0’ or ‘?’. For example, in the AWID dataset, about 11% of the overall data contains no information. It is necessary to remove all these fields because they just add noise or unwantedly increase the computational complexity of the proposed methodology.
5. At this level, the AWID dataset is pruned to 80 attributes (155 in total) and the UNSW-NB15 dataset is reduced to 31 attributes (49 in total). The next task of this analysis is to measure correlation among different attributes of the dataset and to find whether any of the attributes is dependent or not. Therefore, statistical feature space analysis is implemented to shortlist a subset of independent attributes. For example, in the AWID dataset, Figure 4.4, Figure 4.5, Figure 4.6 and Figure 4.7 reveal the statistical analysis. Figure 4.4 shows the linear relationship between the MAC timestamp and Epoch time. Both fields are totally dependent on each other. Changing one field would have same impact on the other field. Therefore, we select epoch time and remove MAC timestamp. Figure 4.5 also reveals the dependency of epoch time and frame length. Frame length varies w.r.t epoch on every time stamp and also the density shows the multimodal distribution. Figure 4.6 represents independency of sequence number and epoch time. The sparsity of two-dimensional distribution shows an important independent relation of both fields. Therefore, we select both attributes as relevant fields. Similarly, in Figure 4.7, same sparsity exists between data length and epoch time. A rule of thumb for feature

space analysis is to select those attributes/fields which show sparsity or randomness.

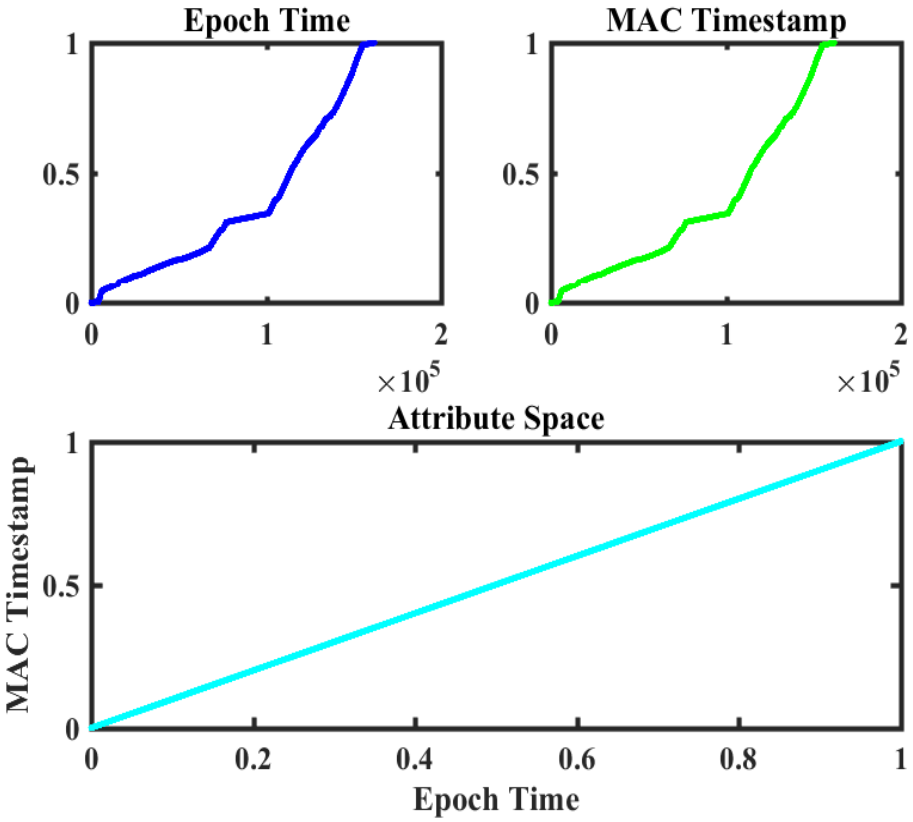

Figure 4.4 2D attribute space of AWID [70].

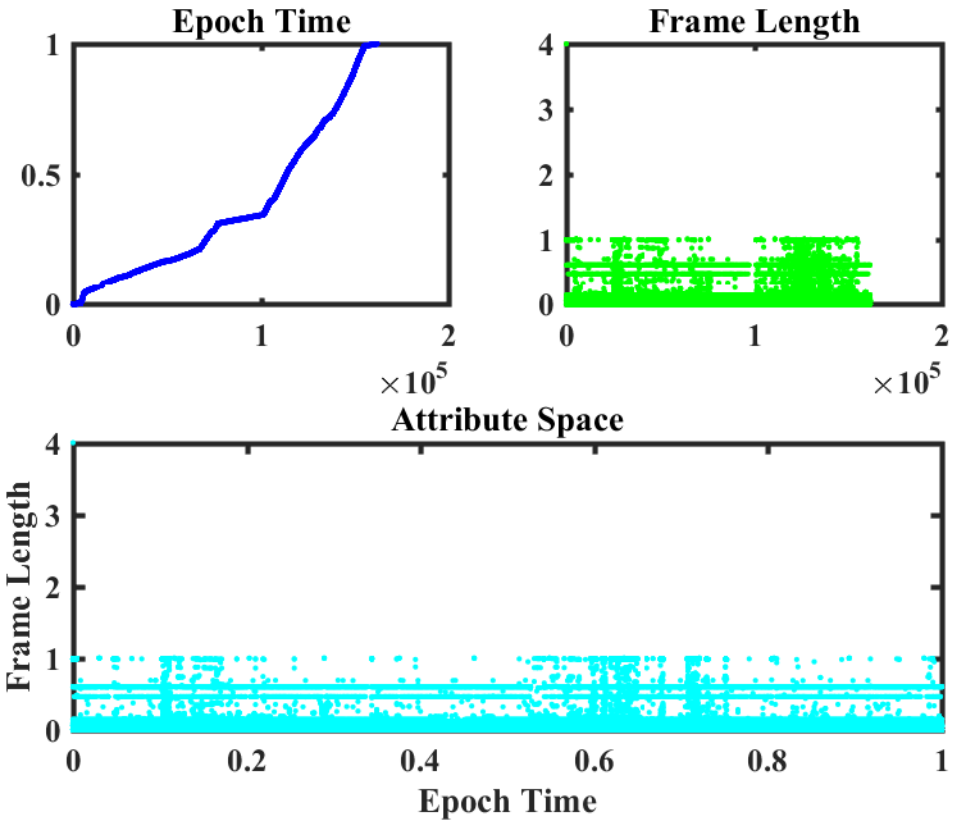

Figure 4.5 2D attribute space of AWID [70].

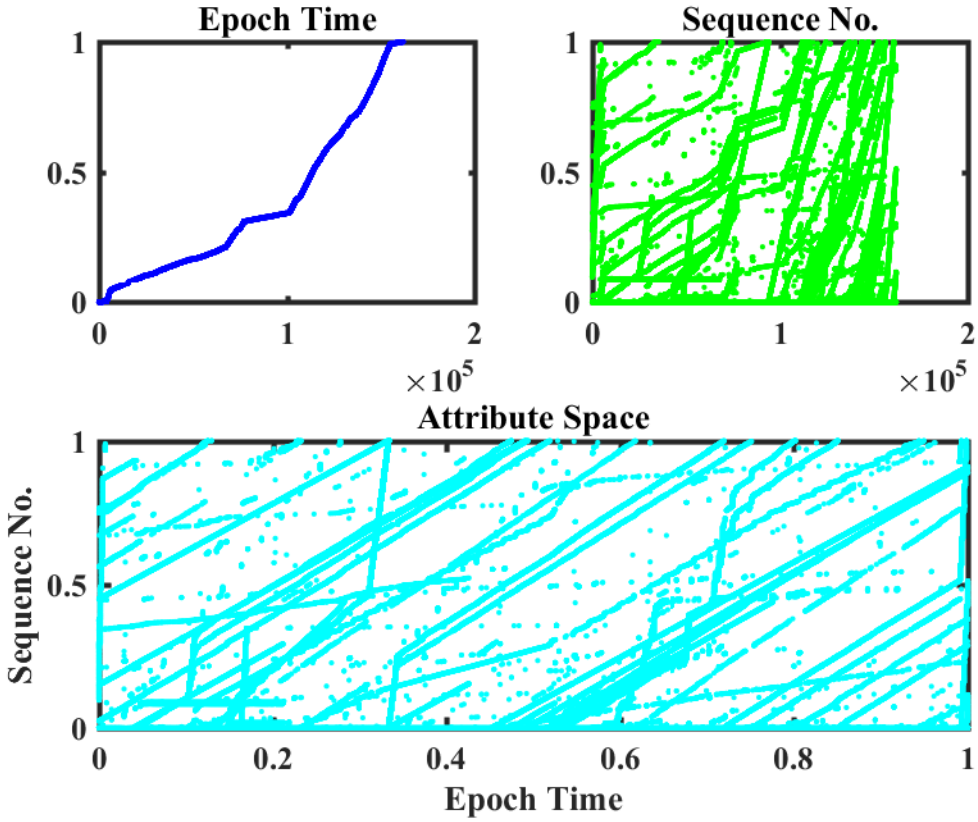

Figure 4.6 2D attribute space of AWID [70].

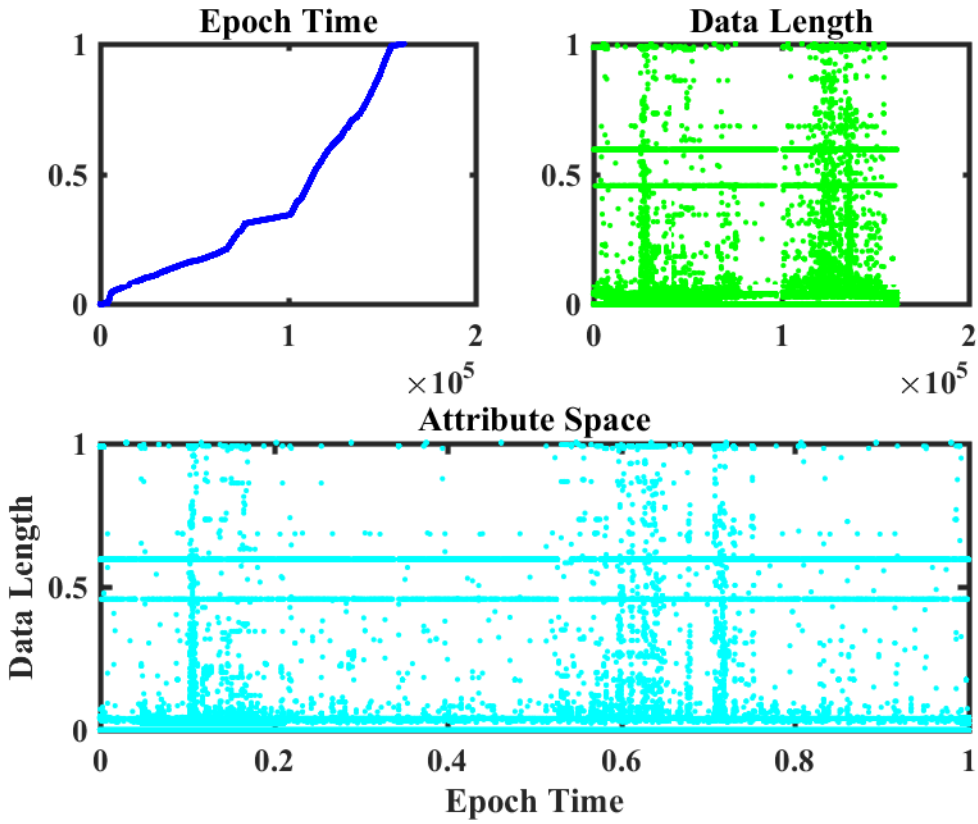

Figure 4.7 2D attribute space of AWID [70].

6. Based on the domain knowledge, in AWID, we know that there are about 33 fields which contain flag information of either 1 or 0. All these flags don't contain any information with respect to the attack. After pruning these fields, 47(out of 155) are left. In addition, the other 29 fields which carry vendor related information e.g. IBSS status, Block Ack bitmap, Multi-TID, Short Preamble, environment etc. are also redundant fields. At this stage, 18 AWID **(80-33-29= 18)** attributes are left.

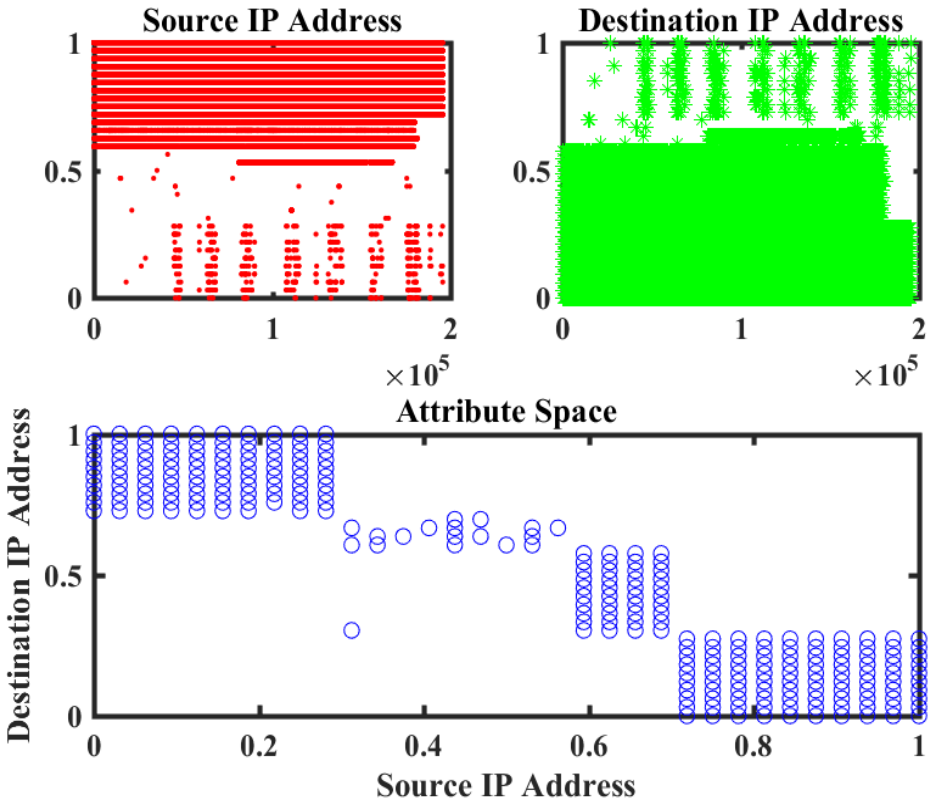

Figure 4.8 2D attribute space of UNSW-NB15 [70].

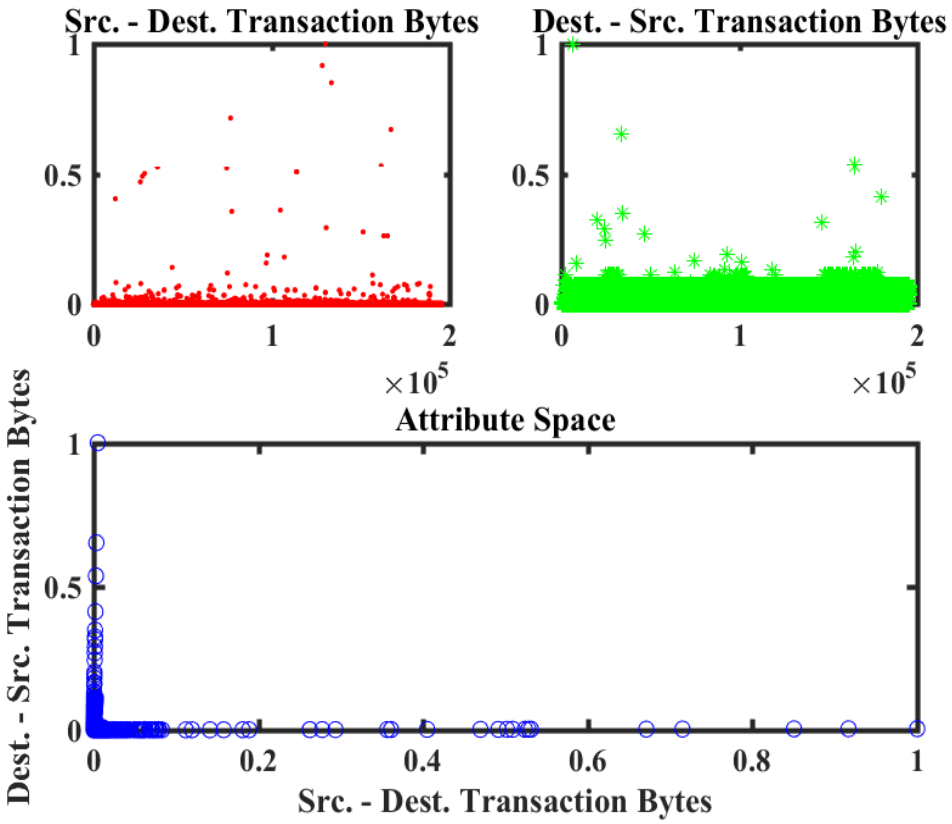

Figure 4.9 2D attribute space of UNSW-NB15 [70].

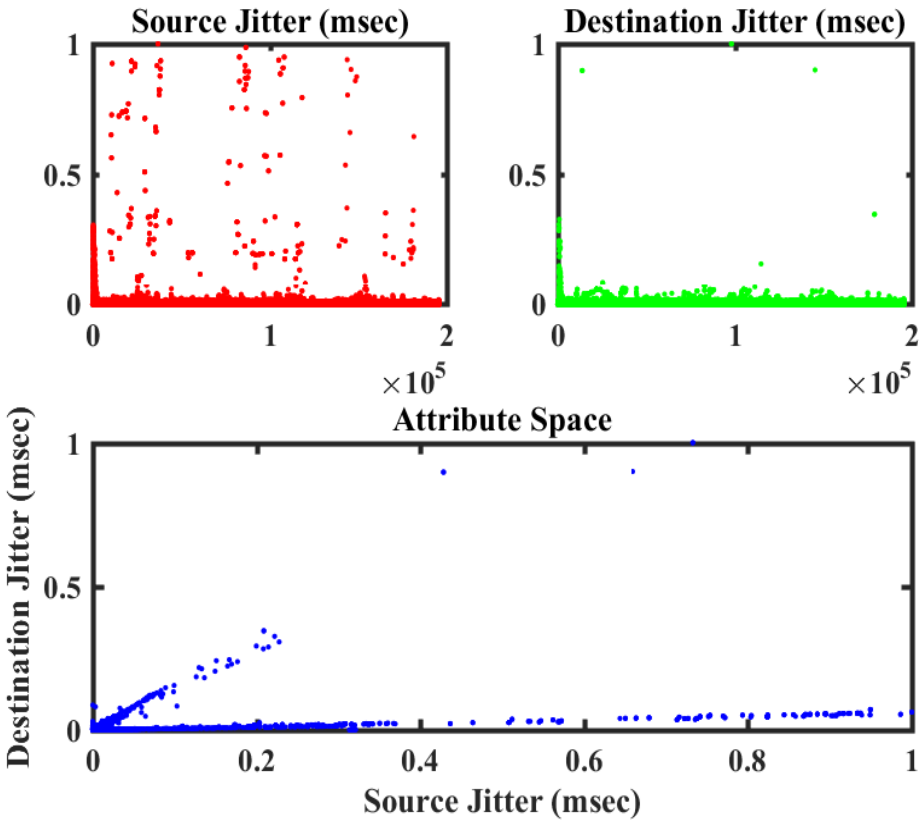

Figure 4.10 2D attribute space of UNSW-NB15 [70].

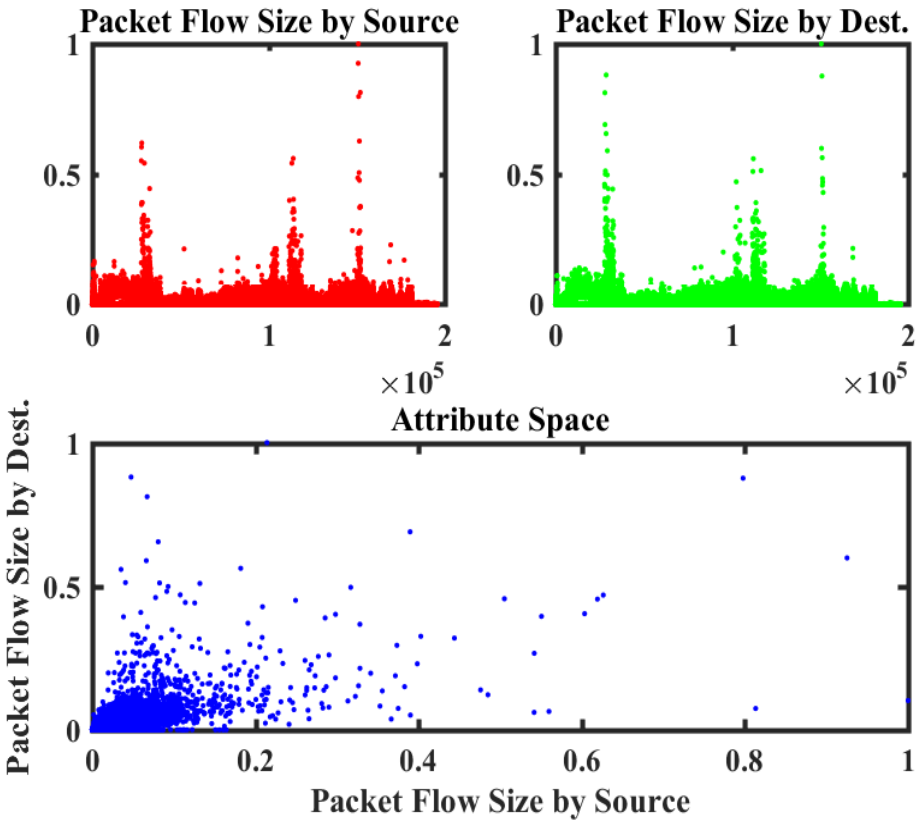

Figure 4.11 2D attribute space of UNSW-NB15 [70].

7. For AWID, we have 6-time related attributes which are linearly dependent and carry the same information so only one is needed. Likewise, we have 5 fields carrying source and destination addresses information so, only one pair is needed because 3 others show the same information. Thus, the 7 remaining attributes have either a sparse or random relationship amongst each other and are shortlisted as a final subset of most important attributes of AWID dataset.

Using all the necessary steps discussed above, we have pruned 155 raw data attributes one after the other to finally reach to 7 attributes.

Table 4.3 AWID shortlisted attributes [70].

Sr. Number	Attributes	Meanings
1	frame.time_epoch	Epoch time
2	wlan.da	Destination address
3	wlan.sa	Source address
4	frame.len	Frame length
5	data.len	Data length
6	wlan.seq	Sequence number
7	Class	Labels Normal/Attack

Table 4.4 UNSW-NB15 shortlisted attributes [70].

Sr. Number	Attributes	Meanings
1	Srcip	Src IP address
2	Sport	Src port number
3	Dstip	Dest. IP address
4	Dsport	Dest. port number
5	Proto	Transaction protocol
6	State	Indicates to the state and its dependent protocol
7	Sbytes	Src to dest. transaction bytes
8	Dbytes	Dest. to src transaction bytes
9	Sttl	Src to dest. time to live value
10	Dttl	Dest. to src time to live value
11	Sload	Src bits per second
12	Dload	Dest. bits per second
13	Sjit	Src jitter (mSec)
14	Djit	Dest. jitter (mSec)
15	Sintpkt	Src interpacket arrival time (mSec)
16	Dintpkt	Dest. interpacket arrival time (mSec)
17	Class	Labels Normal/Attack

Chapter 5 Proposed algorithm

Artificial neural network (ANN) is a nonlinear mapping model that relates a set of inputs to a set of outputs. It learns this close mapping using training data and then generalizes the acquired cognition to a new set of data. ANN Multi-Layer Perceptron (MLP) network is one of the most commonly used classifiers in the domain of Cyber Security. MLP has usually three types of layers: input, hidden, and output layer. Every layer is made up of at least m neurons (where m>1) and each neuron of a hidden layer is linked with all the neurons of the neighbouring layers. A bias or a constant number is also connected to each neuron. There are two types of connections between neurons: first layer connections and second layer connections. Each connection has its individual weight which is multiplied to the neuron value of the previous connection layer. The number of input and output neurons is determined by input and output data dimensions. However, finding the appropriate number of hidden layers and hidden neurons is an open-ended research problem which is usually decided heuristically. Cognitively speaking, the number of hidden layers and neurons required in a particular MLP network is proportional to its classification power. However, considering more than optimum number of hidden layers and neurons leads to overfitting of the classifier and significantly raises the computational complexity.

Once the weights, activation function, bias values, input and desired output vectors are determined, the MLP ANN model should be trained. Training the network means choosing suitable topology of an MLP and feeding it with sufficient training data. This enhances the generalization capability significantly. Various learning algorithms can be used to train an MLP. The best and most commonly used algorithm is back propagation [82], which is also employed in our research experimentation. In the aspect of learning, there must be some stopping criteria for the learning

algorithm. In the learning process of our proposed algorithm, we have employed the following universal set of rules as stopping criteria:

1) If the Mean Squared Error (MSE) reaches to a minimum predefined value.

2) If Gradient error reaches to a predefined value.

3) If number of iterations becomes equal to predefined value.

Note: the difference between network output (produced value) and desired output (feeding value) is interpreted as error.

The literature shows that many researchers have been working on finding an optimal size for neural networks over the last couple of decades. A significant number of pruning algorithms have been proposed so far e.g. sensitivity-based pruning (SBP), penalty-based pruning, cross validation methods, magnitude-based methods, evolutionary based pruning and correlation-based pruning to name a few. By definition pruning is described as a process of network trimming within the assumed elementary topology, which is usually achieved by evaluating the sensitivity of the total error to the omission of each attribute or neuron in the network. An attribute/field which is indifferent/unaffected to the error changes gets eliminated in the testing phase. A reduced network topology produces more promising results than before its pruning.

Moreover, an idea of a sensitivity-based feature reduction mechanism was first introduced in 1995 to anticipate the American economy system [83] . It was a novel research idea to prune the redundant features of datasets particularly related to business and economics. Recently, another research group proposed the same idea to inspect the shortcomings of a gear box [84]. The performance of this mechanism is compared with the other machine learning algorithms and it is shown that this algorithm has better performance.

The core idea of this sensitivity-based feature pruning algorithm is to train a particular MLP ANN model by feeding it a raw dataset and finding the minimum mean squared error of the dataset. After training a network, an ith field/attribute is replaced with its mean value throughout the dataset. Then the already trained network is tested by feeding the same dataset of which one attribute is now replaced with its mean value and again calculating the MSE value e.g. Figure 5.1 demonstrates all the necessary steps required to compute the sensitivity of an attribute of the dataset. Further, Figure 5.2, Figure 5.3 and Figure 5.4 elaborate the basic idea behind selecting an ith attribute, computing the mean value of that attribute, and replacing an ith attribute by its mean value.

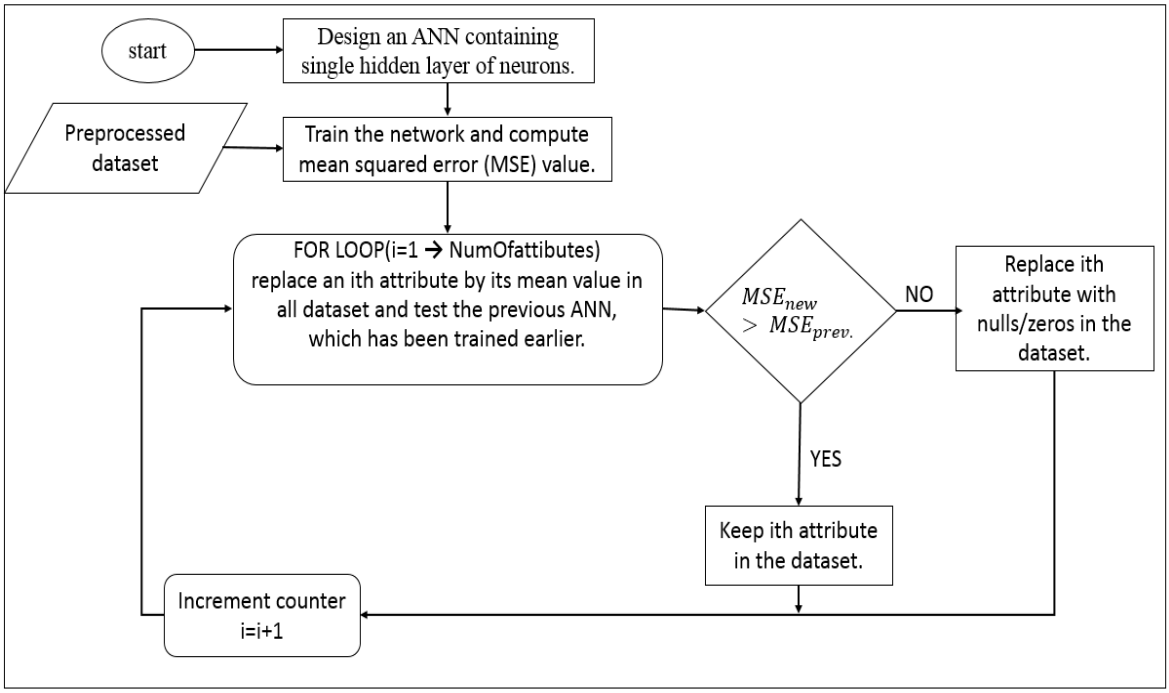

Figure 5.1 Flowchart of sensitivity-based pruning algorithm

A1	A2	A3		A154
a(1,1)	a(1,2)	a(1,3)		a(1,154)
a(2,1)	a(2,2)	a(2,3)		a(2,154)
a(3,1)	a(3,2)	a(3,3)		a(3,154)
.	.	.		.
.	.	.	. . .	.
.	.	.		.
a(161873,1)	a(161873,2)	a(161873,3)		a(161873,154)

Figure 5.2 AWID dimensions (161873 training examples, 154 attributes).

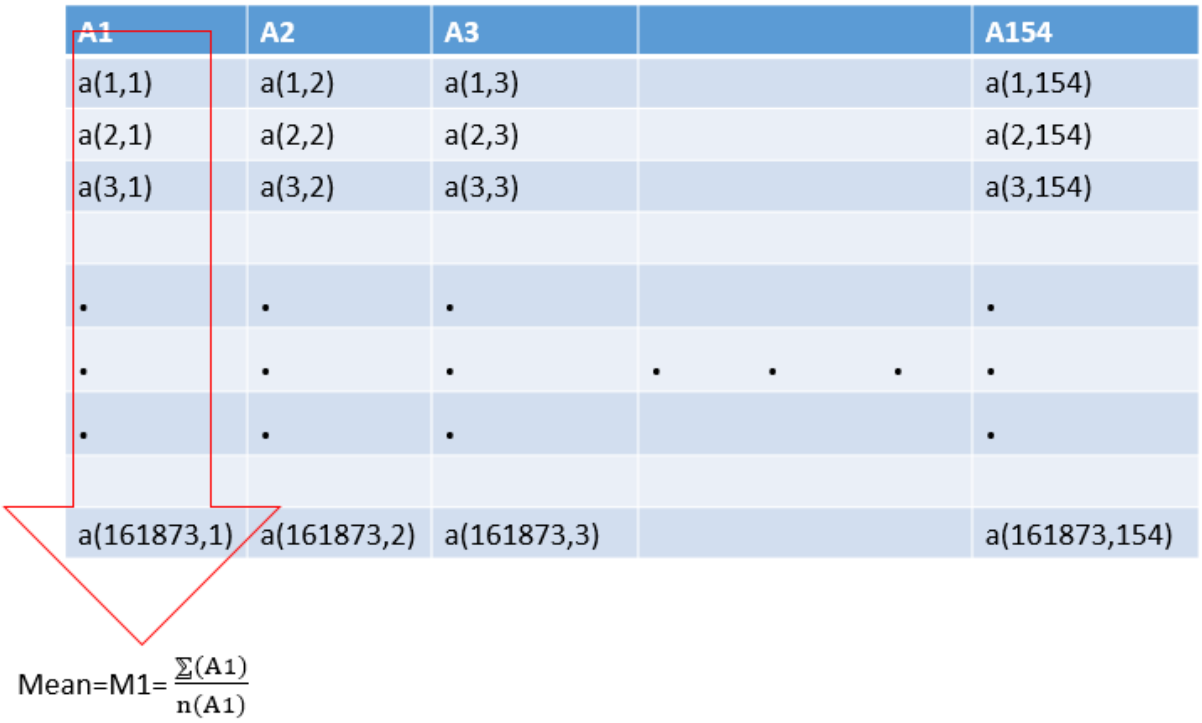

Figure 5.3 Computing mean value of 1st attribute.

A1	A2	A3		A154
M1	a(1,2)	a(1,3)		a(1,154)
M1	a(2,2)	a(2,3)		a(2,154)
M1	a(3,2)	a(3,3)		a(3,154)
.	.	.		.
.	.	.	. . .	.
.	.	.		.
M1	a(161873,2)	a(161873,3)		a(161873,154)

Figure 5.4 Replace 1st attribute by its mean value in all dataset and test the previous ANN, which has been trained earlier.

We repeat the same procedure for all the attributes of a dataset. For example, in an AWID dataset, we have 155 attributes in total. First, we replace 1st attribute with its mean value and test the network. We record the updated MSE value and compare it with the previous MSE value. The attribute remains unchanged in the dataset if the updated/new MSE value is greater than the previous MSE value of the trained network. On the other hand, we replace the attribute with nulls/zeros in the dataset if the updated MSE value is equal or less than the previous MSE value. Meanwhile, we also compute the sensitivity level of attribute by subtracting previous MSE value from the updated MSE value. We repeat the same procedure one after another until we reach the 154th attribute. The importance of an attribute is completely based on the magnitude and sign of the sensitivity value.

There are two main facts behind the significance of this sensitivity value:

1) The sensitivity value is proportional to the correlation of attributes with the launched attacks. For instance, greater error produced by the replaced attribute means that particular attribute is highly dependent on the launched attack and removing this field leads to greater error.

2) In the case of large error, the mean value itself doesn't provide any powerful information because the field is behaving randomly with large variance.

In this thesis, we have also extended the sensitivity-based pruning algorithm by introducing the idea of first order statistical moment (standard deviation) and multiscale box counting fractal dimension. Instead of replacing an attribute with its mean value, we substitute standard deviation and fractal dimension value in that particular attribute for more distinguishing and convincing results. We also validate the results by replacing uniform random numbers in each field. It seems that the proposed algorithm is harmonious with higher order statistical moments and it generates more or less same results by utilizing either mean, standard deviation or random numbers.

5.1 Sensitivity estimation algorithm with neural networks [70]

Matlab Neural Network toolbox is used to run the following algorithm.

Step 0: Initialization:

Variables	**Symbols**
Mean Squared Error	MSE
Set of attributes	$\mathcal{X}$
First layer weights	$W1$
Second layer weights	$W2$
Function input	$\mathcal{Z}$
Function output	$\mathcal{A}$
Prediction output	$\mathcal{H}$
Sigmoid function	$\mathcal{g}$
Output	$\mathcal{Y}$
Sensitivity of each attribute	$\mathcal{S}$
Mean value of each attribute	$\bar{\mathcal{X}}$

Standard deviation of each attribute	σ
Random numbers	r

Step 1: ANN training (single hidden layer):

Feedforward training function:

$$\mathcal{Z} = \sum \mathcal{W}1 * \mathcal{X}$$

$$\mathcal{A} = g(\mathcal{Z})$$

$$\mathcal{H} = g(\mathcal{A} * \mathcal{W}2)$$

$$MSE = \frac{1}{2} * \sum (\mathcal{H} - \mathcal{Y})^2$$

Step 2: ANN testing with mean values:

FOR LOOP ($i = 1 \rightarrow NumOfattributes$)

If $MSE(\bar{\mathcal{X}}) > MSE(\mathcal{X})$, compute $\mathcal{S} = MSE(\bar{\mathcal{X}}) - MSE(\mathcal{X})$ and keep the attribute unchanged in dataset.

else

compute $\mathcal{S} = MSE(\bar{\mathcal{X}}) - MSE(\mathcal{X})$ and replace the attribute with zeros in the dataset.

end

end

Step 3: ANN testing with standard deviation value:

Repeat step1 to step 2, by replacing $\bar{\mathcal{X}}$ with σ. And produce a subset of reduced attributes.

Step 4: ANN testing with random numbers:

Repeat step 1 to step 2, by replacing $\bar{\mathcal{X}}$ with $\mathcal{r}$. And produce a subset of reduced attributes.

5.2 Box-counting fractal dimension algorithm [70]

1. Assume an attribute window contains N samples or set of examples.

2. Choose the large size cover (K_{large}) such that the attribute window should provide at least 30 covers in the first scale.

$$K_{large} \geq \frac{\log 30}{\log \mathcal{b}}$$

Where, following expression calculates K_{large}.

$$K_{large} = \left[\frac{\log N}{\log \mathcal{b}}\right] - \left[\frac{\log 30}{\log \mathcal{b}}\right]$$

3. Choose small size cover (K_{small}) such that it contains at least 2 samples per cover.

4. To scale an attribute, iterate process from K_{large} till small level.

5. Calculate total number of covers needed at k-level, by computing:

$\mathcal{n}$k = $\mathcal{b}$k where, $\mathcal{b}$ is a generic number base which is kept 2 in this work.

$$N_K = \frac{N}{\mathcal{n}\mathrm{k}}$$

5.3 Hidden Neurons Pruning Mechanism

Selecting the appropriate training algorithm and choosing balanced architecture of an ANN MLP are the most critical issues with large applications. The lack of general methods to estimate an appropriate network size is considered another research challenge in the domain of machine learning. After pruning raw internet data sets, we propose another methodology called the Hidden Neurons Pruning Mechanism (HNPM) to prune the hidden layer of neural network. The proposed

scheme not only reduces computational complexity significantly but also determines an adequate size of hidden layer while maintaining the maximum classification accuracy.

Step 1: Design of an oversized neural network (154,154,1):

The most common technique to tackle this problem is known as pruning, which includes the training of an oversized network and then eliminating redundant hidden neurons while maintaining maximum classification accuracy. The overall purpose of this pruning technique is to reduce computation complexity and to achieve better generalization performance. Following the same approach, we first designed an oversized network. In Figure 5.5, the size of input layer depends on the number of raw attributes of the AWID data set (which is 154 + label class). Generally, in an oversized network the number of hidden neurons is kept equal or greater to the number of input variables. With a single output neuron, we train this network for the sufficient number of epochs. The maximum classification accuracy found for this oversized network is 97.8%.

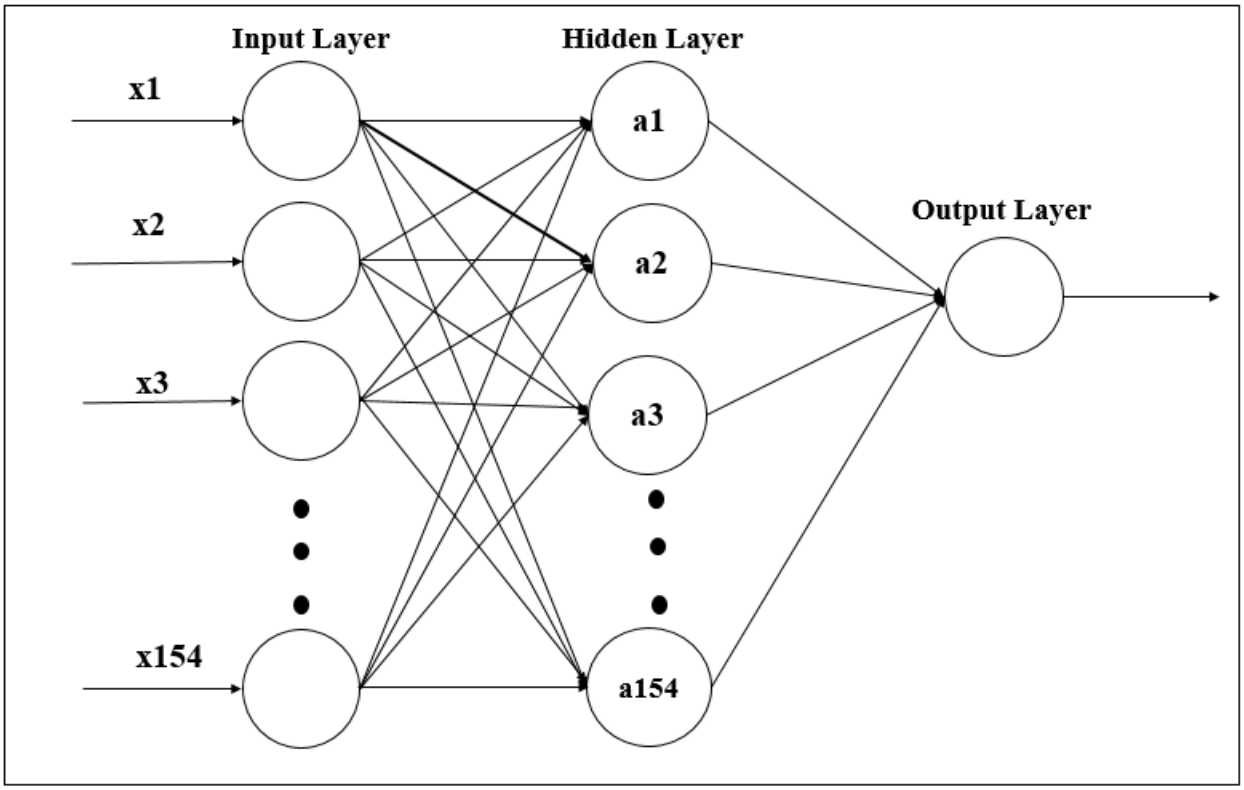

Figure 5.5 Single hidden layer oversized network (154,154,1).

Step 2: Design of a reduced neural network (154,64,1):

In Figure 5.6, we designed another architecture which contains almost 64 hidden neurons in total as compare to the oversized network. The input and output layer are the same as the previous network which is 154 and 1 respectively. After training this reduced topology for the sufficient number of epochs, the maximum accuracy rate found is 98.9 %. It is interesting to note that the detection rate has been roughly increased up to 1% by placing fewer neurons (64), which is 2/3 times less than before (154). By comparing the outcomes of both topologies as shown in Figure 5.5, Figure 5.6, it can be estimated that generalization performance largely depends on the optimal size of the hidden layer. Using a large number of hidden neurons, the network either over fit the training data or network training becomes immensely long. Sometimes overfitting causes

noise in the network and as a result the generalization performance becomes worse for new data. On the other hand, the network usually confines on the surface of local minima for an insufficient number of hidden neurons which leads to the problem of under fitting of data. Therefore, the decision of choosing an appropriate number of hidden neurons remains a great challenge.

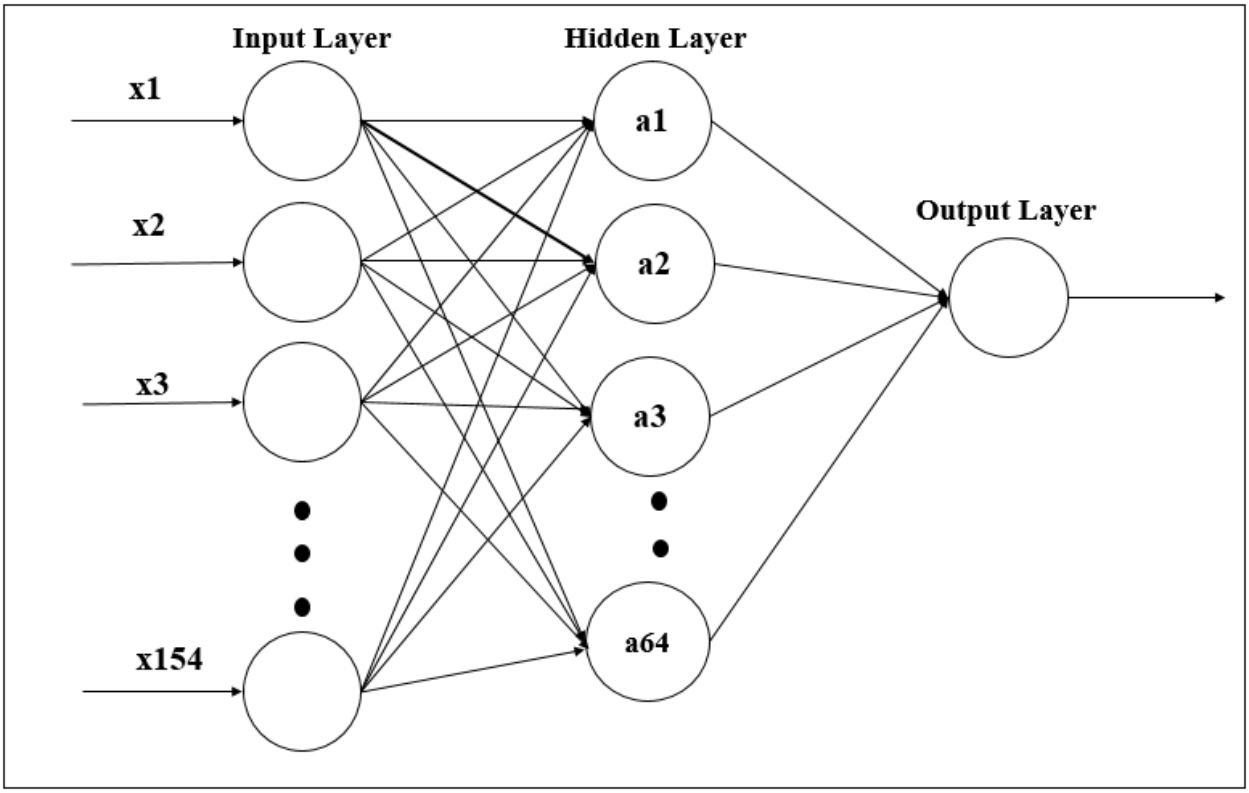

Figure 5.6 Single hidden layer reduced network (154,64,1).

Step 3: Testing reduced network (154,64,1) and recording outputs of hidden neurons:

To eliminate irrelevant hidden neurons, first, we train the architecture demonstrated in Figure 5.6 for the sufficient number of epochs. After training, we record the outputs of 64 hidden neurons for each training example, which is also called the testing phase. The AWID data set

contains 161873 examples in total. By feeding each example one by one, we record the 64 output values of hidden neurons. At the end of this testing phase, we recorded a new data set of the size 64*161873. These 64 columns are basically higher-order features.

In research world, there has always been a confusion when it comes to defining the difference between the attributes and features of a data set. The majority of researchers assume that attributes and features are the same from the perspective of their definition, information and behavior. However, an attribute itself is meaningless unless it is processed. The difference between attributes and features can be elaborated by different set of examples. E.g. a stream of 16 bits contain zeros and ones only which don't contain any information unless we apply Shannon's entropy or information theory to extract meaningful information before transmitting these bits. So, the same relation exists between attributes as unprocessed and features as processed information or data. Similarly, 3-bit binary digits have 8 possible combinations. "000, 001, 010, 011, 100, 101, 110, and 111" in general these are just possible arrangements of ones and zeros but in the world of digital logics and designs, these combinations carry specific information. So again, in the same perspective, we believe that features are always extracted from raw data attributes.

Step 4: Double layers network (64,1):

In Figure 5.7, we have designed another two-layer architecture. The first layer is the input layer and the second layer are output. It is important to note that the first layer weights in two-layer architecture (64,1) are kept the same as the second layer weights of three-layer architecture (154,64,1). Using the same set of weights, we feed higher-order features (64*161873) into the topology Figure 5.7 and test the network. It's interesting to note that again the overall classification accuracy found is the same as was found previously which 98.9% is.

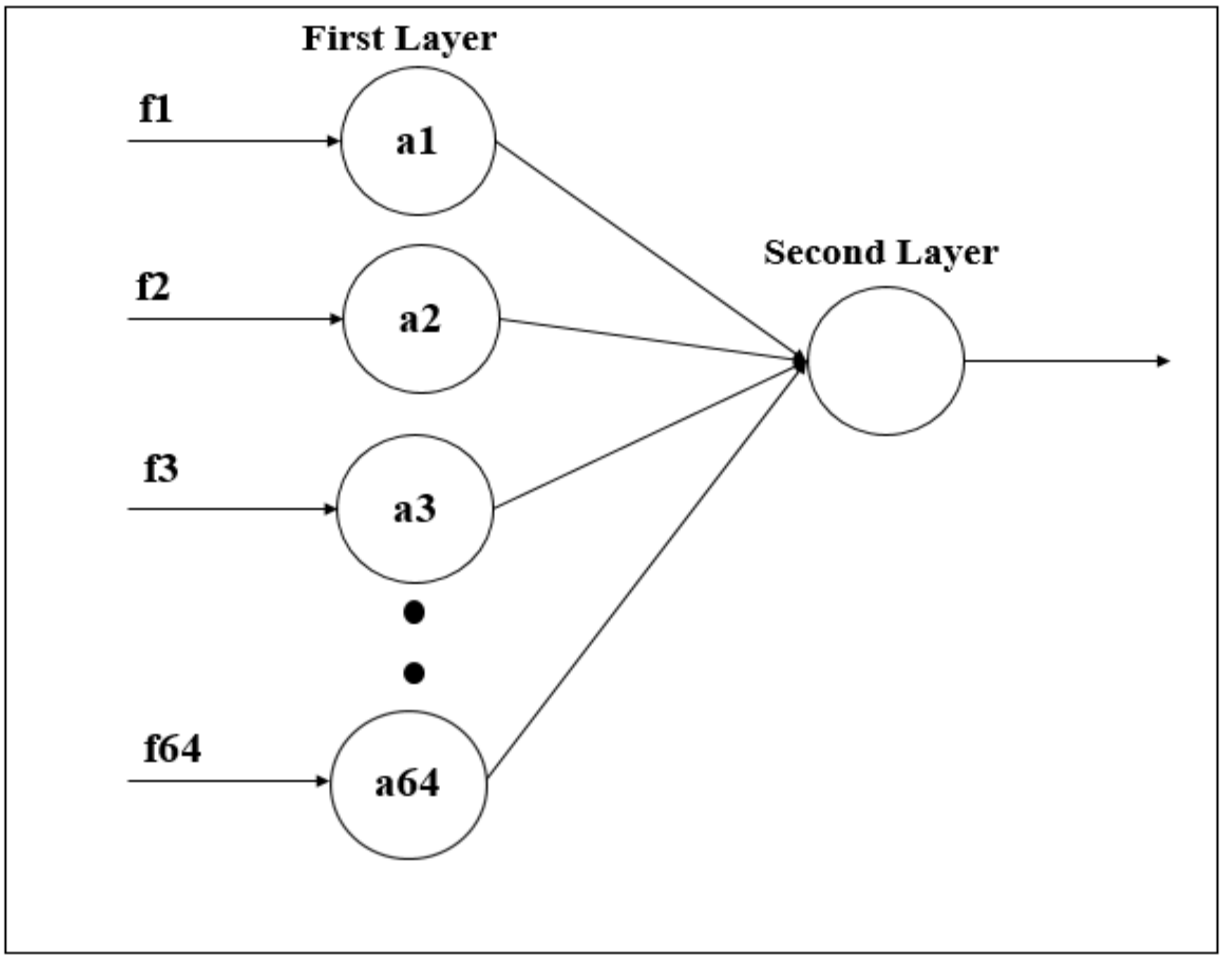

Figure 5.7 Double layers network mapping (64,1).

Up till now, we have reduced the overall network size from 154:154:1 to 64:1 while maintaining the classification accuracy of 98.9%. At this stage the question arises that can the hidden neurons be reduced further?

Step 5: Sensitivity based pruning of the newly recorded dataset (64*161873):

To further prune the hidden layer, we now apply the sensitivity algorithm (described earlier) on the newly recorded dataset with 64 higher-order set of features, (see Figure 5.8 for more details). The experimental results (Chapter 06) show that the sensitivity-based algorithm removes another

18 redundant higher-order features. Our proposed methodology (to determine a reduced size of hidden layer of an ANN) which includes HNPM and SBP mechanisms, successfully shrinks the architecture of neural networks when it comes to handling large internet datasets. It not only prunes the number of useless hidden neurons but also determines fairly reduced size of network while maintaining the possible maximum classification accuracy of 99.2%.

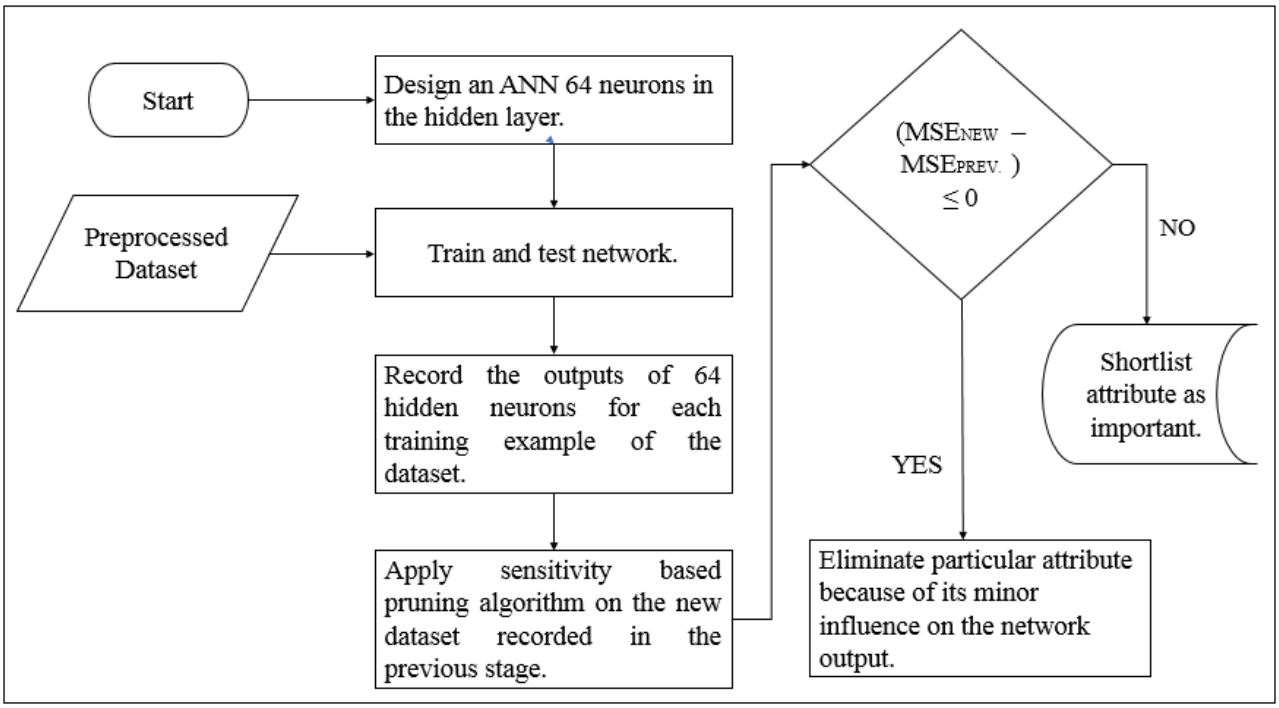

Figure 5.8 Flowchart of hidden neurons pruning mechanism.

Chapter 6 Experimental results and analysis

To assess the performance of the proposed algorithm, extensive research experiments have been conducted. The first phase of the experiments is conducted to reduce the redundant dimensions of the raw dataset. Further, the performance of the proposed algorithm is validated using two diverse internet datasets. In the second phase, we tackle another research challenge by estimating an adequate size of the neural network architecture to achieve less computational complexity and better performance in generalization. Also, the efficacy of the proposed mechanism is determined by achieving the maximum classification accuracy of more than 98%.

In addition, the experiment of the proposed methodology has been simulated on MATLAB R2016a which runs in Linux O.S, with 128 GB virtual RAM CPU, operates @3.3GHZ.

6.1 The performance of input layer pruning algorithm

The sensitivity value of each attribute is proportional to the training error of the dataset. Each attribute is ranked as per the ascending value of the error. Therefore, higher the sensitivity means higher the significance carried by the field. Figure 6.1, Figure 6.2, Figure 6.3 represents AWID shortlisted set of attributes by substituting mean, standard deviation and random numbers in the proposed algorithm. As can be interpreted from the Figure 6.1, Figure 6.2, Figure 6.3, and Figure 6.9, the shortlisted attributes (particularly numbered as 8,38,77,79,82 and 154) have more prominent sensitivity values and five of them (8,77,79,82 and 154) are exactly same attributes which have already been shortlisted through human cognitive analysis (see Table 4.3 and Table 4.4 for more details.).Further it is noted that human cognitive analysis selects Epoch time as time related attribute whereas sensitivity-based pruning algorithm shortlist MAC timestamp (which is number 38) as one of the important attributes.

Similarly, in Figure 6.4, Figure 6.5, and Figure 6.6, it can be inferred that SBP algorithm exactly shortlist the same set of attributes which have already been listed in Table 4.4. However, the experimental results show that single scale analysis using mean, standard deviation and random numbers doesn't contribute significantly in selecting important attributes. It is important to note that first order statistical moments produce similar results or provide almost the same sensitivity values for a shortlisted subset of variables. Basically, mean represents the central value of a set of examples of a particular attribute whereas the standard deviation represents how spread out the examples of a particular attribute are. In order to evaluate the distribution of data sets we repeat the whole algorithm again for standard deviation, by calculating the weight of each attribute and keeping the sensitivity threshold level same as it was previously. We found out that mean and variance have a close relationship because it is easier to predict the distribution followed by respective datasets. In both datasets, we found that the mean value of each attribute is close enough or almost equal to the standard deviation value and by definition the exponential distribution is the special case of the exponential families of distribution in which mean and standard deviation values of a set of variables are the same. So, cognitively speaking exponential distribution is another interesting way of getting the same subset of reduced attributes by placing first and second order moments. Similarly, randomness is another tool to analyze the behavior (whether following any sequence or random distribution) among the examples of an attribute Getting same sensitivity level depicts the characteristic of random variation of the examples of each attribute of the respective internet datasets. Therefore, cognitively it makes more sense to accept results by placing random numbers if both datasets already hold exponential distribution.

On the other hand, when box-counting fractal dimension algorithm is implemented instead of single scale mean, standard deviation and random numbers. It produces more distinguishing

results as shown in Figure 6.9, and Figure 6.10 for both datasets. The results produced by multiscale fractal dimension validate the selection of raw data attributes by human cognitive analysis. It therefore strengthens the concept of fractal dimension which basically provides multi-scale analysis to extract close equations among different variables, which also resembles human analytical approach in its performance.

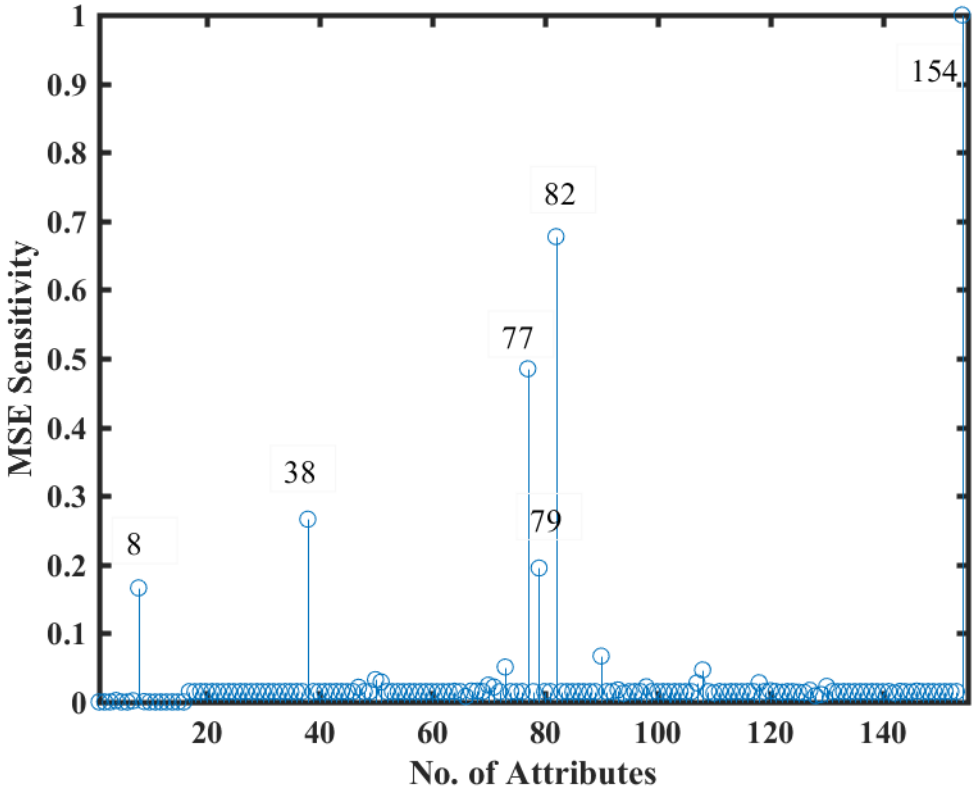

Figure 6.1 AWID: shortlisted attributes_Mean

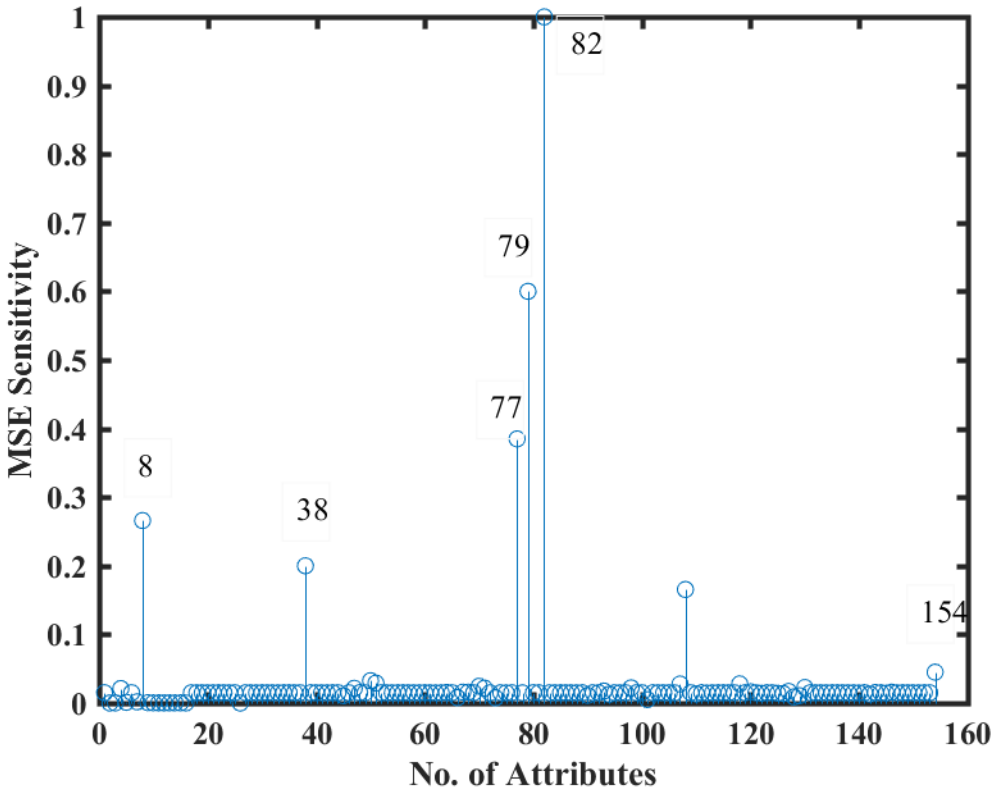

Figure 6.2 AWID: shortlisted attributes_Standard Deviation

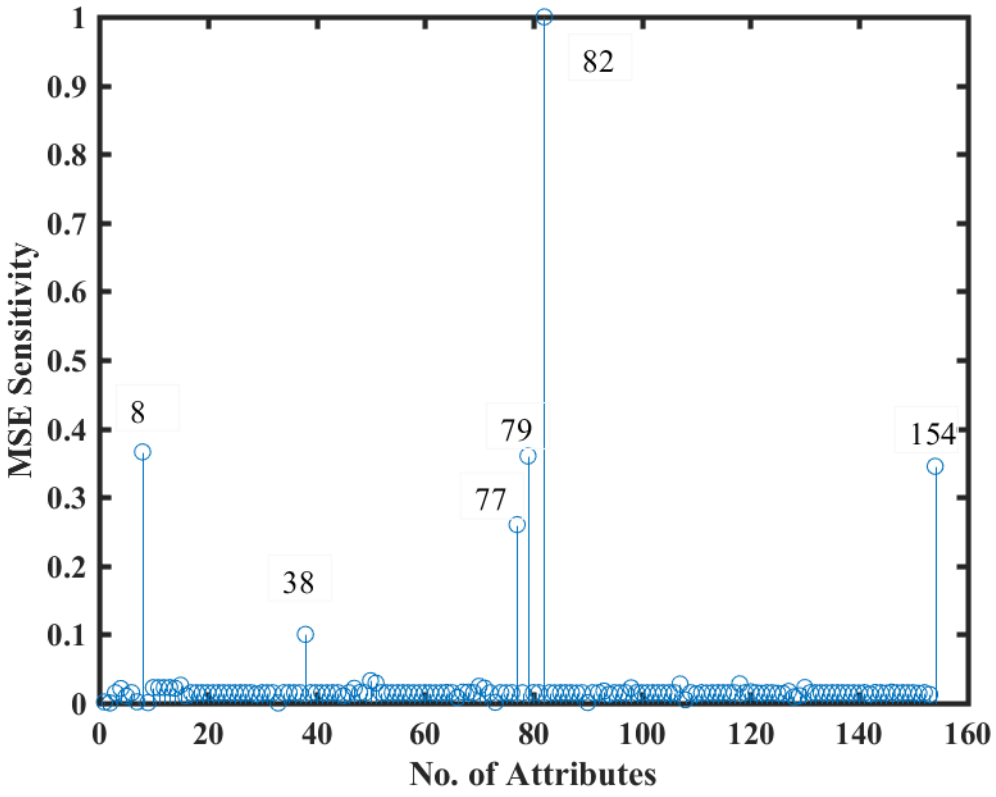

Figure 6.3 AWID: shortlisted attributes_Random Numbers

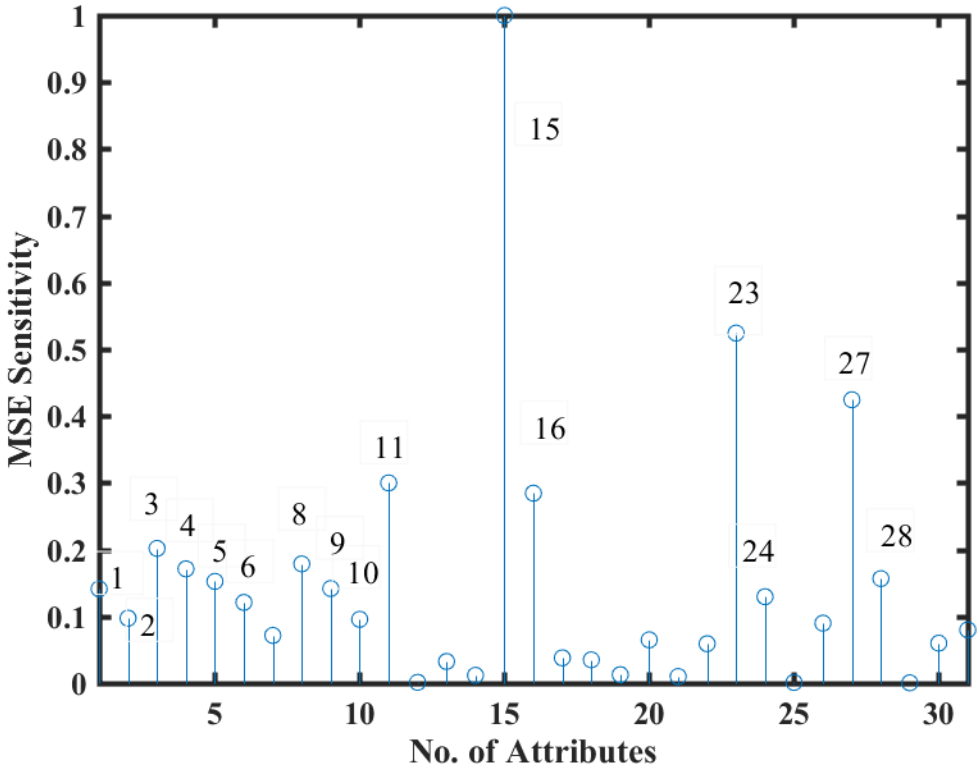

Figure 6.4 UNSW-NB15: shortlisted attributes_Mean

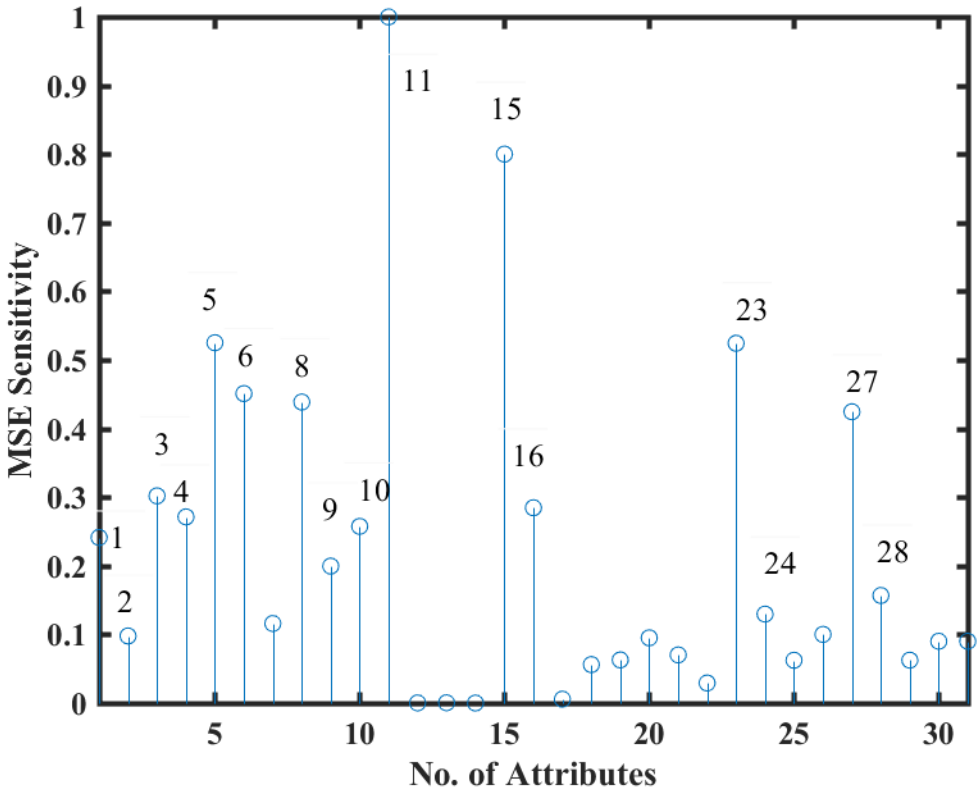

Figure 6.5 UNSW-NB15: shortlisted attributes_Standard Deviation

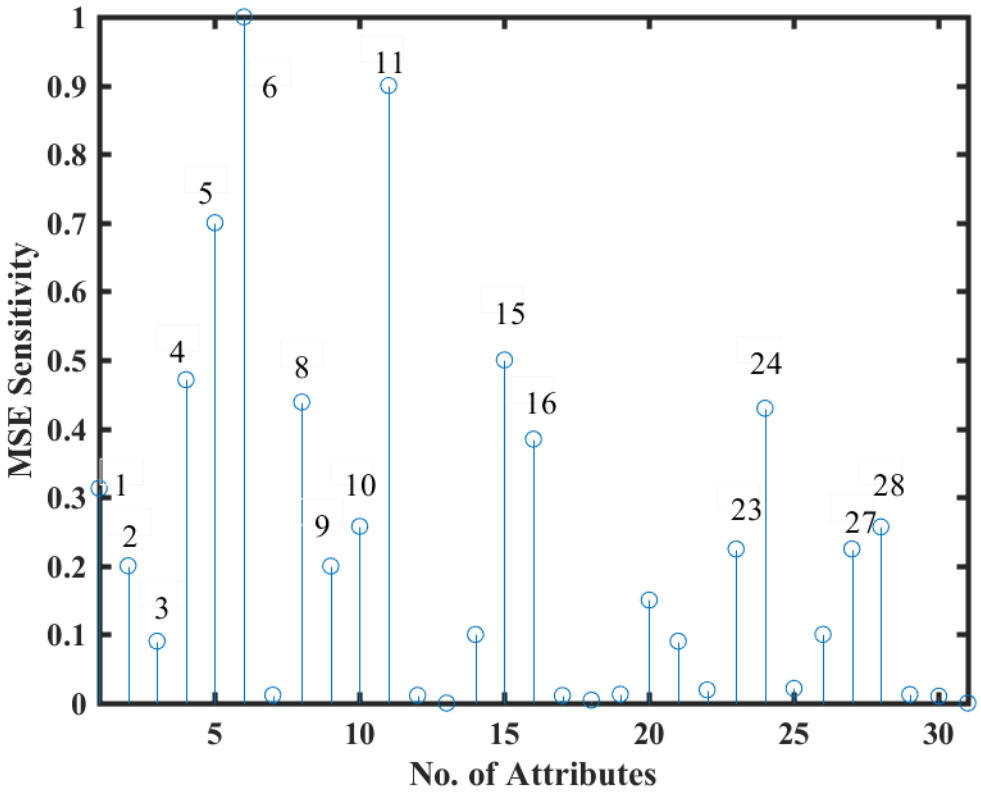

Figure 6.6 UNSW-NB15: shortlisted attributes_Random Numbers

Figure 6.7 and Figure 6.8 represents the fractal dimension plot of each attribute of both datasets. It can be seen that fractal dimension is changing w.r.t the change in each attribute and is certainly carrying all the possible information of each sample included in that attribute. By inspecting results, we have found that the AWID dataset has more variations/fluctuations whereas the UNSW-NB15 is relatively less noisy. Using deliberate human analytical analysis and the proposed algorithm, the AWID is reduced from 155 to 7 fields. It can be easily inferred that the AWID dataset is noisier and therefore the fractal dimension plot shows more fluctuations. We found a lot of garbage values in the various fields of the AWID dataset and therefore eliminated them. However, UNSW-NB15 is relatively clean dataset and a smooth curve of the fractal

dimension plot is its evident. It is important to note that noise factor of any dataset may bias ANN MLP results. A careful human cognitive analysis is required while preprocessing the dataset otherwise the factors of bias and variance compromise the generalization power of the network.

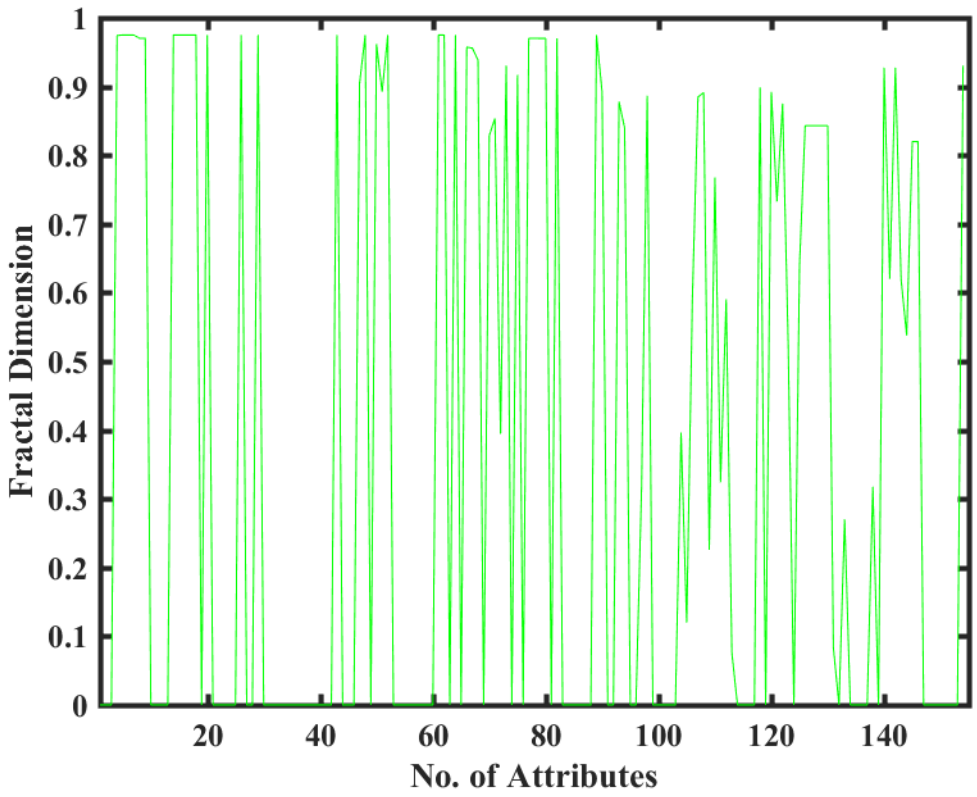

Figure 6.7 AWID: fractal dimension of attributes [70].

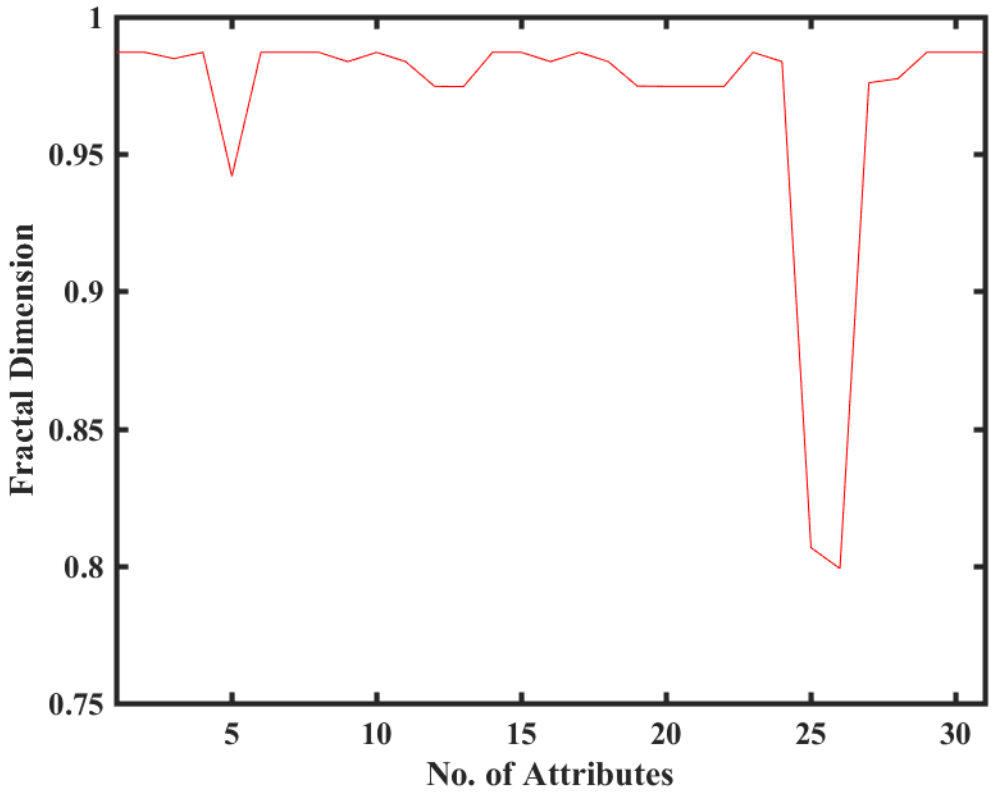

Figure 6.8 UNSW-NB15: fractal dimension of attributes [70].

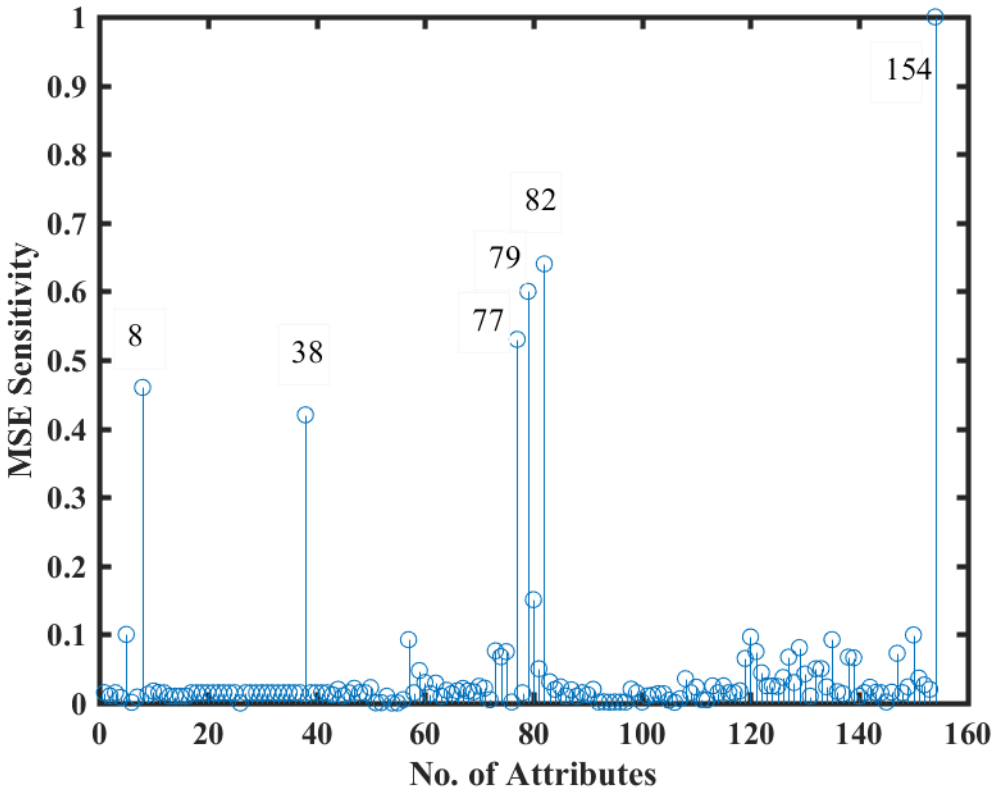

Figure 6.9 AWID: shortlisted attributes_Fractal Dimension

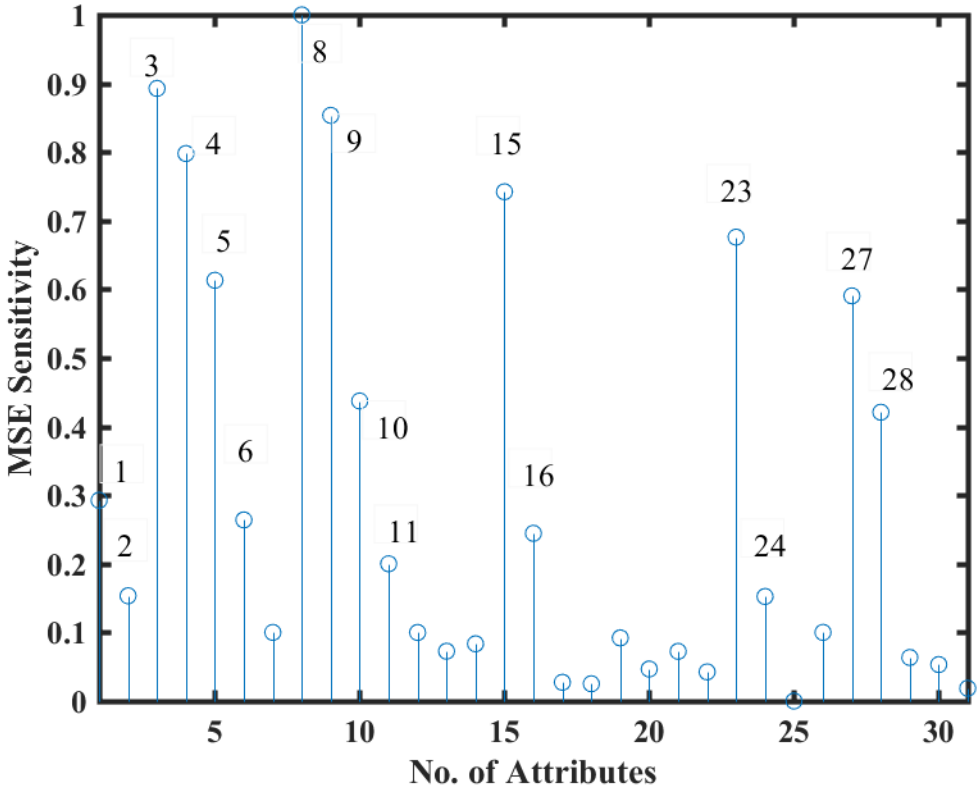

Figure 6.10 UNSW-NB15: shortlisted attributes_Fractal Dimension

As per the statistical analysis, there are two main outcomes of the proposed methodology:

1) Single scale analysis using first order moments (mean and standard deviation) produce the same results and can be treated as equal which means that the discriminating power is relatively low.

2) Multi-scale analysis discriminates attributes closer to human cognitive analysis. It can be inferred that the box-counting fractal-based mechanism is cognitive in nature, as it autonomously selects attributes based on domain knowledge, experiential learning, and complexity

analysis. The multi-scale nature of the morphological dimension extracts the close/hidden relationship among attributes and recorded attacks.

6.2 The performance of hidden layer pruning algorithm

In the second phase of experiments, we implement the sensitivity-based technique on the new recorded dataset (64*161873) to find a reduced size of the hidden layer of an ANN MLP architecture. The new dataset is the collection of hidden neuron values for each training example (already discussed in previous the chapter). All fields of this dataset are basically higher-order features. In this section, we again apply the SBP algorithm on this new dataset and try to eliminate redundant/irrelevant hidden neurons. Figure 6.11, Figure 6.12, and Figure 6.13 represent the sensitivity values of higher-order features by replacing mean, standard deviation, and the box-counting fractal dimension.

1. From the outcomes of this experiment, it can be inferred that mean, standard deviation and box-counting dimension produce the same results and can be treated as equal. While pruning the raw data attributes, we examined that first and second order moments play the same role which means their discriminatory power was relatively low. However, multiscale analysis produced more powerful or satisfactory results.
2. Figure 6.14 reveals interesting statistics of this experiment. We found that classification accuracy is inversely proportional to the number of hidden neurons. 154 hidden neurons help to classify 97.8% of data accurately whereas 46 relevant neurons increase the detection rate up to 99.2%. From these results it can be

deduced that the number of redundant/ irrelevant hidden neurons not only increases the computational complexity but also reduces the overall generalization power of the network.

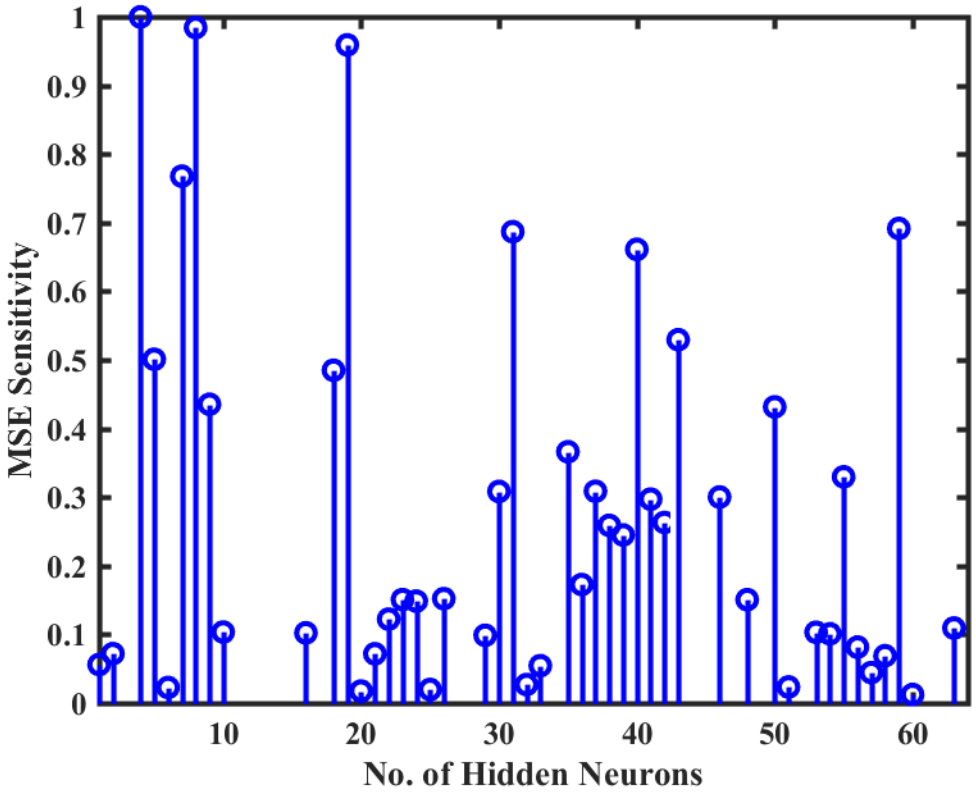

Figure 6.11 AWID: shortlisted features_Mean.

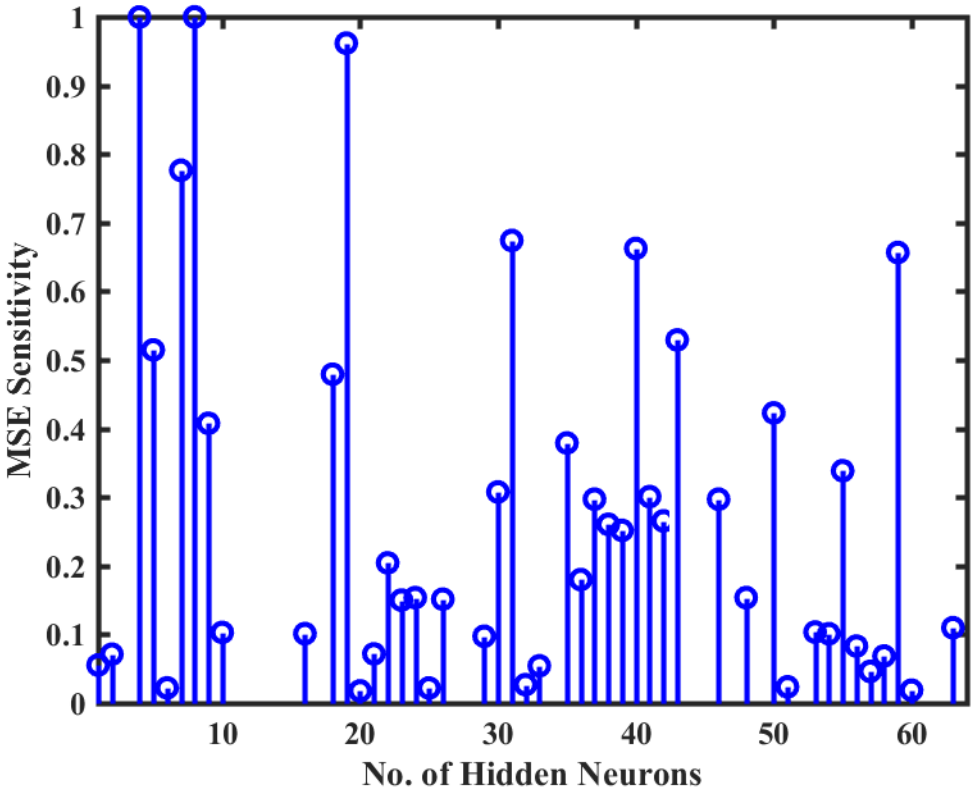

Figure 6.12 AWID: shortlisted features_Std. Deviation.

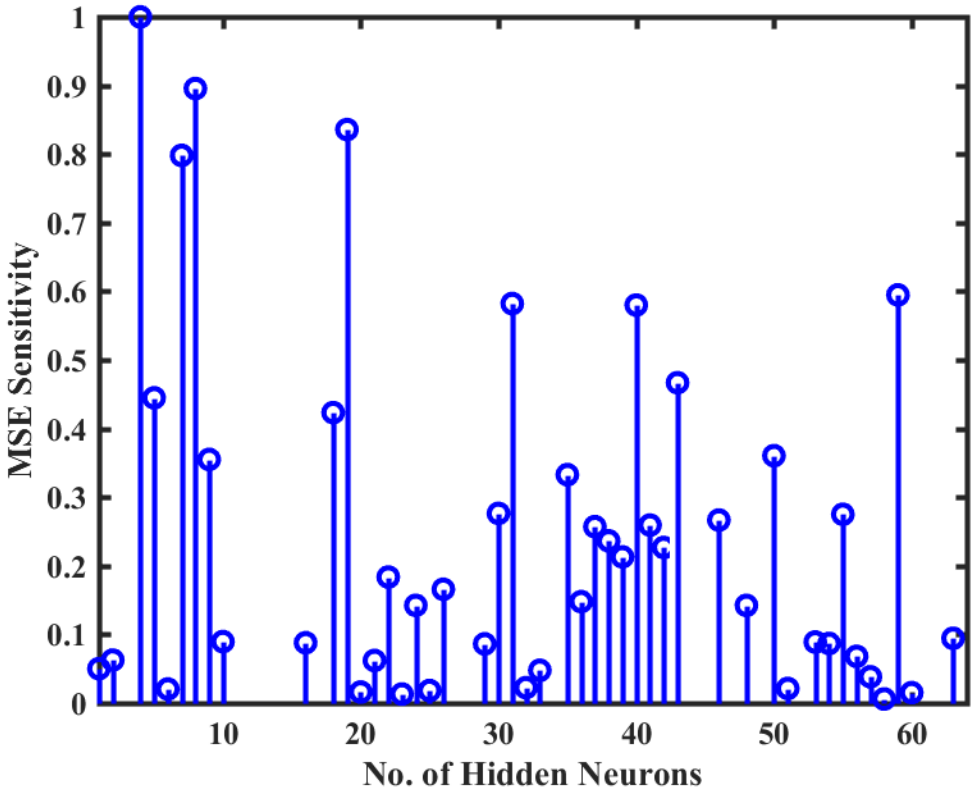

Figure 6.13 AWID: shortlisted features_Fractal Dimension.

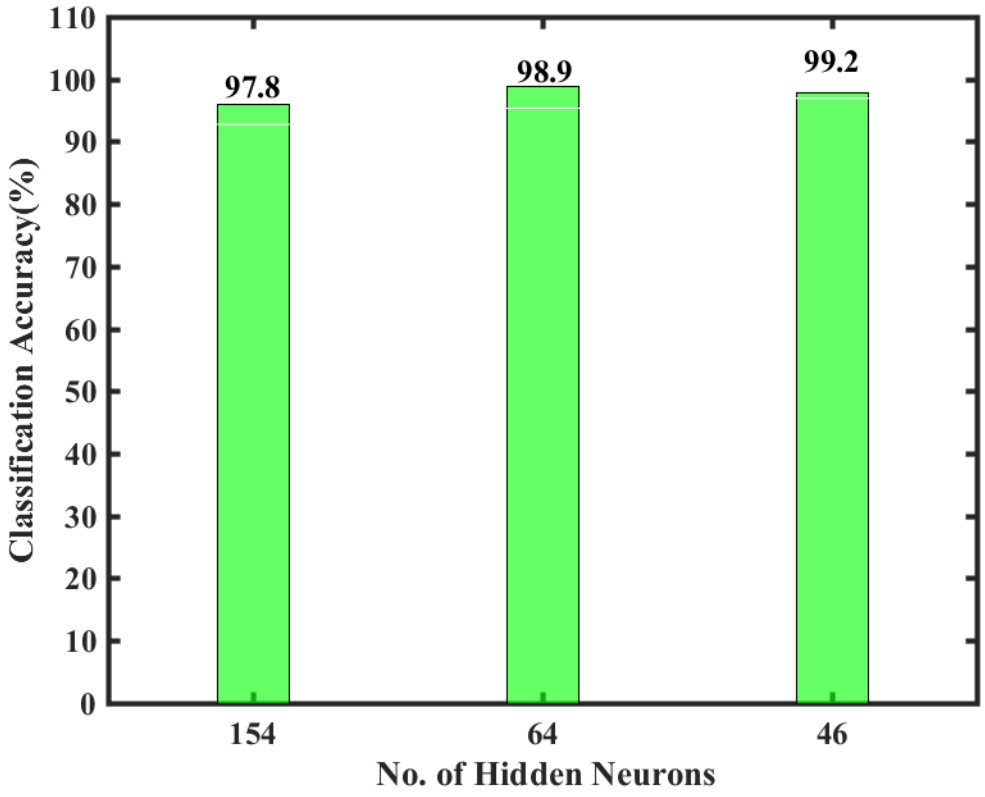

Figure 6.14 Performance evaluation of HNPM scheme.

Chapter 7 Conclusion and recommendations for future work

This thesis presents a novel application of cognitive machine learning. We propose an intelligent fractal-based methodology using neural networks for the dimensionality reduction of diverse internet datasets. The goal of this proposed mechanism is to reduce computational complexity by eliminating redundant raw data attributes and to enhance the generalization power of the ANN architecture by estimating an adequate size of the hidden layer. The experimental results confirm that the proposed model has enough potential to prune raw data attributes as well as higher-order features without requiring any special hardware. Further, the performance of the proposed model is validated using two diverse internet datasets.

Another important aspect of this proposed mechanism is to present an application for the box counting morphological fractal dimension algorithm. The experimental results confirm that cognitive selection of attributes using domain knowledge, complexity analysis and experiential learning can be achieved using fractal based multi-scale analysis. Further, first order and second order statistical moments don't play a significant role to select important fields, whereas fractal dimension provides discriminatory sensitivity values for relevant selection, which also validates the human analytical methods.

Many of the experiments, tests, and transformations have been left for the future due to time limitations (i.e. real and noisy datasets collection and analysis take enormous time and even their simulations with millions of data samples might take a couple of days for a single iteration). Future work is always needed to not only enhance the performance of the proposed model but also required to evaluate/investigate the different available dimensionality reduction techniques to achieve better solutions. The proposed recommendations for future work are as follows.

a) Our proposed mechanism is essentially a hybrid feature selection technique, which is cheaper in terms of computational complexity. In the future, we would like to explore this technique with more detailed statistical analysis methods e.g. mutual information measure among different attributes, cross-correlation measure among higher-order features, gain ratio, symmetry information, linear dependency etc.

b) In the proposed mechanism an artificial neural network is employed as the learning algorithm. Although ANNs are better to deal with big datasets but their main drawback is simulation/learning speed. In the future we would like to implement other advanced and deep learning algorithms.

c) In this research, we implemented a hybrid feature selection technique using a supervised learning algorithm. In future work, we would like to experiment some feature extraction techniques e.g. non-linear component analysis or independent component analysis using un-supervised learning algorithms.

d) As we know, the performance of the ANN depends on its own architecture. In this research work we proposed HNPM mechanism to select relevant hidden neurons. In the future work, we want to implement a complexity based higher-order features/neurons selection mechanism to determine an appropriate size of the hidden layer of neural networks.

e) This research work uses the morphological (geometry-based) fractal dimension to measure the sensitivity rank/value of each attribute. In work we would like to implement a variance-based fractal dimension as well as entropy-based fractal

dimension as a multiscale analysis tool to measure the hidden complexity of each dataset variable.

f) The proposed research highlights three cognitive components required to analyze complex internet datasets (i.e. domain knowledge, experiential learning and complexity analysis). In future work we would like to explore other salient cognitive components to further enhance the performance of the proposed mechanism.

References

[1] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald and Edin Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 1, Feb 2015.

[2] Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu and Shuo Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 67, May 2016.

[3] Amir Hussain, Erik Cambria, BjörnSchuller and NewtonHoward, "Affective neural networks and cognitive learning systems for big data analysis," ELSEVIER, Oct 2014.

[4] Witold Kinsner, "Towards Cognitive Machines: Multiscale Measures and Analysis," in *Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference*, Beijing,China, Sept. 2007.

[5] Hessam Zakerzadeh, Charu C. Aggarwal and Ken Barker, "Managing dimensionality in data privacy anonymization," *Knowledge and Information Systems*, vol. 49, no. 1, pp. 341-373, Oct. 2016.

[6] Timothy Dyster, Sameer A. Sheth and Guy M. McKhann, "Ready or Not, Here We Go: Decision-Making Strategies From Artificial Intelligence Based on Deep Neural Networks," *Neurosurgery*, vol. 78, no. 6, June 2016.

[7] Augasta and Kathirvalavakumar, "Pruning algorithms of neural networks — a comparative study," *Open Computer Science*, vol. 3, no. 3, pp. 105-115, 2013.

[8] Min Chen, Shiwen Mao and Yunhao Liu, "Big Data: A Survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171-209, April 2014.

[9] V.Bolón-Canedo, N.Sánchez-Maroño and A.Alonso-Betanzos, "Recent advances and emerging challenges of feature selection in the context of big data," *Knowledge-Based Systems*, vol. 86, pp. 33-45, Sept. 2015.

[10] Isabelle Guyon, Steve Gunn, Masoud Nikravesh and Lotfi A. Zadeh, Feature Extraction: Foundations and Applications, vol. 25, P. A. o. Sciences, Ed., Springer, 2006.

[11] SeongyounWoo and Chulhee Lee, "Incremental feature extraction based on decision boundaries," *Pattern Recognition*, vol. 77, pp. 65-74, May 2018.

[12] L.Ladha and T.Deepa, "FEATURE SELECTION METHODS AND," *International Journal on Computer Science and Engineering*, vol. 3, no. 5, May 2011.

[13] Andreas G. K. Janecek, Wilfried N. Gansterer, Michael A. Demel and Gerhard F. Ecker, "On the relationship between feature selection and classification accuracy," in *FSDM'08 Proceedings of the 2008 International Conference on New Challenges for Feature Selection in Data Mining and Knowledge Discovery*, 2008.

[14] Jianyu Miao and Lingfeng Niu, "A Survey on Feature Selection," *Procedia Computer Science,* vol. 91, pp. 919-926, 2016.

[15] Muhammad Arif Mohamad, Dewi Nasien, Haswadi Hassan and Habibollah Haron, "A Review on Feature Extraction and Feature Selection for Handwritten Character Recognition," *International Journal of Advanced Computer Science and Applications,* vol. 6, no. 2, pp. 204-212, 2015.

[16] Sufal Das and Bhabesh Nath, "Dimesionality Reduction using Association Rule Mining," in *Industrial and Information Systems, 2008. ICIIS 2008. IEEE Region 10 and the Third international Conference*, March 2009.

[17] A. Sheik Abdullah, C. Ramya, V. Priyadharsini, C. Reshma and S. Selvakumar, "A survey on evolutionary techniques for feature selection," in *Emerging Devices and Smart Systems (ICEDSS)*, Oct. 2017.

[18] Bing Xue, Mengjie Zhang, Will N. Browne and Xin Yao, "A Survey on Evolutionary Computation Approaches to Feature Selection," vol. 20, no. 4, pp. 606 - 626, Nov 2015.

[19] Mark A. Hall, "Correlation based feature selection for Machine Learning," The University of Waikato, Hamilton, NewZealand, April 1999.

[20] Daoqiang Zhang, Songcan Chen and Zhi-Hua Zhou, "Constraint Score: A new filter method for feature selection with pairwise constraints," *Pattern Recognition,* vol. 41, no. 5, pp. 1440-1451, May 2008.

[21] Shih-WeiLin, Tsung-YuanTseng, Shuo-YanChou and Shih-ChiehChen, "A simulated-annealing-based approach for simultaneous parameter optimization and feature selection of back-propagation networks," *Expert Systems with Applications,* vol. 34, no. 2, pp. 1491-1499, Feb 2008.

[22] S.-C. Chen, S.-W. Lin, T.-Y. Tseng and H.-C. Lin, "Optimization of Back-Propagation Network Using Simulated Annealing Approach," *Systems, Man and Cybernetics*, July 2007.

[23] Wang Yongxiong and Kai Li, "Feature and weight selection using Tabu search for improving the recognition rate of duct anomaly," in *2014 IEEE International Conference on Robotics and Biomimetics*, April 2015.

[24] Siti Rohaidah Ahmad, Nurhafizah Moziyana Mohd Yusop , Azuraliza Abu and Mohd Ridzwan Yaakub, "Statistical Analysis for Validating ACO-KNN Algorithm as Feature

Selection in Sentiment Analysis," in *International Conference on Applied Science and Technology*, 2017.

[25] L.S. Oliveira, N. Benahmed, R. Sabourin , F. Bortolozz and C.Y. Suen, "Feature subset selection using genetic algorithms for handwritten digit recognition," in *2002 Proceedings of XIV Brazilian Symposium on Computer Graphics and Image Processing*, Florianopolis, Brazil, Aug 2002.

[26] Chanika Sukawattanavijit, Jie Chen and Hongsheng Zhang, "GA-SVM Algorithm for Improving Land-Cover Classification Using SAR and Optical Remote Sensing Data," *IEEE Geoscience and Remote Sensing Letters,* vol. 14, no. 3, pp. 284-288, Jan 2017.

[27] Yashar Maali and Adel Al-Jumaily, "A novel partially connected cooperative parallel PSO-SVM algorithm: Study based on sleep apnea detection," in *2012 IEEE Congress on Evolutionary Computation*, Brisbane, QLD, Australia, June 2012.

[28] Xin Jin, Anbang Xu, Rongfang Bie and Ping Guo, "Machine Learning Techniques and Chi-Square Feature Selection for Cancer Classification Using SAGE Gene Expression Profiles," *Data Mining for Biomedical Applications,* vol. 3916, pp. 106-115, 2006.

[29] Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Sajid Ali and Abdur Rehman, "Comparison of feature selection methods in text classification on highly skewed datasets," in *Electrical Engineering and Computing Technologies*, Karachi, Pakistan, Feb 2018.

[30] Verónica, Noelia and Amparo, "A review of feature selection methods on synthetic data," *Knowledge and Information Systems,* vol. 34, no. 3, pp. 483-519, 2013.

[31] Mohd Shamrie Sainin and Rayner Alfred, "A genetic based wrapper feature selection approach using Nearest Neighbour Distance Matrix," in *3rd Conference on Data Mining and Optimization*, Putrajaya, Malaysia, Aug 2011.

[32] Ron Kohavi and George H. John , "Wrappers for feature subset selection," *Artificial Intelligence,* vol. 97, pp. 273-324.

[33] Sebastián Maldonado and Richard Weber, "A wrapper method for feature selection using Support Vector Machines," *Information Sciences,* vol. 179, pp. 2208-2217, 2009.

[34] Chuan Liu, Wenyong Wang, Qiang Zhao, Xiaoming Shen and Martin Konan, "A new feature selection method based on a validity index of feature subset," *Pattern Recognition Letters,* vol. 92, no. 1, pp. 1-8, June 2017.

[35] Md. Monirul Kabir, Md. Monirul Islam and Kazuyuki Murase, "A New Wrapper Feature Selection Approach using Neural Network," in *SCIS & ISIS* , Japan, 2008.

[36] Li Zhuo , Jing Zheng, Fang Wang, Xia Li , Bin Ai and Junping Qian, "A GENETIC ALGORITHM BASED WRAPPER FEATURE SELECTION," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2008.

[37] Shailesh Shrestha, Zbigniew Bochenek and Claire Smith, "EXTREME LEARNING MACHINE FOR CLASSIFICATION OF HIGH RESOLUTION REMOTE SENSING IMAGES AND ITS COMPARISON WITH TRADITIONAL ARTIFICIAL NEURAL NETWORKS (ANN)," in *EARSeL*, 2014.

[38] Haldun Aytug and Gary J. Koehler, "New stopping criterion for genetic algorithms," *European Journal of Operational Research,* vol. 126, no. 3, pp. 662-674, Nov 2000.

[39] Thammakorn Saethang, Santitham Prom-on, Asawin Meechai and Jonathan Hoyin Chan, "Sample Filtering Relief Algorithm: Robust Algorithm for Feature Selection," *Advances in Neuro-Information Processing*, vol. 5507, pp. 260-267, 2009.

[40] R.P.L.DURGABAI, "Feature Selection using ReliefF Algorithm," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 3, no. 10, Oct 2014.

[41] Patricia E.N. Lutu and Andries P. Engelbrecht, "A decision rule-based method for feature selection in predictive data mining," *Expert Systems with Applications,* vol. 37, pp. 602-609, 2010.

[42] Fatemeh Amiri, Mohammad Mahdi Rezaei Yousefi, Caro Lucas, Azadeh Shakery and Nasser Yazdani, "Mutual information-based feature selection for intrusion detection systems," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1184-1199, July 2011.

[43] Laith Mohammad Abualigah, Ahamad Tajudin Khader and Essam Said Hanandeh, "A Novel Weighting Scheme Applied to Improve the Text Document Clustering Techniques," *Innovative Computing, Optimization and Its Applications,* vol. 741, pp. 305-320, 2017.

[44] Noam Slonim, Gurinder Singh Atwal, Gašper Tkačik and William Bialek, "Information-based clustering," in *in the proceedings of PNAS*, 2005.

[45] Yanjun Li, Congnan Luo and Soon M. Chung, "Text Clustering with Feature Selection by Using Statistical Data," *IEEE Transactions on Knowledge and Data Engineering,* vol. 20, no. 5, pp. 641 - 652, 2008.

[46] Jia Hongjie, Ding Shifei, Ma Heng and Xing Wanqiu, "Spectral clustering with neighborhood attribute reduction based on information entropy," *JOURNAL OF COMPUTERS*, vol. 9, no. 6, pp. 1316-1324, 2014.

[47] Mohamed Bennasar, Yulia Hicks and Rossitza Setchi, "Feature selection using Joint Mutual Information Maximisation," *Expert Systems with Applications,* vol. 42, no. 22, pp. 8520-8532, 2015.

[48] Yvan Saeys, Iñaki Inza and Pedro Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics,* vol. 23, no. 19, p. 2507–2517, 2007.

[49] Sebastián Maldonado, Juan Pérez, Richard Weber and Martine Labbé, "Feature selection for Support Vector Machines via Mixed Integer," *Information Sciences,* vol. 279, pp. 163-175, 2014.

[50] Huanzhang Fu, Zhongzhe Xiao, Emmanuel Dellandréa, Weibei Dou and Liming Chen, "Image Categorization Using ESFS: A New Embedded Feature Selection Method Based on SFS," *Advanced Concepts for Intelligent Vision Systems,* vol. 5807, pp. 288-299, 2009.

[51] Hua Yin and Keke Gai, "An Empirical Study on Preprocessing High-Dimensional Class-Imbalanced Data for Classification," in *2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS)*, New York, NY, USA, 2015.

[52] Saúl Solorio-Fernández, J. Ariel Carrasco-Ochoa and José Fco. Martínez-Trinidad, "A new hybrid filter–wrapper feature selection method for clustering based on ranking," *Neurocomputing,* vol. 214, no. 19, pp. 866-880, 2016.

[53] Habib Motieghader, Ali Najafi, Balal Sadeghi and Ali Masoudi-Nejad, "A hybrid gene selection algorithm for microarray cancer classification using genetic algorithm and learning automata," *Informatics in Medicine Unlocked,* vol. 9, pp. 246-254, 2017.

[54] Lingyun Gao, Mingquan Ye, Xiaojie Lu and Daobin Huang, "Hybrid Method Based on Information Gain and Support Vector Machine for Gene Selection in Cancer Classification," *Genomics, Proteomics & Bioinformatics,* vol. 15, no. 6, pp. 389-395, 2017.

[55] Abdullah Saeed Ghareb, Azuraliza Abu Bakar and Abdul Razak Hamdan, "Hybrid feature selection based on enhanced genetic algorithm for text categorization," *Expert Systems with Applications,* vol. 49, no. 1, pp. 31-47, 2016.

[56] Isabelle Guyon1 and Andr´e Elisseef, "An Introduction to Feature Extraction," in *Feature Extraction Studies in Fuzziness and Soft Computing*, Springer, Berlin, Heidelberg, 2007.

[57] I. T. Jolliffe, Principal Component Analysis- Second Edition, Springer, New York, NY, 2002.

[58] Roy S. Berns and Di-Yuan Tzeng, "A review of principal component analysis and its applications to color technology," *Reasearch and Application,* 2005.

[59] Silvia Cateni, Marco Vannucci, Marco Vannocci and Valentina Colla, "Variable Selection and Feature Extraction Through Artificial Intelligence Techniques," in *Multivariate Analysis in Management, Engineering and the Sciences*, 2013.

[60] Zena M. Hira and Duncan F. Gillies, "A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data," *Advances in Bioinformatics,* 2015.

[61] Divya Gupta, Poonam Bansal and Kavita Choudhary, "The State of the Art of Feature Extraction Techniques in Speech Recognition," *Speech and Language Processing for Human-Machine Communications,* vol. 664, pp. 195-207, 2017.

[62] Aapo Hyvarinen, Juha Karhunen and Erkki Oja, "How to find the independent components?," in *Independent Component Analysis*, Toronto, JOHN WILEY AND SONS, 2004, pp. 1-476.

[63] Dominic Langlois, Sylvain Chartier and Dominique Gosselin, "An Introduction to Independent Component Analysis: InfoMax and FastICA algorithms," *Quantitative Methods for Psychology,* vol. 6, no. 1, pp. 31-38, 2010.

[64] Charles R. G. Guttmann, Robert V. Mulkern, Hakon Gudbjartsson, Carl-Fredrik Westin, Hale Pinar Zengingonul, Werner Gartner, Richard L. Robertson, Walid Kyriakos, Richard Schwartz, David Holtzman, Ferenc A. Jolesz and Stephan E. Maier, "Multi-component apparent diffusion coefficients in human brain," *NMR in Biomedicine,* vol. 12, no. 1, pp. 51-62, 1999.

[65] Y. Mori, "Nonlinear Principal Component Analysis," in *Nonlinear Principal Component Analysis and Its Applications*, JSS Research Series in Statistics, 2016.

[66] Michael E. Tipping and Christopher M. Bishop, "Probabilistic Principal Component Analysis," vol. 61, no. 3, pp. 611-622, 1999.

[67] Witold Kinsner, "Towards cognitive security systems," in *2012 IEEE 11th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, Kyoto, Japan, 2012.

[68] Klaus-Robert Muller, "Machine learning for BCI: towards analysing cognition," in *2016 4th International Winter Conference on Brain-Computer Interface (BCI)*, Yongpyong, South Korea, 2016.

[69] Vincent C. Müller and Matej Hoffmann, "What Is Morphological Computation? On How the Body Contributes to Cognition and Control," *Artificial Life,* vol. 23, no. 1, pp. 1 - 24, 02 March 2017.

[70] Danish Kaleem and Ken Ferens, "A cognitive approach for attribute selection in internet dataset," in *2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, Oxford, UK, 2017.

[71] Yingxu Wang, "On abstract intelligence and brain informatics: Mapping the cognitive functions onto the neural architectures," in *2012 IEEE 11th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, Kyoto, Japan, 2012.

[72] J. M. Taylor and V. Raskin, "Towards a formal theory of social roles in cognitive computing and cognitive informatics," in *2014 IEEE 13th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, London, UK, 2014.

[73] Tu JV, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes.," *J Clin Epidemiol,* vol. 49, no. 11, pp. 1225-1231.

[74] "WolframMathWorld: Dimension," Mathworld.wolfram.com, 2014. [Online]. Available: http://mathworld.wolfram.com/Dimension.html.

[75] Elon Lindenstraus and Benjamin Weiss, "Mean topological dimension," *Israel Journal of Mathematics,* vol. 115, no. 1, pp. 1-24, 2000.

[76] Witold Kinsner, "A Unified Approach to Fractal Dimensions," *International Journal of Cognitive Informatics and Natural Intelligence,* vol. 1, no. 4, pp. 26-46, 2007.

[77] Soumya Ranjan Nayak, Jibitesh Mishra and Pyari Mohan Jena, "Fractal Dimension of GrayScale Images," in *Progress in Computing, Analytics and Networking*, 2018.

[78] Witold Kinsner and Warren Grieder, "Amplification of signal features using variance fractal dimension trajectory," in *8th IEEE International Conference on Cognitive Informatics*, Kowloon, Hong Kong, China, 2009.

[79] Xinghua Shi, Jienan Pan, Quanlin Hou, Zhenzhi Wang, Qinghe Niu and Meng Li, "Micrometer-scale fractures in coal related to coal rank based on micro-CT scanning and fractal theory," *Fuel,* vol. 212, no. 1, pp. 162-172, 2018.

[80] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou and Stefanos Gritzalis, "Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 1, pp. 184 - 208, 2015.

[81] Nour Moustafa and Jill Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, 2015.

[82] Nazri Mohd Nawi, R. S. Ransing, Mohd Najib Mohd Salleh, Rozaida Ghazali and Norhamreeza Abdul Hamid, "An Improved Back Propagation Neural Network Algorithm on Classification Problems," in *Database Theory and Application, Bio-Science and Bio-Technology*, Berlin, Heidelberg, 2010.

[83] J. Utans, J. Moody, S. Rehfuss and H. Siegelmann, "Input variable selection for neural networks: application to predicting the U.S. business cycle," in *Computational Intelligence for Financial Engineering*, New York, USA, 1995.

[84] A.Hajnayeb, A.Ghasemloonia, S.E.Khadem and M.H.Moradi, "Application and comparison of an ANN-based feature selection method and the genetic algorithm in gearbox fault diagnosis," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10205-10209, 2011.