

**UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION
INSTITUTE OF SCIENCE AND TECHNOLOGY**

**GUIDED FEATURE SELECTION AND DIMENSIONALITY REDUCTION
METHOD FOR IDS IMPROVEMENT IN DDOS ATTACKS**



MASTER THESIS

Saif Abdulfattah AL-HELLI

DEPARTMENT OF INFORMATION TECHNOLOGY

MASTER THESIS PROGRAM

DECEMBER 2017

**UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION
INSTITUTE OF SCIENCE AND TECHNOLOGY**

**GUIDED FEATURE SELECTION AND DIMENSIONALITY REDUCTION
METHOD FOR IDS IMPROVEMENT IN DDOS ATTACKS**



MASTER THESIS

Saif Al-HELLI

1406050011

DEPARTMENT OF INFORMATION TECHNOLOGY

MASTER THESIS PROGRAM

Supervisor: Asst. Prof. Dr. Ayhan AKBAŞ

Saif AL-HELLI, having the student number 1406050011 and enrolled in the Master Program at the Institute of Science and Technology at the University of Turkish Aeronautical Association, after meeting all of the required conditions contained in the related regulations, has successfully accomplished, in front of the jury, the presentation of the thesis prepared with the title of Guided Feature Selection and Dimensionality Reduction Method for IDS Improvement in DDoS Attacks.

Supervisor : Asst. Prof. Dr. Ayhan AKBAS

University of Turkish Aeronautical Association

Jury Members : Asst. Prof. Dr. Hüseyin POLAT

Gazi University

Asst. Prof. Dr. Javad RAHEBI

University of Turkish Aeronautical Association


Asst. Prof. Dr. Ayhan AKBAS

University of Turkish Aeronautical Association

Thesis Defense Date: 12.12.2017

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.



12.12.2017

Saif AL-HELLI

ACKNOWLEDGEMENTS

I would like to thank my first thesis supervisor Asst. Prof. Dr. Ayhan AKBAŞ for his continuous guidance throughout this thesis. He made himself available for questions and help, consistently gave good insight and a practical outlook. His patient and constructive behavior of feedback were priceless in helping me achieve this thesis. Last but not least. I am forever grateful to never forget people my wife, my daughter, my parents, and siblings for their encouragements, continuing support, prayers, and love.

Thank you THKU.

12.12.2017

Saif AL-HELLI

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	x
LIST OF ACRONYMS	xi
ABSTRACT	xii
ÖZET	xiv
CHAPTER ONE	1
1. INTRODUCTION	1
1.1 Overview	1
1.2 Problem Description	1
1.3 Motivation	3
1.4 Objective	4
1.5 Main Contributions	4
1.6 Thesis Organization	5
CHAPTER TWO	6
2. RELATED WORK AND LITERATURE SURVEY	6
2.1 DDoS Detection Techniques.....	6
2.1.1 Statistical Based.....	6
2.1.2 Soft Computing.....	8
2.1.3 Knowledge-Based.....	9
2.1.4 Data Mining and Machine Learning.....	10
2.2 Literary Summaries With Methods of Selection and Reduction of the Datasets	11
2.3 Summary	15
CHAPTER THREE	16
3. BACKGROUND	16
3.1 Introduction	16
3.2 Intrusion Detection System IDS	17
3.2 Type of IDS.....	18
3.2.1 Analysis Technique	19
3.2.2 Data Sources	19
3.4 DDoS Attacks	21
3.4.1 Methods of DDoS Attack	22
3.5 Types of Attacks DoS and DDoS	23
3.5.1 Smurf Attack.....	23
3.5.2 TCP-SYN Attacks	23
3.5.3 Neptune Attack.....	24
3.5.4 Teardrop Attack.....	24
3.5.5 Back Attack	24
3.5.6 Flood Attacks.....	25

3.5.7	UDP Flood.....	25
3.5.8	ICMP Flood.....	25
3.5.9	HTTP Flood.....	25
3.6	DDoS Attack Operation.....	26
3.6.1	Flood Net.....	26
3.6.2	Trinoo (Also Called Trin00).....	26
3.6.3	The Tribe Flood Network (TFN).....	27
3.6.4	Stacheldraht (German for "Barbed Wire").....	27
3.6.5	Trinity.....	27
3.6.6	Shaft.....	27
3.7	Data Mining and Machine Learning.....	27
3.8	Learning Models.....	28
3.8.1	Supervised Learning.....	28
3.8.2	Unsupervised Learning.....	28
3.9	Concept and Application of Neural Network.....	29
3.10	Back-Propagation.....	30
3.11	ANN-BP Algorithm in IDS DDoS.....	30
3.12	Data Preprocessing.....	31
3.13	Normalization.....	31
3.14	Dimensionality Reduction.....	32
3.14.1	Principal Component Analysis.....	32
3.14.2	Singular Value Decomposition (SVD).....	34
3.15	Feature Selection.....	34
3.15.1	Wrapper Method.....	34
3.15.2	Filter Method.....	35
3.15.3	Hybrid Method.....	35
3.15.4	Classification Algorithm.....	35
3.16	Mutual Information (MI).....	35
3.17	Empirical Cumulative Distribution Functions.....	36
3.18	Summary.....	36
CHAPTER FOUR.....		37
4. DATASETS USED TO IMPLEMENT AND EVALUATE PERFORMANCE OUR MODEL.....		37
4.1	Introduction.....	37
4.2	NSL-KDD.....	37
4.3	Alkasasbeh Dataset.....	41
4.5	Summary.....	43
CHAPTER FIVE.....		44
5. DESIGN AND METHODOLOGY.....		44
5.1	Introduction.....	44
5.2	Proposed System Architecture.....	44
5.2.1	Preprocessing and Labeling Dataset Stage.....	46
5.2.1.1	The datasets.....	46
5.2.1.2	Preprocessing and labeling dataset.....	47
5.2.1.3	Normalization.....	48
5.2.2	Phase Two: Feature Selection.....	50
5.2.2.1	Mutual information (MI).....	50
5.2.2.2	Empirical cumulative distribution functions.....	52
5.2.3	Phase 3: Dimension Reduction.....	53
5.2.3.1	Enhanced SVD.....	53

5.2.3.2 Singular value decomposition (SVD).....	56
5.2.3.3 PCA Algorithm.....	58
5.2.4 Phase Four Neural Network DDoS Detector.....	61
5.2.4.1 Back-propagation.....	62
5.2.4.3 Residual network.....	64
5.3 Training and Testing Stages.....	65
5.4 Crosse –Validation.....	65
CHAPTER SIX	66
6. RESULTS And DISSCUSION	66
6.1 Introduction.....	66
6.2 Environment.....	66
6.3 Matlab	67
6.3 Performance Measurements.....	67
6.4 Confusion Matrix	68
6.5 Experiment and Results	69
6.5.1 Experimental One.....	69
6.5.2 Experimental Two	75
6.5.3 Experimental Three	79
6.5.4 Experimental Four.....	82
6.6 Results Discussion	84
CHAPTER SEVEN	87
7. CONCLUSIONS AND FUTURE WORK	87
7.1 Conclusions.....	87
7.2 Future Works	88
REFERENCES	89
CURRICULUM VITAE	96

LIST OF TABLES

Table 4.1	: NSL-KDD dataset features.....	39
Table 4.2	: Numbers and names of DoS attacks in NSL-dataset.....	41
Table 4.3	: Shows the number of records and name attack.....	41
Table 4.4	: Of Normal distribution and attack in the overall dataset.....	42
Table 4.5	: Alkawasbeh dataset features.....	42
Table 5.1	: Class labeling of NSL-KDD dataset.....	48
Table 5.2	: Class labeling of Alkawasbeh dataset.....	48
Table 6.1	: Instance of Confusion matrix.....	69
Table 6.2	: Numbers of features selected by MI with Data1.....	70
Table 6.3	: Names of reduced features at the developed SVD with Data1.....	71
Table 6.4	: Performance measurement of our model with multi-class Data1.....	71
Table 6.5	: Performance measurement of our model with binary class Data1.....	72
Table 6.6	: Numbers of features selected by MI with Data2.....	73
Table 6.7	: Names of reduced features at the developed SVD with Data2.....	74
Table 6.8	: Performance measurement of our model with multi-class Data2.....	74
Table 6.9	: Performance measurement of our model with binary class Data2.....	75
Table 6.10	: Performance measurement of 5 cases DR by PCA with multi-class Data1.....	76
Table 6.11	: Performance measurement of 5 cases DR by PCA with binary class Data1.....	76
Table 6.12	: Highest performance measurement of DR by PCA with multi-class Data 1.....	76
Table 6.13	: Performance measurement of 5 DR by PCA with multi-class Data2.....	77
Table 6.14	: Performance measurement of 5 DR by PCA with multi-class Data 2.....	79
Table 6.15	: Highest performance measurement of DR by PCA with multi-class Data1.....	78
Table 6.16	: Performance measurement of 5 DR SVD with multi-class Data1.....	79
Table 6.17	: Performance measurement of 5 DR by SVD with binary-class Data1.....	79
Table 6.18	: Highest performance measurement by SVD with multi-class Data1.....	80
Table 6.19	: Performance measurement of 5 cases DR by SVD with multi-class Data2.....	81
Table 6.20	: Performance measurement of 5 cases DR by SVD with multi-class Data2.....	81
Table 6.21	: Highest performance measurement in DR by SVD with multi-class Data2.....	81
Table 6.22	: Performance measurement with multi-class Data1 all features.....	82
Table 6.23	: Performance measurement with binary-class Data1 all features.....	83

Table 6.24 : Performance measurement with multi-class Data2 all features.....	83
Table 6.25 : Performance measurement with binary-class Data2 all features.....	83
Table 6.26 : Comparison accuracy and time the multi-class Data1 cases in multi-class Data1	84
Table 6.27 : Comparison accuracy and time FSDR with other cases in binary- class Data 1	84



LIST OF FIGURES

Figure 3.1 : Average costs of cybercrime.	17
Figure 3.2 : Common CIDF structure for IDS system.	18
Figure 3.3 : An easy DoS attack from three parallel attackers.....	21
Figure 3.4 : DDoS attack structure.....	22
Figure 3.5 : TCP SYN flood attack.	24
Figure 3.6 : General structure of backpropagation based on neural network	30
Figure 5.1 : Component architecture of the proposed work.....	46
Figure 5.2 : Mutual information with feature selection method.	52
Figure 5.3 : Structure of the enhanced SVD algorithm.....	55
Figure 5.4 : Structure of BackPropagation based on Neural Network with multi-layer.....	62
Figure 5.5 : Details of the NN in our model.....	64
Figure 6.1 : Empirical CDF for uncertainty MI value with Data1.....	70
Figure 6.2 : Training and testing error curve based on BPNN and FSDR.....	72
Figure 6.3 : Empirical CDF for uncertainty MI value with Data2.....	73
Figure 6.4 : Training and testing error curve based on BPNN and FSDR.....	75
Figure 6.5 : Accuracy rate of BPNN with ten selected features using PCA Data1.....	77
Figure 6.6 : Accuracy rate of BPNN with 10 selected features using PCA Data2.....	78
Figure 6.7 : Accuracy rate of BPNN with 20 selected features by SVD Data1.....	80
Figure 6.8 : Accuracy rate of BPNN with 10 selected features by SVD Data2.....	82
Figure 6.9 : Comparison between (FSDR) and others results with multi- class Data1.....	85
Figure 6.10 : Comparison between (FSDR) and others results with multi- class Data2.....	85
Figure 6.11 : Comparison (FSDR) and others results with multi-class Data1 consuming time in testing.....	86
Figure 6.12 : Comparison (FSDR) and others results with multi-class Data1 consuming time in testing.....	86

LIST OF ACRONYMS

TCP	: Transmission Control Protocol
UDP	: User Datagram Protocol
IMAP	: Internet Message Control Protocol
MSOS	: Microsoft Operating Systems
NN	: Neural Network
MLP	: Multilayer Perceptron
BP	: Backpropagation
SV	: Singular Vector
SVD	: Singular Value Decomposing
PC	: Principal Component
PCA	: Principal Component Analysis
ECDF	: Empirical Cumulative Distribution Function
MI	: Mutual Information
DDoS	: Distributed Denial of Service
DoS	: Denial of Service
DR	: Dimension Reduction
FS	: Feature Selection

ABSTRACT

GUIDED FEATURE SELECTION AND DIMENSIONALITY REDUCTION METHOD FOR IDS IMPROVEMENT IN DDoS ATTACKS

AL-HELLI, Saif

Master, Department of Information Technology

Thesis Supervisor: Asst. Prof. Dr. Ayhan AKBAŞ

12 Dec 2017, 96 pages

With the rapid development of the computer technologies, network security has become one of the essential issues in recent years. Distributed denial of service (DDoS) attacks are getting more and more important security threat as the improvements on network speeds create new challenges for traditional intrusion detection system (IDS) to overcome. Moreover, the IDS systems have to deal with a huge set of data with many redundant and non-useful features from systems and networks to be monitored and efficiently managed in real-time. We present an effective procedure composed of feature selection and dimensionality reduction (FSDR). In the features selection, we applied mutual information (MI) for weight the whole feature space with the use of empirical cumulative distribution function to select a subset of features that depend on the uncertainty value in MI. Also, we improved the singular value decomposing (SVD) algorithm to reduce the dimensions of the new space of the features selected in MI, and we implemented with the back-propagation neural network algorithm to detect various types of DDoS attacks. Primary experiments are implemented in MATLAB environment. Two datasets were used to test our method. First, the NSL-KDD dataset has reduced the dimensions from 41 to 4. Besides, a modern dataset of DDoS attacks created by Alkasasbeh [1] and its features are reduced from 27 to 6 with highest

classification accuracy is still obtained, after carrying out with five-fold cross-validation. Our suggested method (FSDR) can efficiently minimize dimensions and diminish their computational overhead without jeopardizing on classification accuracy of attacks.

Keywords: Distributed Denial of Service, Intrusion Detection System, Mutual Information, Empirical Cumulative Distribution Function, Singular Value Decomposing, Dimensional Reduction, Feature Selection, MATLAB.



ÖZET

DDOS SALDIRILARINDA IDS İYİLEŞTİRME İÇİN YÖNLENDİRİLMİŞ ÖZELLİK SEÇİMİ VE BOYUTLULUK İNDİRGEMESİ YÖNTEMİ

AL-HELLI, Saif

Yüksek Lisans, Bilgi Teknolojileri Bölümü

Tez Danışmanı: Yrd. Doç. Dr. Ayhan AKBAŞ

12 Aralık 2017, 96 sayfa

Bilgisayar teknolojilerinin hızla gelişmesi ile birlikte, ağ güvenliği son yıllarda önemli meselelerden biri haline gelmiştir. Ağ hızlarındaki gelişmelerin geleneksel saldırı tespit sistemi (IDS) için üstesinden gelmesi gereken yeni zorluklar yaratması nedeniyle, dağınık hizmet engelleme (DDoS) saldırıları giderek daha önemli bir güvenlik tehdidi oluşturmaktadır. Dahası, IDS sistemleri, sistemlerden ve ağlardan gelen gerçek zamanlı olarak izlenmesi ve etkili bir şekilde yönetilmesi gereken çok sayıda veri ile uğraşmak zorunda kalmaktadır; bu da, birçok gereksiz ve kullanışsız özellik içeren büyük ölçekli veriler yüzünden önemli bir sorun teşkil etmektedir. Bu tezde, özellik seçimi ve boyutluluk indirgemeden (FSDR) oluşan etkili bir yöntem sunulmaktadır. Seçilen özellikler alt kümesinde, ortak bilgideki (MI) belirsizlik değerine bağlı olan özelliklerin bir alt kümesini seçmek için ampirik kümülatif dağılım fonksiyonunun kullanımı ile tüm özellik alanının ağırlığı için ortak bilgi (MI) uygulanmıştır. Bundan sonra, MI'da seçilen özelliklerin yeni alan boyutlarını azaltmak için tekil değer ayrıştırma (SVD) algoritması geliştirilmiş ve çeşitli DDoS saldırılarını saptamak için geri yayılım sinir ağı algoritması uygulanmıştır. Birincil deneyler MATLAB ortamında uygulanmıştır. Deneysel sonuçlar bize FSDR yönteminin iki veri kümesinin optimal sayıda özelliğini seçebildiğini göstermiştir. İlk olarak, NSL-KDD veri setinin değerlendirilmesi boyutları 41'den 4'e düşürmüştü ve Alkasasbeh [1] tarafından oluşturulan DDoS saldırılarının modern bir veri setine ek olarak ve

özellikleri beş katlı çapraz doğrulama gerçekleştirdikten sonra en yüksek sınıflandırma doğruluğu ile 27'den 6'ya indirilmiştir. Önerdiğimiz yöntem, saldırıların sınıflandırma doğruluğunu tehlikeye atmadan boyutları verimli bir şekilde en aza indirebilmekte ve hesaplama yükünü azaltabilmektedir

Anahtar Kelimeler: Dağınık Hizmet Engelleme, Saldırı Tespit Sistemi, Ortak Bilgi, Ampirik Kümülatif Dağılım Fonksiyonu, Tekil Değer Ayrıştırma, Boyutluluk İndirgeme, Özellik Seçimi, MATLAB.



CHAPTER ONE

INTRODUCTION

1.1 Overview

With the swift growth of network technologies, security has become one of the primary issues now. One significant security menace is the Distributed Denial of Service (DDoS) attack. The continuous improvement of network speed brings new challenges to traditional detection methods. DDoS attacks make a victim decline supplying normal services on the internet by overwhelming an extraordinary number of malicious traffic. Attackers do not use security vulnerabilities in a networked system but launch attacks against their availability. According to the survey conducted by Arbor Networks [2], it has shown that DDoS attacks have proliferated in size of DDoS attacks. Toward the start of 2016, we noticed the most significant attacks being around 500Gbps. In the later months of 2016, they saw the adaptation of various IoT-based botnet DDoS attacks that were near breaking the 1 Tbps. Despite the fact that there have been many studies on detecting DDoS attacks, the proposed solutions cannot recognize flooding attacks from unexpected changes of legitimate activity.

1.2 Problem Description

In recent years, DDoS attacks and their effects have been increased dramatically. Not only DoS attacks got more prominent, but they also became more frequent and sophisticated. These attacks have several patterns like overwhelming the server using some exploits weak points in service by using software application, the attacks utilize all available resources on the destination device, and that consume the entire available bandwidth for the victim device.

The attackers are attempting to use computer resources of original users. In general, it consists of stakeholders' efforts to prevent the functioning of Internet sites and services to be executed efficiently or temporarily often. One such solution to detect the DDoS attack is IDS. The IDS ensures availability by checking traffic against malicious action employing a predefined set of signatures representing different attack behavior. One of the major difficulties with these systems it cannot detect suspicious behavior application if none of the predefined signatures matches the pattern of data being searched [3]. However, the amount of data that must be monitored and processed for the effective management of real-time networks and systems has become a significant issue in large-scale data. In real time the data comprises several features many of which are redundant and irrelevant. Since these features directly impact on them.

Also, many anomalies only become visible after analysis and examining data in traffic network from multiple locations. For data analysis, preprocessing stages are needed to clean up and convert the data to an appropriate format, so that sufficient mining is done. Preprocessing is especially helpful for data consisting of complicated structures and multiple dimensions. Consequently, miniaturization is performed as a preprocessing stage into which irrelevant and unimportant sizes are identified and eliminated. It is an important stage of mining data to help in improving the performance of machines learning. The main advantage of diminishing dimensions is to enhance data classification of data groups which improves the computational efficiency and improves data visualization. It is primarily because the server cannot keep track of the desired source. Besides, it is difficult to detect attacks without the specific features of the attack being accurately selected. Machine learning algorithms are not a novel approach to identify DDoS attacks, and the accuracy varies according to many data due to the massive traffic of the rapid growth in the types of internet intrusions and the multiplicity of cybersecurity methods demand scalability and durable. Also, over-fitting is a difficulty faced in the machine learning is over-fitted including a part of data. Also, many anomalies only become visible after analysis and examining data in traffic network from multiple locations. For data analysis, preprocessing stages are needed to clean up and convert the data to an appropriate format, so that sufficient mining is done. Preprocessing is especially helpful for data consisting of complicated structures and multiple dimensions. Consequently,

miniaturization is performed as a preprocessing stage into which irrelevant and unimportant sizes are identified and eliminated. It is an important stage of mining data to help in improving the performance of machines learning. The main advantage of diminishing dimensions is to enhance data classification of data groups which improves the computational efficiency and improves data visualization. It is primarily because the server cannot keep track of the desired source. Besides, it is difficult to detect attacks without the specific features of the attack being accurately selected. Also, over-fitting is a difficulty faced in the machine learning is over-fitted including a part of data.[4].

Over-fitting may become a negative influence on the DDoS attacks detection performance. Increasing the dimension of the dataset increases the size of the data region leading to sporadic data. In a typical situation these data are not uniformly spread over the search area, and usually, a larger proportion of training data is present in the corner. These data are more difficult to classify than the data at the center. Therefore to find statistically reliable results, it demands training data size. Consequently, high dimensions lead to problems known as the curse of dimensionality making classification with a high number of dataset dimensions particularly difficult [5]. One of the key steps in detecting DDoS attacks and analyzing traffic well is features selection and dimension reduction the number of features or variables for a smaller group, adequate to obtain necessary information for network management purposes. Given this one of the possible improvements to our current research is focused on selecting the best prominent features of DDoS attacks detection to increase accuracy detection and reduce the complexity of the model processing. It is worth mentioning that a few studies dealt with identifying features and reducing the dimensions for DDoS attacks.

1.3 Motivation

Dimension reduction and feature selection have an active role pre-processing phase in numerous of machine learning algorithms usage. As mentioned prior, a data set consisting of multiple features and a few instances may cause overfitting. Typically, the tracking of suspicious and variable traffic and new attacks may not be disclosed unless an intensive study of traffic and could not be disclosed in only a limited example. There are fewer cases of such attacks, but several features are

extracted. These features may not reflect their connection to an impact on a particular type of attack. These attacks can destroy the network and the basic. For this reason, it is necessary to find a solution to this problem. Moreover, the aggregated data sets may include interrelated features that may give better results when incorporated.

Hence, the motivation of this task is developing a method for feature selection which can detect interrelated features across a vast range of possible features and could be employed for DDoS attacks detection, also recognize the new attacks that can damage the networks or the user's properties by exploiting limited examples of attacks.

1.4 Objective

The principal objective of this thesis is to discover an appropriate methodology for detecting DDoS attacks. Increase the network security by detecting new types of DoS attacks to improve the network infrastructure performance by minimizing data complexity and processing time. The objective of our method is achieved by adopting features selection and reducing data dimensions to enhance the accuracy of machine learning detection.

1.5 Main Contributions

The contribution of this work is as follows:

1. In this research, we have developed a new model of preprocessing data which proposed as a combination between Mutual Information Feature Selection (MIFS) and Singular Value Decomposition Dimension Reduction Method named (FSDR) to estimate the relationship between output class and any input feature. Firstly, in the feature selection phase, MI algorithm uses to rank all datasets with weights scoring for each feature by descending order. Besides, Empirical Cumulative Density Function (ECDF) employs for expanding the variance between weights MI and then selecting subset features to depend on the uncertainty value in MI theory. The Negative scoring values are ignored because they are useless and positive values are chosen. This is considered the first guide to selecting a partial set of features. The final stage, we enhanced singular value decomposition algorithm SVD to eliminate noise by using the least number of dimensions. Maintaining

accuracy high was achieved which selects features depending on this methodology. It is not required to determine the threshold value manually for the candidate selection process as required in previous methods and studies.

2. We develop a DDoS and DoS system detector based on anomaly detection to improve the of abnormality detection of attacks by using Artificial Neural Network (ANN) as the classifier, also modifying the ANN structure and designing the back-propagation ANN algorithm to suit the nature of the datasets used in the search.

1.6 Thesis Organization

The structure of the thesis is arranged as follows: Chapter 1 introduces the objectives of this thesis. It begins with an introduction to the denial service attack. This chapter also describes the research motivation, description of the problem objectives scope and contributions. Chapter 2 surveys the literature of the related works in the same area of our research. Chapter 3 explains in detail background study in addition to the methods used to detect denial attacks as well as ways to reduce the dimensions and selection of features. Chapter 4 illustrates the datasets that will be used in our method. Chapter 5 demonstrates our proposed methodology besides some technical issues and configuration. Additionally, it illustrates the integrated phases of the proposed framework and the algorithm for detecting and algorithm combine for feature selection and dimension reduction. Chapter 6 demonstrates the simulation results which are presented and analyzed. Furthermore, it explains the experimental tendency and the implementation concerning the detection. Chapter 7 concludes this work with future work recommendations toward additional improvements.

CHAPTER TWO

RELATED WORK AND LITERATURE SURVEY

Research presented and evaluated the different approaches and methods to detect and reduce the DDoS attacks and their impact on the various networks environmentalists. Many methods and techniques have been prevailed to limit the effects of the attack. In this chapter, we display a synopsis of some related studies in the area of detecting DDoS. Therefore, this chapter has been divided into two parts. The works in the first group indicate to use DDoS attack detection methods to reveal the attacks, whereas the works in the second group review of the methods used to feature selection and impact regarding increasing classification accuracy and reduction computational complexity DDoS attacks in network traffic. These studies can be laid out in couple main categories.

2.1 DDoS Detection Techniques

In this section, we review the standard ways to detect denial of service attacks in the literature summary which are classified in four directions which are reported Monowar et al. [6].

1. Statistical based
2. Soft Computing
3. Knowledge-based
4. Data mining and Machine learning

2.1.1 Statistical Based

Statistical methods can be adapted to distinguish regular traffic patterns and attack DDoS attacks. Typically, a statistical model is built to identify the regular

pattern and the statistical model that is tested if it belongs to the usual pattern that has been determined in advance if it does not develop, and its pattern is of an attack type.

There are several statistical methods used such as correlation analysis between properties was used to distinguish DDoS attack traffic from normal traffic proposed N Hoque et al. [7]. The technique handles a Triangular Area Matrix (TAM) to collect correlation values among the features. The real network was divided traffic into multiple time windows, pre-process the captured data and then extract three features by entropy. The threshold value has defined an attack when the test profile deviates from the normal profile with a value greater than the threshold. The method gives high detection accuracy for the CAIDA and TUIDS datasets.

Özçelik and Brooks [8] suggested a novel DDoS detection approach: Cumulative Sum - Entropy. In this method, the extra coding technique is introduced in the entropy of the packet header to improve the detection efficiency. The presented method examined approach the usage of operational network traffic and performing DDoS attacks on university traffic network used with the researcher employed to test the method inside the university network. Results showed increase detection rate and lessening false positive rates and outperform the detection method utilize the entropy of packet header field. Entropy provides the measurement of the level of disorder and data correlation, the entropy alteration when an attack differs based on noticed packet header field. Wavelet was used according to filter out the long-term variants of the observed entropy values to decrease the number of false alarms.

Lee et al. [9] adopted a method to identify and control DDoS attacks proactive performance to use of cluster analysis. Identify the essential features affecting the service distribution attack by calculating the entropy equation to select features. The author identifies nine essential features were selected using the entropy such as the source and target IP, port number, number of packets and packet rate as well as the packet rate for the ICMP, UDP, and TCP SYN protocols. They create groups include the usual traffic other group containing abnormal traffic. 2000 DARPA Intrusion Detection scenario specifics was used to evaluate this detection method. As a result, that technique divided datasets into normal part and attack part.

Al-Mamory and Ali [10] created a method to detect DDoS attacks. This can be done utilizing entropy concept to measure the irregular change in traffic by the phases of the attack, next applying algorithm DBSCAN to classify data in the normal and

attack status. The models for DDoS traffic is constructed based on extracted centroid points from every cluster, which was used for testing phase using Distance-based classification. They applied the Euclidean distance function to calculate the distance to the cluster centroid of the corresponding traffic class. The author identifies nine essential features were selected using the entropy such as the source and target IP, port number, number of packets and packet rate as well as the packet rate for the ICMP, UDP, and TCP SYN protocols. The experiment was made on the DDoS attacks dataset LLDoS10 from MIT Lincoln Laboratory to train performance an investigation into his detection system in a real network environment.

2.1.2 Soft Computing

There is an increase in the use of learning models such as radial basis functions, genetic algorithms, and neural networks to detect denial of service attacks and their ability to classify automatically and in intelligent ways. The impressions of the DDoS attacks have been analyzed, and relevant factors that affect the attack were analyzed.

Big data used in the detection system is capable of analyzing the high velocity, and high volume network flowed in real time. It separated the actual and attacked traffic efficiently. Hsieh and Chan [11] proposed DDoS detection technique based on Neural Networks identify the features of packets effectively, carried out in the Apache Spark cluster, by using 2000 DARPA LIDoS 1.0 dataset. Seven features are used to DDoS attack like durability Number of Packets, Average of Packet Size, Time Interval Variance, Packet Size Variance, Number of Bytes, Packet Rate, and durability Bite Rate. The system is compiled from the open source big data computing framework Apache Spark and drawn up by R language which is low-cost and easy to implement. The result exhibited the accuracy is about 94%.

Barati et al. [12] Genetic Algorithm (GA) and Artificial Neural Network (ANN) are used for feature selection and attack detection sequentially in the hybrid method. Genetic Algorithm GA applied to select best features, next apply Multilayer Perceptron MLP to increase the DDoS intrusion detection rate. The datasets CAIDA UCSD 2007 anonymized traces from a DDoS attack, and The CAIDA UCSD Anonymized Internet Traces 2013 a normal traffic is used in the experiments to evaluate the method. Results are showed method could detect DDoS attack with high accuracy and reducing False Alarm

Oo and Phyu [13] use data mining function based on K-Nearest Neighbors method increase detection and classification attacks the traffic pattern to regular and numerous attacks system was proposed consists of two main stages. At first, the selected features are extracted from packets based on these features values. The proposed packet classification algorithm is developed. Then, K-NN is used to compare with the result from the proposed algorithm. Extract features of different packets indicate the nature of the DDoS attack in traffic data. These features can be used to identify DDoS attacks Time Interval Variance, Packet Size Variance, Packet Rate, Bitrate, No of SYN, No of ACK, No of FIN, No of PSH The result reveals 98.92%, 95.39% for attack and normal accuracy and lowest error rate.

2.1.3 Knowledge-Based

In these knowledge-based methods, a network is scanned for changes, events, and attack types that target the network and the entire network is examined. Attacks are often formulated to identify facts and the reality of attack types being carried out on the network.

A low value for the threshold allows for faster attack detection, but it also raises the number of false-positives. According to in Purwanto et al. [14] presented method developing an agent to observe the packet traffic rate outgoing and incoming based on identifying initial deployments. Normal connections on TCP can be determined by the degree of transmitted packets to received packets from an assigned destination. The outcomes revealed that the value of the traffic gave a great value at the start of execution If there is not the adequate packet to determine if the traffic is legitimate. Data was provided by Cistech Limited, by using Nakina Systems IP Solutions. Many kinds of research, they only converged to detect the traffic anomaly, however not to distinguish the types of abnormality that were identified such as flash crowd, models of a botnet and types of DDoS. The author does a survey to discriminate anomaly traffic detection system based on process and capability focus on anomaly detection system method comprising traffic features. We focus on each function of pre-processing and detection process .We detect the only abnormality based on each important research subject to be solved. Types of an anomaly, and prevention system that include the process to overcome the attack like the statistical pattern, forecasting

techniques, find essential data by information theory, soft computing, and signal processing.

2.1.4 Data Mining and Machine Learning

There is little diversity in statistical methods and data mining. The strategies of data mining are designed to take advantage of big data collections that make the statistical approach incapable of processing it. Eventually, the statistical process tries to create hypotheses from well-defined datasets, but the data mining focuses on producing hypotheses from unstructured and unknown data.

One of the significant difficulties in the detection regarding an application layer DDoS attack is the non-availability of features to detect so attacks. Yadav and Subramanian [15] proposed the model following classifying the application layer DDoS attack traffic handling feature learning by Stacked Auto-Encoder, first, learn to feature through and consider them, so features of application layer DDoS attack datasets, then a logistic regression classifier used for classification. Moreover, abstract features are learned as Source IP Address, Timestamp, Time Zone, URL, Response Code, Number of Byte Sent, Refer and User Agent by adjusting the layers in the SAE to be able to detect the attack in the application layer according to the specified feature. The is created in Smart and Secure Environment (SSE) laboratory. It contains different types of application layer DDoS like request flooding, session flooding, and asymmetric attacks. The results of the test showed purposes of classifying the attack traffic from the normal traffic within average detection rate of 98.99% and an average False Positive Rate of 1.27%.

Data mining algorithm is utilized a DDoS attack detection model. Keerthika et al. [16] offered detection architecture aim capture the shift of Web traffic caused by attacks under the flash crowd and the entropy of the recognized appropriate data to the HsMM can be a measure of the anomaly. The suggested system is employed PCA, ICA, and HsMM. The test was handled with diverse applications of DDoS attack forms. The dataset employed from a real trace of attacks. Results were shown during the detection threshold of entropy is set 5.3, the DR is 90%, and the FPR is 1%.

According to Kato and Klyuev [17] using network packet analysis and utilizing machine learning techniques and algorithms SVM with RBF (Gaussian) kernel to train and test the DDoS attack detection system to study the patterns of DDoS attacks. The

authors prepared analyzed large numbers of network packets provided by CAIDA three types of datasets and five features. The accurate detection method is in identifying the DDoS attacks. The extracted some features including the source IP address, destination IP address, the time interval in seconds between packets, and packet size in bytes from the dataset. The authors experiment detection system was more than 85% accurate with all types of the dataset and 98.7% specific with five features.

2.2 Literary Summaries With Methods of Selection and Reduction of the Datasets

In this second set, we will review some literary summaries that relate to methods of selection and reduction of the dataset. Most of the researchers in the first part of this chapter, which deals with means of detecting the attack, did not elaborate on the features and characteristics and ways to reduce the distance and how to choose. Therefore, we will address this section, which relates to the primary purpose of this research. Using all features from the datasets can cause significant memory and disk usage and make the detection phase very slow. Therefore, the aim of feature selection is choosing the most characteristic features which may have the most discrimination power over the dataset. Sasikala et al. [18] introduced a multi-method of filtering consisting of several stages of extracting feature minimizing dimensions by PCA and the author method of features selection correlation based Feature Selection CFS. The following step is a method of making an order of features in the new subset selected by Symmetrical Uncertainty (SU) as specified and classification. Before being chosen for the final stage by the SVM algorithm to select the efficient subset of the dataset for an examination on the enhancement of detection accuracy and best feature subset selection. The method applies to 22 different medicals dataset.

Liao et al.[19] investigated the difference in user behavior based on a weblog, so they introduced a series of 14-dimensional feature space based on user behavior (normal, attack) to describe properties of user behavior. The studying showed the relative variations or similarities in DDoS attackers or regular users. Durability applying three classical data mining classification algorithms are adapted to the detection, which are Naive Bayes, RBF Network, and C4.5. Empirical results showed that advised features are proper to differentiate legal users or attackers of the

application layer. Dataset was used came from two parts and divided four types of attacks, the first part is derived from an actual Clark Net-HTTP dataset, the second part records are generated from the emulator. However, there is a problem increase the span on the time window and consider real-time detection. The results show the correctly classified instances are quite high around 99.98%.

Balkanli et al.[3] exhibited the best informative features for a robust detection performance on Backscatter DDoS traffic are located to be the frame. That uses C4.5 classifiers trained on initial attack traffic is possible, by the feature selection techniques, namely Chi-Square and Symmetrical Uncertainty, are applied to build different C4.5 Decision Tree classifiers. The results revealed that the auto-generated rules by the C4.5 Decision Tree classifier using only seven features could reach more than 95% precision in detecting new Backscatter DDoS attack patterns.

Buragohain et al. [20] proposed detection method that utilizes a set of feature selectors using PCA-based feature selector and correlation-based feature selector. An attack detection method was presented that assists in identifying various attack scenarios employing a statistical method more precisely a method based on information gain. The statistical approach towards the network analysis based on deviation from the standard conduct of the network traffic is presented. The algorithm was divided into three segments of analysis for rate, packet, and protocol with employed CAIDA dataset in test stage method. The attacks are first divided into various categories using threshold for each one. These limits are based on experimental results.

Robinson and Thomas [21] proposed to decrease type false positive and type's false, harmful errors, boosting precision, recall and keeping detection accuracy by employed, supervised ML algorithms to evaluation and ranking of some features. This work suggests the effectiveness of performance-based classifiers notably the performance algorithm of Ad boost with Random Forest as the base classifier. A group of available datasetss such DARPA scenario-specific datasets CAIDA DDoS attack 2007 and CAIDA are utilized to evaluate the algorithms. The features of the datasets considered Average Packet Size, Number of Packets and Time Interval Variance, Packet quantity Variance, In that paper, the software-defined network SDN is employed to deterring and detecting DoS attacks. Research shows that the use of SDN brings a new opportunity to defeat denial of service attacks in an electronic computing

environment. The new features will be extracted to reduce DDoS attacks. Per of Bytes, Packet Rate, and Bitrate.

Fadlil et al. [22] developed the work of IDS by identifying the reduced features Information Gain, and the Gain Ratio was adopted as techniques for features selection. Additionally, utilize the Feature Vitality Based Reduction Method (FVBRM method) for getting and distinguishing the reduced group of features which are essential. The number of features is reduced for intrusion detection by employing one of the efficient algorithms Naive Bayes that perform this purpose. In this way, the NSL-KDD dataset of 41 intrusion detection features reduced to 24 features produced by the reduction based on dynamic features.

Jia et al. [23] suggested detection system of DDoS attack build on hybrid heterogeneous multi-classifier ensemble algorithm and design a heuristic detection algorithm based on (SVD) Singular Value Decomposition. Ensemble classification model comprises three main modules (primary data processing unit module, the heterogeneous multi-classifier detection module, and classification result module). In the first model group. The first training dataset is first divided into disassembly training subgroups based on SVD. The new training data subgroup is created by converting an independent linear basis. Second, the first test datasets are further subdivided into data subsets identical to the original dataset's training features. Eventually, new test data subsets are created by the forest rotation. In the different multi symmetric detection module. The new subsets of training data and subsets of test data are utilized as inputs for element classifier. Then, the results of the detection are obtained. In this third part of the model, the results obtained from this model are collected. The voting system based on the significant voting method (KDD) Cup 1999 dataset employed to feature selection and classification. Experimental outcomes indicate that the detection method is suitable and in TNR, the detection is 99.6% or more sensitive and stable. Feature selection presents an effective technique to examine the dataset.

Harbola et al. [24] tried to examine the NSL-KDD cup 99, dataset applying various classification algorithms. Firstly, tests are prepared in the environment of WEKA. The accuracy of the various algorithms is as well calculated. The accuracy is improved and the algorithmic running time is also reduced. The Greedy Forward Selection method which runs on a greedy method for feature selection.

Selected 20 attributes from 41 features. Classification accuracy of the algorithm KNN and the rate was approximately 100%.

Osanaiye et al. [25] used the method that named an "ensemble-based multi-filter feature a selection of EMFFS." This method brings together the output of four filtering methods to select features to achieve and improve the selection of features. In this technique, four types of filtering methods were used individually for an initial range of features. Methods have been used IG, gain-ratio, chi-square, and Relief filter techniques are utilized to rank the feature set of the original dataset to set up an individual group before setting the threshold to a one-third division of the ranked features of forty-one features. The results of this method will be by merging the output of each filtering method and adopting a simple plurality vote to define the selected final feature. A threshold is chosen to determine the attributes that are showed between the four filtering means and set to 3. Next, merge all the specified feature sets, a counter is employed to define the common features of the threshold set. The features that satisfy the threshold criteria are assembled and then classified. The results show amounted to 99.67% accuracy.

Khan et al. [26] focused on identifying some of the most likely attributes of an attack denial service based on the calculation of weights using entropy calculations. The proposed method relies mainly on using Shannon entropy and granular to calculate the weights to select the most important features of their impact in the denial of service attack. This method is based on several steps which we will briefly mention starting with entering the dataset related to denial of service attack. Here is the NSL-KDD dataset most frequently employed is used. The normalize data to be formatted and are valued between 0-1 using the Min-max function, calculates the probability of the features in the normal state and the case of anomalies, calculates the weights of the features in case of the abnormality using the entropy and then calculate the average weight of each feature. A granulation property is then applied to improve features in the case of anomalies based on their weight calculations logarithm complexity. Elements can be considered to have an essential value after exceeding the specified threshold.

2.3 Summary

A detailed survey of the various problems in the detection of denial of service distribution attacks and advanced work algorithms has been carried out in detecting such attacks in diverse literature in the world so far. This chapter has been divided into two parts. The first part included the methods of detection of attacks, which were identified in four ways, including the methods statistical analysis, methods of prior knowledge of the characteristics of the network and attack, techniques of soft computing and methods of data mining algorithms. The second part of this chapter combines methods of detection of denial of service attacks as well as ways of selecting features and minimizing the dimensions of the features. Limitations of literature presentation used only methods for picking features and detection algorithms or choosing means of reducing dimensions with detection algorithms. The need for a conventional approach involves several ways to minimize dimensions and methods of selecting features with detection algorithms by maintaining data disaggregation escalating the detection rate and limiting time processing.

CHAPTER THREE

BACKGROUND

3.1 Introduction

In this chapter, the focus is on the various research works that have been conducted so far at the identification of DDoS attack and intrusion detection. The survey has been of the types DDoS using different approaches. The attack was revealed, Such as vulnerabilities in the attacks and the presentation of the most important algorithms used in the field of detection in detail.

With the rapid increase in the number of services supplied the Internet entirely. The number of attacks of these services are by the same token increased. Any matter of compromising confidentiality, integrity or availability of online or offline computer system is called an intrusion. Many attacks caused a lot of corporate financial losses, and we will review some of these attacks. In 2017, WannaCry is deemed the latest attack incursion that deadened computers in at least 150 countries. It is estimated financial losses could increase to billions of dollars, one of the most damaging occurrences including the so-called Ransomware. Cyber's risk modeling company Cyence estimates the potential cost of hacking at \$ 4 billion [27, 28]. This statistic displays average costs of cybercrime attacks within chosen countries in 2016. During a survey, it was observed that the average cost of cybercrime attacks in the United States equaled to 17.36 million U.S. dollars. Germany ranked second, consumption for an average of \$ 7.84 million per company attack. According to a survey of global companies in August 2016, the business turmoil was the most expensive for the business that is the subject of cyber-attacks. Overall, the commercial service sector had the highest yearly costs affected by cybercrime. Anderson[29] reported design security controls of an operating system of each user that lead to control the much

components of the operating system. Security audits trails perform a very significant role in security software of computer systems. Providing how to supply the user with resources and multi-level information sharing systems against the threat of the malicious user. Improving checking then monitoring is the essential the capacity of client systems, Figure (3.1) shows the cost of cybercrime in individual countries in August 2016 (the US \$ 1 million) [30].

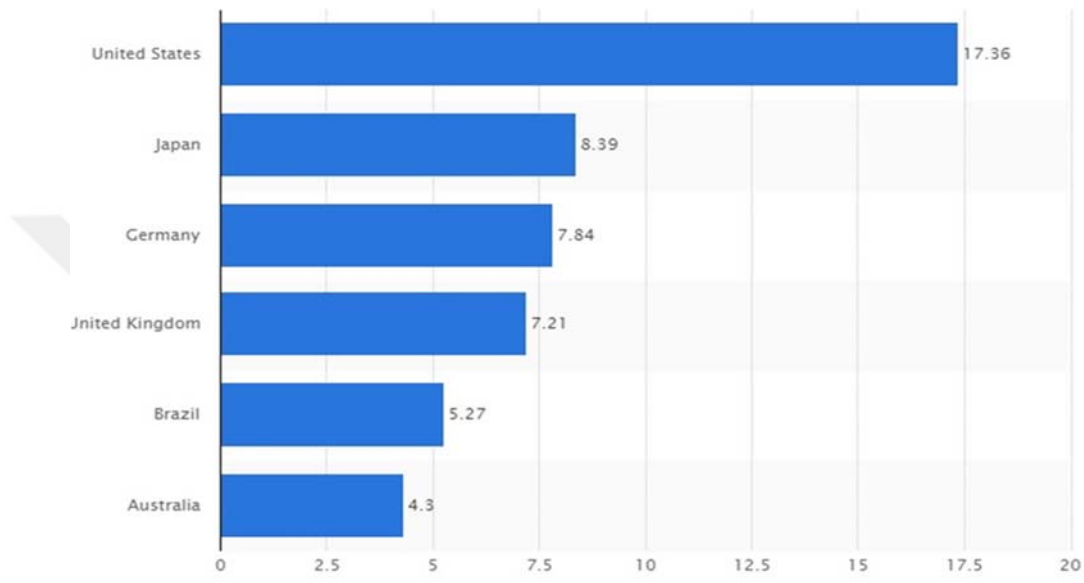


Figure 3.1: Average costs of cybercrime.

3.2 Intrusion Detection System IDS

Intrusion detection is the procedure of surveillance and enrollment events in network systems, computers, and analysis to obtain indicators of those incidents that occur and potential access[31]. These functions can be summed up, including threats and breach of security policies or standard security policies. Attacks and threats have several causes, such as malicious programs, attackers getting unauthorized access to systems and misuse by authorized system personnel for unauthorized privileges. Working of IDS is based on network measurements of the normal state where when the arrival of any traffic cannot be identified. The IDS sends warning messages. These are called suspicious traffic, and sometimes false alarms and traffic may be normal traffic. The basis of IDS work is a filtering of traffic based on the rules, and natural policies of the network and many programs provide the service Bro IDS, Suricata, Snort and many other applications. There is a difference between IDS and IPS. The

first is filtering the traffic network, detecting suspected cases and giving warning messages. IPS Intrusion Prevention System is a setting that blocks or blocks suspicious traffic identified with IDS [32], which composed of four parts:

1) E-ingredients (“Event-boxes”) is the first part of IDS, and its purpose to monitor and record the events of the target systems and then it obtains the event information and then it is analyzed at another stage.

2) D-ingredients (“Database-boxes”): In this section, information is stored in the data box from the previous stage to be processed and analyzed in subsequent stages.

3) A-ingredients (“Analysis-boxes”): In this component, analysis, processing, and monitoring are performed to detect and classify anomalies and create a kind of alert in the crucial cases.

4) R-elements (“Response-boxes”) are the final and key part of the IDs stage and it is responsible for making the appropriate decision in case of any security violation.

The Figure of (3.2) displays the general structure of IDS according to [C007].

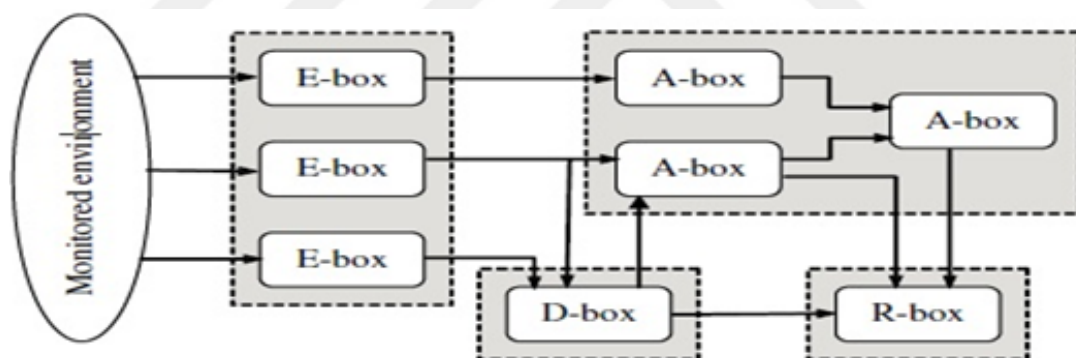


Figure 3.2: Common CIDF structure for IDS system.

3.2 Type of IDS

IDS can broadly be categorized into two parts according to [33]:

1. Analysis technique performed to recognize intrusion, which is divided into misuse IDS and anomaly IDS.
2. Data sources used in intrusion detection methods, which is divided into Host and network-based.

3.2.1 Analysis Technique

1) Signature-based: misuse or signature, a system to detect patterns of attack, this method is based on keeping types of attack in the database and are comparable to network traffic to detect incoming malicious traffic. Signatures are specific attack patterns, features, and conditions that are stored in the IDS database. This IDS class has several advantages, including the identification of offensive predictions in a transparent and fast manner and the low positive false rate. Regarding negatives, it is difficult to detect attacks that are not in the database. Advance knowledge of the types of attack knows the environment in which the attacks are handled. The updating database to attack continuously to keep up with the modern attack types. Moreover, the alerts are raised to the behavior of the undefined with the knowledge that the action can be positive and correct detection.

2) Anomaly detection is a way to detect abnormal traffic behavior in the network. This method is used to identify suspicious traffic on the training of the traffic network in which is normal condition and then compare the network traffic entering the system with the normal behavior, if not very close to the launch of warning messages. It is worth mentioning the training of the normal functioning of the traffic network becomes IDS More sophisticated and efficient. Part of the training data is used for known anomalies and attacks so that the system can identify them. This category of IDS has positives by detecting several types of new and unknown attacks.

However, this is at the expense of a high positive false rate. The definition of intrusion detection rules is not easy, and the efficiency of the system depends on the durability of the rules and their testing on datasets. Anomaly IDS uses different types of processing techniques, such as statistical analysis, knowledge base and machine learning, to understand the normal behavior of the network and therefore it is more complicated than signature IDS. There are examples of IDS such as Snort and Bro-IDS detection programs

3.2.2 Data Sources

Host-based IDS is used to scan, monitor and analyze events for only traffic of the host computing system. HIDS is a type of software that is installed on host computers such as Enterasys Dragon HID, the keylogger, Symantec ESM and other

programs. HIDS can be used to test all behavior on machines from the number of unsuccessful entries and to provide a complete real-time image of the user's activities. HIDS investigate features that attacks escape from NIDS. It can detect local events by HIDS. HIDS roles in the main system, encrypted and existing traffic shall be decrypted, and conflicts can be identified during applications. In terms of negatives to require much effort in the installation and management and some types of attacks cause loss of the performance of HIDS functions. In addition to, the vulnerability of attacks and the DoS system audit logs to occupy a large area of the space, which affects the performance

Network Intrusion Detection Systems (NIDS) is a type of IDS deployed at the network level to distinguish the malicious action in the network such as the attack on the network and the unexpected and the scanning of the ports and the volume of traffic by monitoring network traffic. NIDS has many advantages as it is easy to manage and does not require large expenses on individual machines. It is not liable to direct attack. The downside in NIDS, its failure to detect attacks when the network size becomes too large, many network devices as switches have limited access to port control and the inability of NIDS to analyze the encrypted data which makes some traffic invisible during processing. NIDS operates on the principle of signature identification. Among the types of NIDS encompass Cisco NIDS, Snort, and Netprowler. NIDS is more suitable for organizations and private networks with many users due to their size of data and resources.

3.3 Denial of Service Attack (DoS Attack)

Denial of Service (DoS) attacks can be defined as an explicit effort to block access to services. It uses flood links DoS attack to cripple network services, programs running on the server, and tedious server resources. Those attacks hinder genuine clients from accessing the network service[34]. As the name implies, a DoS attack to perform a network, host, or another piece of infrastructure unusable by genuine users. Web servers, e-mail servers, DNS servers and institutional networks can all be subject to DoS attacks. DoS attacks divided into three kinds:

1) Vulnerability attack:

This includes sending a few explicit designed messages to a vulnerable application or operating system running on a targeted host. If the proper sequence of

packets is sent to a vulnerable application or operating system, the service can prevent or, worse. The host can crash.

2) Bandwidth flooding:

The attacker sends a flood of packets to the central host such a lot of packages that the target's access links will become clogged, stopping legitimate packets from seizing the server.

3) Connection flooding:

The attacker builds a high number of half of the open or fully open TCP connections to the goal host that it prevents valid accepting connections. Despite the fact that the attacker seldom makes use of his machine to perform the DoS attack, it commonly includes only one attacking machine and internet connection. Figure (3.3) describes an easy DoS attack from three attackers. Although it is viable that different attackers perform the attack of one victim in parallel, each attacker uses a most straightforward one machine to employ the attack.



Figure 3.3: An easy DoS attack from three parallel attackers.

3.4 DDoS Attacks

One of the well-known threats to cybersecurity is Distributed Denial-of-Service (DDoS) attacks in which the victim network is bombed by the whole of the worthy number of attack packets originating from many computers. The hosts that operate these tools have an attack that they call zombies and can be attacked by an attacker taking something. A successful DDoS attack manages to degrade the subject and cause serious harm to critical infrastructure elements. A Denial-of-Service DoS attack is a malicious try to consume a service and thus denying reach to it for legitimate users. A Distributed Denial-of-Service (DDoS) the attack is a subset of the DoS attack, which determines suggests, which is carried out in a distributed fashion. Attacking obtains

by attackers commonly employ what is called a botnet to execute the attack [35]. Performing the DDoS attack achieved by the attacker determines to discover the vulnerable hosts which can be utilized to start the DDoS attack by applying the variety of survey methods named random scan, multiple scan list, topographic scan, local subnet scan, scan switch, and partitioned permutation scanning. After knowing the vulnerable devices, the attacker injects malware or attack tools into the device by utilization the weakness in the machines[36]. The Figure (3.3) below illustrates the DDoS attack structure.

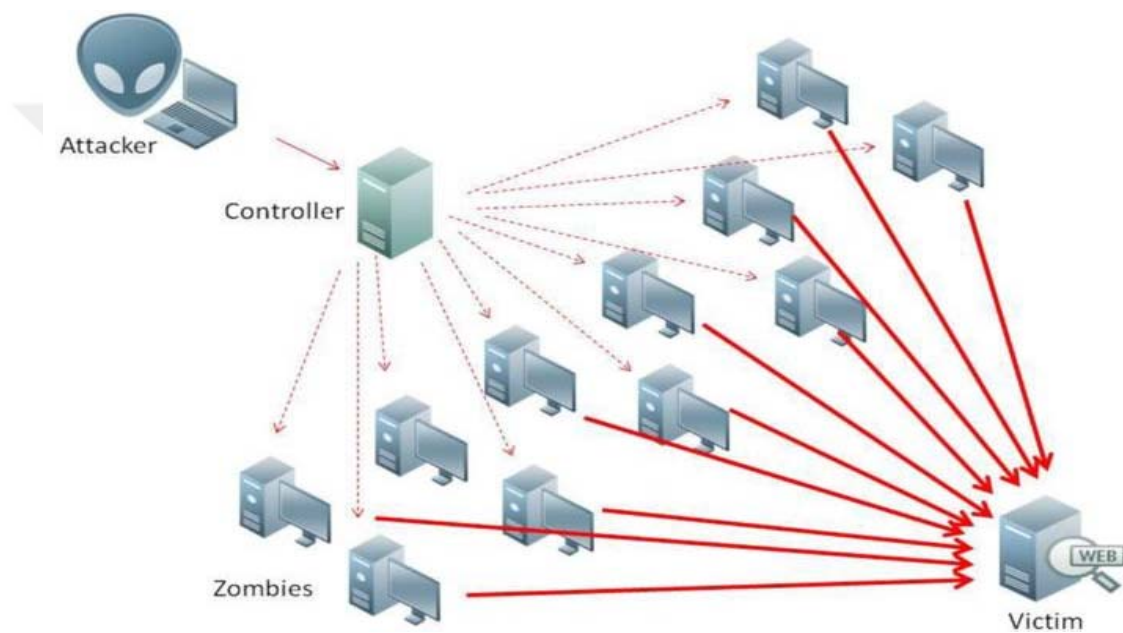


Figure 3.4: DDoS attack structure.

3.4.1 Methods of DDoS Attack

There are several methods of attack, for example, ICMP Flood attacks, and the attacker was executed for sending a non-stop of packets to submerge the victim's device. The extreme size of traffic controls the sources of the targeted machine, running the CPU cycles, memory, and network bandwidth or packet buffers. A simple bandwidth exhaustion attack could make the most the throughput limits of servers or network system employing sending vast amounts of small packages and conquer the available resources. The purpose attacks perform reducing of the system and result in an entire website shut down [37]. Attacks software or logic did not immediately utilize weaknesses in TCP/IP protocols or network applications. Instead, they use the

expected behavior of protocols such as the TCP, the UDP, and IMAP for the attacker's benefit.

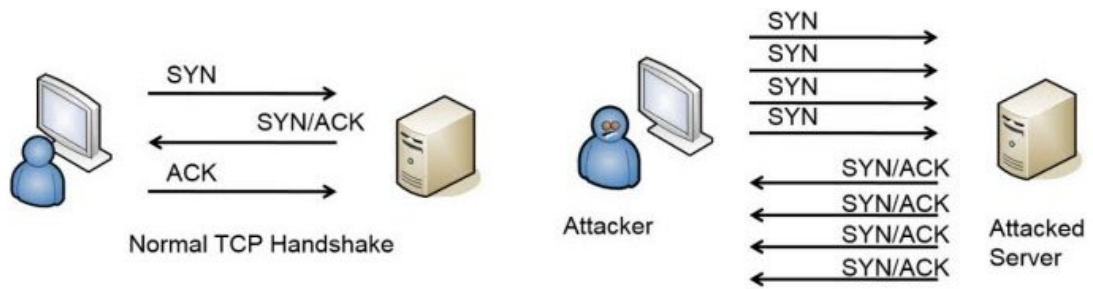
3.5 Types of Attacks DoS and DDoS

3.5.1 Smurf Attack

A Smurf attack is a kind of DDoS attack named amplification attack, which is amplified network traffic through the compromised systems movement before reaching the victim machine[38]. The attack was carried out by dumping the victim's computer using the ICMP echo and responding messages. Ping requests are sent to a specific broadcast request. Counterfeit the IP address of the attack source and send it to the victim's device address. The computer receives the broadcast address range and responds to ICMP echo messages.

3.5.2 TCP-SYN Attacks

Attacking DDoS resources are created by the attacker by sending packets, the purpose of these attacks is determined by attacking the network protocol communications, which leads to choking of the network and thus depriving the legitimate users of the service TCP-SYN attack[39]. This attack occurs when the handshake operation is not completed between the sender and attacker before sending data packets in full. The system beginning broadcasts a SYN request and the system will receive answering by returning an ACK forward with its own SYN request. After that, the sender will send back its ACK to be connected to the two systems. The receiver will send another packet of the SYN packet after a period, so the resources of the processor and the memory will be satisfied to the recipient after repeating the process several times because the system will keep the request until it becomes time-out. However, that does not reply way received, and server resources are consumed [40]. In this case, TCP-SYN DoS attack to occur to an efficient and powerful method when the zombie sends delusive requests from TCP-SYN to the victim. The next Figure (3.5) depicts TCP-SYN DDoS attack.



TCP handshake and SYN flood attack

Figure 3.5: TCP SYN flood attack.

3.5.3 Neptune Attack

It is a type of attack that we can distinguish between the normal traffic and attack by scanning the large numbers of packets directed SYN packets to the device (target) comes with the host and thus cannot access it [41].

3.5.4 Teardrop Attack

This attack exploits vulnerabilities in Microsoft systems as well as older Linux systems. Incorrect or interfere IP fragments are sent to the destination device and thus crash IP sent to the destination and an IP fragment containing duplicate payload was forwarded to the destination device. A virus in the TCP/IP code fragmentation reassembly of the OS caused various fragments to be mishandled and destroyed as a consequence of this action [42].

3.5.5 Back Attack

An attack that causes DoS in the Apache server, this attack sends incorrect Web requests to port 80 accompanied by many requests and malformed. The server cannot control many requests so the server will slow down. The IDS can detect this attack by examination the load of the packet in the network. The claims that carry more than some appearances should be examined in the payload attack [41].

3.5.6 Flood Attacks

Flood attacks are widespread. An attacker uses zombies to sending massive amounts of traffic to the victim's device, with a purpose to aggregate the victim system's network bandwidth alongside IP traffic. The system beneath attacks slows down, fails and denies getting entry to valid users. Flood attacks can be released the usage of each user Datagram Protocol (UDP) and internet control message protocol (ICMP) packets [43].

3.5.7 UDP Flood

In this type of attack, the attacker utilizes User Datagram Protocol UDP. The attacker sends the victim server network a huge volume to overwhelm of UDP packets by the zombies to particular or random ports on the victim's system. It is almost easy that UDP flood attacks are harmful because of the UDP protocol. That does not need the handshake mechanism to create the connection. Vast packets of UDP are sent to ports on the destination machines that effective bandwidth[42], CPU time and memory exhausted because the process of these packets causes the aim machines to become not able to be used for genuine users and the system failed.

3.5.8 ICMP Flood

In ICMP flood attacks, includes types of attack referred to as Ping flood and Ping of death. The ping flood is a particularly easy attack in which an attacker sends a massive range of ICMP echo request messages to the targeted system the intention of this attack is to overflow victim's buffer or to absorb the overall bandwidth of a sufferer so that not strong communication could be done to and from the victim device. Floods are not considered to be a completely public danger because many countermeasures were evolved in opposition to this type of attack [44].

3.5.9 HTTP Flood

This type of flooding attack web server by requests HTTP.Port 80 is leaving the HTTP default port without filtering on most servers to allow HTTP traffic through the

Internet Flood [37]. HTTP has considered the most popular kinds of attacks executed by Botnets. HTTP flood is a class of bandwidth attack application.

3.6 DDoS Attack Operation

Attack tools, DDoS attack tools have been created to block servers from one or multiple sites by overwhelming the victim with massive amounts of traffic. The single client is the one that remotely controls vast amounts of traffic created by various sites. The attack types differ according to about the tools used, as they are built according to communication between the handlers and the client. The tools do not catch data or penetrate the computing system, but rather obstruct normal network traffic to the host. There are many denials of service tools and will be addressed and explained the essential tools used in these attacks. These programs utilize the server and client structured to permit an attacker to redirect attacks to multiple machines[45]. Many of these programs do not require wide knowledge and experience to be used and are often available and easily accessible on the Internet. These programs can hide themselves during interruptions as well as erase any directory of their activities. Also these programs possible to uninstalled and disable itself when doing work and meet the specific conditions. On the other hand, traceability to these tools is not easy because these tools falsify the source address by manipulating the titles and by hiding their real location.

3.6.1 Flood Net

FloodNet is an application based on the language of Java, which immerses the target request pages. It do not utilizes a type of TCP/IP flooding in attacks.

3.6.2 Trinoo (Also Called Trin00)

Trin00 is one of the first programs designed to perform denial of service attacks. It used to be written for the well-known Solaris and Linux platforms. It consumes leads denial of service.

3.6.3 The Tribe Flood Network (TFN)

TFN began running after Trinoo. TFN consists of two parts of the program's client and daemon, the programs carry out a DDoS network able of applying a wide variety of attacks, like floods of SYN, ICMP, and UDP.

3.6.4 Stacheldraht (German for "Barbed Wire")

Stacheldraht was launched at the end of the summer of 1999. It is one tool for DoS that mixed functions of Trinoo and TFN. This tool can achieve several many of a kind DDoS attacks consisting of "ICMP flood, UDP flood, and SYN flood" which uses the encrypted TCP packet connection in the attacker.

3.6.5 Trinity

Trinity is a built on the Linux system for DOS attack tool utilized via the hackers to start a vast IP flood against a victim's targeted computer, like its predecessors TFN and Trin00 do. Trinity is able of releasing various types of flooding attacks on a victim site, like UDP, SYN, RST, ACK, and other floods.

3.6.6 Shaft

A Shaft network concerning the concept is analogous to a Trin00. It means flood attack packets attack. IDS is not easily detected. The "Shift" network consists of two sets of handler programs called "shaft master" and the other group of the client, which are larger groups called "shaft node."

3.7 Data Mining and Machine Learning

Data mining and machine learning are concluded of statistical methods. The data mining intends to examine existing data and getting answers for problems that can be fulfilled from existing data [46]. The data mining objectives to understand what is meant by for helping a human but the machine learning approach minimises human efforts and changes key decisions when new information is presented I will learn by doing. The method of machine learning is supervised or unsupervised. The

sophisticated tracking method requires the classification of the maximum probability ML classifier in advance correctly classifying [47]. Researchers employ data mining technic to detect attacks for improving the accuracy detection. Many pre-processing methods so as normalization, discretization, and fuzziness are practiced to increase the character of the training stage

3.8 Learning Models

The NNs have two different types of universal learning models, the first being called supervision of learning, the second type is called non-supervised learning as mentioned in [48].

3.8.1 Supervised Learning

The expression of supervised learning is learning that is supervised by a mentor with the knowledge of background from sample input and output. The mentor provides counseling for classifying whichever is normal and anomalous traffic as malicious, non-malicious. The learning of supervised IDS function can be described in several steps. The mentor analyses and labels part of the network connection. Then, labeled training data is employed via the learning algorithm to apply those rules in general. Classification algorithms adopt rules established to classify the new network data with an alert when the connection is identified as malicious. Multilayer Perceptron (MLP) is associated with the BackPropagation BP algorithm, which is one of the most common in supervised learning.

3.8.2 Unsupervised Learning

It is opposite the supervised learning, and unsupervised learning has no teacher to teach it whichever a bad or good traffic is. Unsupervised learning has the strength to acquire a knowledge of from unlabeled data as well as automatically generate new classes. The learning of supervised IDS function can be described in several steps by the employ of a clustering algorithm. We will be clarified how unsupervised learning works. The first stage, training data is compiled applying the clustering algorithm. A sample of the dataset from the cluster and naming the cluster center with the main type of sample is selected. In the final stage, the named weight carriers can employ to

classify network traffic connections. K-means clustering algorithm is a common learning of unsupervised NNs.

3.9 Concept and Application of Neural Network

Computers can be learned through practice. This is named machine learning. The process of learning is consisted of learning by examples and by analogy. Researchers working on machine learning are usually independent of real practices. New methods of classification can be developed by researchers, whereby comparing performance, for example, the accuracy of the dataset available to evaluate the performance of methods of learning [48]. New ways of classification can be developed by researchers where they are performed by comparison of performance, for example, the accuracy of the dataset for evaluating the effectiveness of the learning method. With the increasing capacity of the learning machine at present, the computer can adapt automatically to complex environments. Currently, based on the use of neural networks. In this thesis, the ANN algorithm will be adopted to predict the best results.

It is possible to deduce from the name of neural networks that they take the neural network signals from the human brain to simulate its structure. At the turn of the century, studies of neural networks began in 1940 by several scientists and researchers[49]. The brain processes information by these neurons and the information processing capacity of the human brain. It is much faster and more powerful than any computer present today. All neuron has a much-uncomplicated structure. However, a large number of neurons can be connected with each other to comprise an enormous and developed mechanism of processing. Neurons can be expressed as the primary neural unit in neural networks. The neuron in the human brain sends electrical signals from multiple sources. Among the neurons, there is a predominant weighted link connecting them as a whole. For this reason, the signal is transmitted from one neuron to another. The output signal of the neuron is split into several parts which transmit the same signal. The output portends with the input connections of other neurons in the network. Neural networks have been used as much as they are now.

3.10 Back-Propagation

Backpropagation algorithm is method to enable the error derivative calculation between the actual output and the target is derived independently for all weights in the network. In multi-layered perceptron NNs the BP algorithm is utilized to calculate the necessary modifications after random selection of the network weights. The work of the algorithm can be divided into four stages. It estimates the feed forward the output layer concerning BP the hidden layer concerning BP and updates the weight if the value of the error function is as follows the algorithm that is turned off if the error value, the role has matured or decreased sufficiently [50]. The next step illustrates the mechanism of the ANN algorithm. Here X is input dataset, Y is output products, and W is weight values sequentially. The θ is a correction required only for hidden layer and as well as after each repetition are continuously updated. The e is an error gradient value, and p represents the number of iterations. The σ represents error gradient values. The Figure (3.6) below shows the NN structure consisting of three layers: input, hidden and output using the BP algorithm.

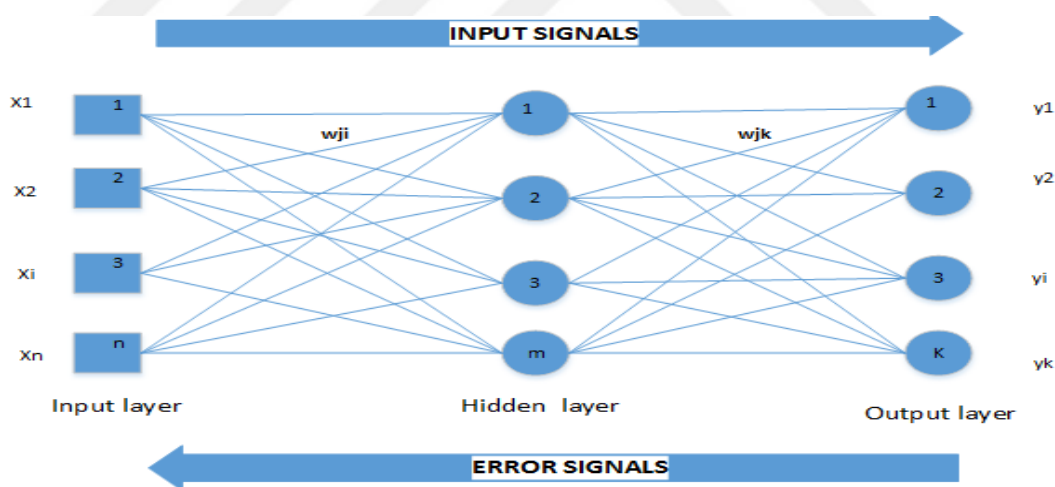


Figure 3.6: General structure of backpropagation based on neural network.

3.11 ANN-BP Algorithm in IDS DDoS

Today's analysis of statistical, expert system, artificial intelligence and many various approaches for attack detection systems. These systems have advantages and drawbacks. In these methods, due to artificial neural networks, ANN have several benefits and many people pay attention to it is like self-learning, self-organizing, and

capable of handling errors. The ANN can distinguish current attack types and detect unknown attack types it is beneficial to use artificial neural networks BP for IDS. Traditional neural networks have some inherent flaws such as easy weaknesses that breakdown to local minimums so set up some networks new neuro IDS for increases durability increasing intelligence and adaptation of IDS .This is leading for inclination the evolution of IDS. The data training in ANN obtained knowledge normal activity and attacks to perform an anomaly detection task [50]. In this thesis, NSL-KDD dataset was employed during the training stage. This dataset was designed and assembled by MIT Lincoln Labs. The target was to investigate and evaluate the intrusion investigation. A standard dataset to be observed is provided including a wide range of interventions imitated in a network of the military environment. It is design for each network connection 41 different attributes and qualitative.

After the training phase, the ANN algorithm learns the characteristic values of the normal state and different cases of attack. When entering any event to the network, it is regarded as at least 41 input values corresponding to 41 various attributes of the event. All of these entries transfer into the ANN's hidden layer. It is then transferred to the output layer. The output node is the outcome.

3.12 Data Preprocessing

Data preprocessing comprises a sequence of steps to turn the raw data from the data extraction into a clean and neat dataset before statistical analysis. Preprocessing is pointed at evaluating and enhancing the characteristics of data to allow reliable statistical analysis. It involves preprocessing data like data cleaning, data integration, data conversion, and data reduction [51].

3.13 Normalization

Normalization is one of the methods of processing data belonging to data transformation. It is performed by a preprocessing module and data for a ready machine learning. The goal of normalization is to accumulate all functions to a common scope so that specific functions do not take priority over other functions. Normalization makes it easy to compare different features in different scales. Also, normalization can accelerate the time needed to train a neural network[52]. In attack

datasets, not all parameters or field values are weighted equally. In such cases, normalization is evaluated as useful before applying an abnormality detection mechanism.

3.14 Dimensionality Reduction

Dimension reduction is often used for research data analysis whenever possible. It helps users to conceive data in low-dimensional spaces. The number of functions to be extracted is huge, and the dimensions are reduced to use the principal analysis of components and distribute maps. The distribution diagram is a multilateral learning method that integrates the original high dimensional space into the small dimensional space. Which make easy detection and aggregation of anomalies with this insertion space[53]. Dimension reduction indicates algorithms and techniques to form new attributes such as a nonlinear or linear collection of the original attributes, which degrades the dimensionality of the data[54]. Instead of selecting, a subset of functions, these techniques convert a particular type of entity, shrink the dimension so that the presentation is as respectful as possible to the original data set, and reduce the dimensions of the dimension's redundancy shrink it. Since the new attributes of the collection are original, the conversion process is also called function construction or function transformation. Creating a new function this process can be executed or combined according to the process of selecting a subset of functions. The original feature set is initially extended by the newly created functional next a subset of the features is employed. There are several methods to reduce the dimensions have been suggested in previous studies, but in this research will be addressed two types of them.

Principal component analysis (PCA)

Singular Value Decomposition (SVD)

3.14.1 Principal Component Analysis

Principal component analysis, also called PCA, is one of the most common approaches to reducing dimension projection. PCA obtains a lower dimensional exemplification of the original data by searching for a linear mapping from higher dimensional data to the lower dimensional illustration. This is achieved by projecting the primary data onto a linear subspace and losing data as much as possible. In PCA,

information is interpreted as the total variance of the original input variables. Accordingly, PCA can be understood as a way to derive from the set of variables from the reduced linear projection in decreasing order of variables. In 1901 Pearson found this technique is the lines and planes that fit well for the given points highly variable data. The PCA algorithm was developed by [55] compelling diagram for characteristics and applications. It produces new features such as linear compounds of the original variables. These new features are named main components (PC). It must fit the following criteria linear combinations of original features, perpendicular to each other and determine the maximum variation in the data[55] Relatively. Few PC can frequently capture data variability and consequently. PCA can obtain a big reduction in size with smaller noise than the original models. A disadvantage of the PCA is that PCs are not always easy to illustrate. Below we will explain the mathematical background of PCA.

The pros and cons of PCA are a general relatively simple, non-parametric method that is helpful for finding new, more informative, non-correlated attributes and can be utilized to decrease the size by discarding various low attributes. When the principal components are perpendicular to each other, each principal component is not related to the other principal components (i.e., it no longer contains information). The principal components are designed to calculate the highest percentage of variation among variables with as a few PCs as possible. For this reason, as a rule, the first PCs constitute a large proportion of the total variance, permitting only low dimensional data to be displayed.

Despite, PCA is restricted to re-representing data as a combination of its basis vectors. The significant disadvantage of PCA is that the potential interpretation of the PC becomes difficult because each PC is a linear combination of all the original variables. Conversely, in systems with many variables, PCA can use to project dimensions to an appropriate number of plots and rotate the principal components toward more meaningful representations [56]. Besides, PCA is sensitive concerning measurement units. If the group of the attribute and the variance are very different, variables with large variance tend to dominate the first few essential elements. In this case, the data must be normalized.

3.14.2 Singular Value Decomposition (SVD)

SVD “is considered to be one of the most important and common methods factorized matrix.” SVD provides an accurate description of any matrix, and As well as facilitate the removal of the insignificant parts of this representation to provide an approximate representation of any number of respectable dimensions. Where will we choose fewer dimensions and besides, the less precision will be for approximation. Suppose, we have $m \times n$ of matrix A of rank r into an output of three matrices [56]. As mentioned in the previous equation in PCA decomposes a square matrix toward an outcome of three matrices. .In spites of the fact that, there are no eigenvectors in non-square matrices. Therefore, it is possible for a random matrix to be decomposed in similar ways by the SVD theorem [57].

3.15 Feature Selection

These methods are used as filters for features assuming that not all features are important [58]. Some features may be redundant, and others are not useful in the classification process due to their random distribution or bias to one category without the other. We need to select features even if they have the most useful features, resource deficiencies, storage, and bandwidth and time reduction. So we need to filter the least essential and helpful information before implementing the classification.

The advantages of selecting features before the classification stage increase the learning process, improve predictive performance, better interpret the basic operation that generated the data and reduces the probability of high separation between the classes[59].The methods of selecting a feature have been analyzed into two categories: the ranking feature and a subset-feature. The feature ranking is a ranking given for each feature calculates the metric omits feature based on the worst performance. Subset selection examines the set of potential features for the best subset [58]. There are many different ways to choose the features according to the researcher [24].

3.15.1 Wrapper Method

The wrapper method is related to learning algorithms. In this mechanism, a classification algorithm used it. Which is defined previously in the selection of features

or subsets. The main problem in determining the quality of the search is in the space of all subsets that can change and require much time [60].

3.15.2 Filter Method

The filtering method involves evaluating the performance of specific features. This technique usually entails the concept of independent ranking features than the pick of the forecaster. Filter methods comprise parsing performance as a single varying classifier and information theoretical ranking criteria. A general technique concerning filtering features involves the ranking them according to how well they split sample data distributions accumulated of couple classes [59].

3.15.3 Hybrid Method

The hybrid method described this method is a merge between the wrapper and filter approach. The hybrid method employs a search algorithm to seek into the space of potential features and appreciate each subset. It achieves noted potential performance by preventing over-fitting and minimizing the complexity of conducting comprehensive research during execution.

3.15.4 Classification Algorithm

Essentially to classify is a prediction of specific output given input. Toward prediction of results, algorithms require being able to process the data set. It is anticipated to be a training set of key attributes and outcome. The data mining algorithm defines the connection among dataset attributes. It is potential to predict the outcome.

3.16 Mutual Information (MI)

Mutual information MI is an essential connotation in information theory. Also, it is broadly employed in artificial intelligence (AI). MI indicates the presence of the information shared among two variables and measuring the extent to which one variable is known will be reducing the value of uncertainty for the other [62]. Higher mutual information means that the uncertainty is largely reduced. Low mutual

information means a small reduced in uncertainty, zeroing shared information among two random variables intends that the variables not be dependent. This relationship is not necessarily linear.

3.17 Empirical Cumulative Distribution Functions

In statistical methods, empirical distribution function indicates the distribution function correlated by practical measurement from the sample. The Empirical Cumulative Distribution Function ECDF is a valuable assessment of the cumulative distribution function CDF to random variable distributions [61].

3.18 Summary

In this chapter detailed DoS attacks and their adversarial and dangerous effects were presented in detail. Also, the methods to detect the attack was explained by using multiple techniques and the use of ML techniques. Regarding the mechanism of classification of attacks using classification algorithms, and also revealed the tools of data processing such as the selection of features, reduce the dimensions and usefulness in accelerating the process of detection attacks reduce the time and increase accuracy.

CHAPTER FOUR

DATASETS USED TO IMPLEMENT AND EVALUATE PERFORMANCE OUR MODEL

This chapter is organized into two major parts, viz. Introduction and the existing dataset used in this thesis. Few data contain new types of denial of service attacks, Therefore, in this research two datasets will be used, which will be mentioned later in this chapter.

4.1 Introduction

When using intrusion detection network especially detection of anomalies, it is challenging to find accurate evaluation distribution and comparison of the system predictable to detect new attacks. Because of lacking appropriate dataset the IDS should test and evaluate the detection of anomalies system by employing genuine labeled track network traffic by comprehensive and updated set of different types of attack before applied in the real environment. It is a significant challenge specific the lack of many of these datasets. For this reason attack detection techniques is evaluated only with a limited number of publicly available datasets.

4.2 NSL-KDD

NSL-KDD dataset is the modified version of KDDCUP'99 [62] published in the literature[63, 64].It is worth mentioning, the Knowledge Discovery and Data mining (KDD) Dataset that is one of the widest and most famous datasets used in a wide range for intrusion detection. Stolfo et al. have prepared this set of data.[65]. It affiliated with the Lincoln Laboratory at MIT is based on the data caught in IDS evaluation approach [66].

KDD consists of two sets training dataset which contains about 4,900,000 million records and test data. It includes nearly 300,000 records with each sample containing 41 features. These instances are classified as either a normal case or a case of attack types in the dataset. The dataset is classified into five categories, four of them are attack types as follows (denial of service attack, the user to root attack, remote to local attack, probing attack) plus normal traffic. The researchers identified some problems in KDDCUP'99 by conducting statistical analysis of data. For example, the most important issues affecting the work of intrusion detection systems evaluating them. The results of a very ineffective assessment of abnormality detection methods have large and redundant records often causing bias learning data in the algorithms. Plus, there are complexity and high difficulty. Besides, to their complexity and high difficulty. Tavallae et al. [67] cope these problems .They suggested a new set of data from KDD called NSL-KDD. A set of records contains a whole KDD dataset. We will review the benefits of NSL KDD, which have been improved from the original KDD dataset.

1. Excess records from training data have been deleted to ensure that redundant data is not biased in learning classifier's algorithms.
2. Duplicate data records have been removed from the test group to maintain the performance of the not biased learning by methods that have better rates of discovery to the repeated records.
3. There is an inverse ratio between the numbers of records selected from each difficulty level to the corresponding percentage of the files in the original KDD group. Therefore, the different classification rates of the learning are mixed in a broader range, making them more efficient in assessing accuracy in different learning algorithms.
4. The results of the evaluation of the various research groups have become coordinated and are easy to compare with other research because of the number of records in the two groups. Training and testing have become reasonable. It makes it reasonable to run experiments completely without the need for random selection of data in small parts.

The dataset contains four categories of attacks. Also, normal traffic and is shown as follows:

1. DoS: The denial of service attacks are target victim resources, consuming them, they cannot be able to respond to authorized requests such as SYN flood, they are attacked, ICMP flood, etc.
2. Probing: surveillance and other attack investigations are intended to access victim information such as port scanning.
3. U2R: owning root properties of the primary user by unauthorized access to access the privileges of the individual account by exploiting existing vulnerabilities such as buffer overflow attacks.
4. R2L: an unauthorized remote access from a remote device that infiltrates the attacker into that calculator and aims to access the victim's machine such as guessing the password of its features using many attempts.

The NSL-KDD consists of 41 features. These features are divided into three types: basic features, traffic features, and content features. Among these are the core features such as duration, protocol, and service that can be derived from TCP / IP connections. Contents features that relate to the attributes through which they can detect abnormal behavior such as name failed logs, logged in, NUM compromised. Finally, by monitoring network traffic where it is calculated using the traffic feature a window period that a 2-second contest. That is also divided into same host features and same service features. We show 41 features of the NSD KDD dataset in detail in this Table (4.1).

Table 4.1: NSL-KDD dataset features.

<i>No</i>	<i>Feature Name</i>	<i>Type</i>	<i>No</i>	<i>Feature Name</i>	<i>Type</i>
1	<i>Duration</i>	<i>Continuous, Numeric</i>	22	<i>Is_guest_login</i>	<i>Discrete, Binary</i>
2	<i>Protocol_type</i>	<i>Discrete, Nominal</i>	23	<i>Count</i>	<i>Continuous, Numeric</i>
3	<i>Service</i>	<i>Discrete, Nominal</i>	24	<i>Srv_count</i>	<i>Continuous, Numeric</i>
4	<i>Flag</i>	<i>Discrete, Nominal</i>	25	<i>Serror_rate</i>	<i>Continuous, Numeric</i>
5	<i>Src_bytes</i>	<i>Continuous, Numeric</i>	26	<i>Srv_serror_rate</i>	<i>Continuous, Numeric</i>
6	<i>Dst_bytes</i>	<i>Continuous, Numeric</i>	27	<i>Rerror_rate 38</i>	<i>Continuous, Numeric</i>

Table 4.1 (Continued): NSL-KDD dataset features.

No	Feature Name	Type	No	Feature Name	Type
7	<i>Land</i>	<i>Discrete, Binary</i>	28	<i>Srv_error_rate</i>	<i>Continuous, Numeric</i>
8	<i>Wrong_fragment</i>	<i>Continuous, Numeric</i>	29	<i>Same_srv_rate</i>	<i>Continuous, Numeric</i>
9	<i>Urgent</i>	<i>Continuous, Numeric</i>	30	<i>Diff_srv_rate</i>	<i>Continuous, Numeric</i>
10	<i>Hot</i>	<i>Continuous, Numeric</i>	31	<i>Srv_diff_host_rate</i>	<i>Continuous, Numeric</i>
11	<i>Num_failed_logins</i>	<i>Continuous, Numeric</i>	32	<i>Dst_host_count</i>	<i>Continuous, Numeric</i>
12	<i>Logged_in</i>	<i>Discrete, Binary</i>	33	<i>Dst_host_srv_count</i>	<i>Continuous, Numeric</i>
13	<i>Num_compromised</i>	<i>Continuous, Numeric</i>	34	<i>Dst_host_same_srv_rate</i>	<i>Continuous, Numeric</i>
14	<i>Root_shell</i>	<i>Discrete, Binary</i>	35	<i>Dst_host_diff_srv_rate</i>	<i>Continuous, Numeric</i>
15	<i>Su_attempted</i>	<i>Discrete, Binary</i>	36	<i>Dst_host_same_src_port_rate</i>	<i>Continuous, Numeric</i>
16	<i>Num_root</i>	<i>Continuous, Numeric</i>	37	<i>Dst_host_srv_diff_host_rate</i>	<i>Continuous, Numeric</i>
17	<i>Num_file_creations</i>	<i>Continuous, Numeric</i>	38	<i>Dst_host_error_rate</i>	<i>Continuous, Numeric</i>
18	<i>Num_shells</i>	<i>Continuous, Numeric</i>	39	<i>Dst_host_srv_error_rate</i>	<i>Continuous, Numeric</i>
19	<i>Num_access_files</i>	<i>Continuous, Numeric</i>	40	<i>Dst_host_error_rate</i>	<i>Continuous, Numeric</i>
20	<i>Numoutbound_cmds</i>	<i>Continuous, Numeric</i>	41	<i>Dst_host_srv_error_rate</i>	<i>Continuous, Numeric</i>
21	<i>Is_host_login</i>	<i>Discrete, Binary</i>	#	_____	_____

The NSL-KDD dataset can be employed for training and testing. NSL-KDD comprises many files, KDDTrain+.ARFF (dataset with binary labels) used for training. KDDTest+.ARFF is utilized to solve binary classification problems. Files KDDTrain+.TXT and KDDTest+.TXT used for training and testing for the full NLS-KDD train set and test set. The KDD train + 20 percent contains a 20% subset of the total training package KDD Train+.Text file, KDDTest-21. Which includes a subset of KDDTest+ and Txt.training file on KDD. Training data is done on 22 types and of the four categories of attack. As for the test group, there are 17 attack types not found in the training group.

The NSL-KDD dataset comprises two sets of training and tests reaching 148,517 records. The normal state distribution of the training dataset is 67343 records, at the rate of normal (53.46%), 45927 DoS In addition to the rest of the attacks, we did not want to mention it because our research aims to detect denial of service attack only. The test set includes the normal number of records 9711 at the rate of normal traffic

43. 08% and 7456 at the rate of DOS 33. 08 %.It is worth noting the rate of DOS attacks against remain of the attack the whole rate was approximately 78.3 % percent. DoS attacks to occur in a network of six types of attacks extracted from a whole NSL KDD dataset normal traffic and DoS attack records.The chosen part of the dataset comprised of a training set including 113271 records and a test set of 15452 records. Six types of denial of service attacks were extracted from the training data four other types (Apache2, Mailbomb, Processtable and Udpstorm) of denial of service attacks from the test data. There are not referred to in the training data.Also, the normal state. Table (4.2) shows the number of records and name attack.

Table 4.2: Numbers and names of DoS attacks in NSL-dataset.

#	<i>Normal</i>	<i>Neptune</i>	<i>Teardrop</i>	<i>Smurf</i>	<i>Pod</i>	<i>Back</i>	<i>Land</i>	<i>Total</i>
<i>Training data</i>	67344	41214	892	2646	201	956	18	113271
<i>Test data</i>	9711	4657	12	665	41	359	7	15452

We will use in this thesis, KDD Train+.txt file for training and the KDDTest+.txt which is the full NLS-KDD test. Table (4.3) shows the name of attacks and number of records.

Table 4.3: shows the number of records and name attack.

#	<i>Normal</i>	<i>Neptune</i>	<i>Teardrop</i>	<i>Smurf</i>	<i>Back</i>	<i>Total</i>
<i>Training data</i>	67344	41214	892	2646	956	113051
<i>Test data</i>	9711	4657	12	665	359	15404

4.3 Alkasasbeh Dataset

The progression of the types of attacks and their diversity using modern techniques has become continual for hinder modern attack types with keep confidentiality, integrity, and service availability to users. It is an incentive for updating the IDS. There are many ongoing challenges for organizations and users to deal with DDoS attacks. Therefore, network security engineers must provide and maintain the service of attack types that cause resource consumption Alkasassbeh et al. [1]. The link is available for the Alkasassbeh dataset published in [68].

They provided a set of data containing modern types of the attack targeting different OSI layers such as Application layer and Network layer. We remark some types of attacks such as HTTP-Flood, SIDDoS. The new dataset has been selected for two reasons: the first is the presence of new kinds of DDoS attacks (HTTP Flood, SIDDoS, UDP Flood, and Smurf). The second reason is that many datasets contain a lot of redundant and duplicate records that affect data education and bias to a particular type without the other, so the results are unrealistic and inaccurate since this dataset is devoid of duplicate and redundant records.

The dataset was created using the network simulator (NS2). NS2 is widely used for its ability to produce excellent and valid results in real network simulation. This set of data contains four types of attack in addition to regular traffic and also comprises 27 features. The following Table (4.3) shows the types and number of attack in the set of evidence. The other Table (4.4) shows details of the features and types.

Table 4.4: of Normal distribution and attack in the overall dataset.

<i>Dataset</i>	<i>Normal</i>	<i>Smurf</i>	<i>UDP-FLOOD</i>	<i>SIDDoS</i>	<i>HTTP-Flood</i>	<i>Total of Records</i>
	1935959	12590	201 344	6665	4110	2160668

Table 4.5: Alkansasbeh dataset features.

<i>Feature No</i>	<i>Feature Name</i>	<i>Type</i>
1	<i>SRC ADD</i>	<i>continuous , Numeric</i>
2	<i>DES ADD</i>	<i>Continuous , Numeric</i>
3	<i>PKT ID</i>	<i>Continuous , Numeric</i>
4	<i>FROM NODE</i>	<i>Continuous , Numeric</i>
5	<i>TO NODE</i>	<i>Continuous , Numeric</i>
6	<i>PKT TYPE</i>	<i>Continuous , Nominal</i>
7	<i>PKT SIZE</i>	<i>Continuous , Numeric</i>
8	<i>FLAGS</i>	<i>Symbolic , Nominal</i>
9	<i>FID</i>	<i>Continuous , Numeric</i>
10	<i>SEQ NUMBER</i>	<i>Continuous , Numeric</i>
11	<i>NUMBER OF PKT</i>	<i>Continuous , Numeric</i>
12	<i>NUMBER OF BYTE</i>	<i>Continuous , Numeric</i>
13	<i>NODE NAME FROM</i>	<i>Symbolic , Nominal</i>
14	<i>NODE NAME TO</i>	<i>Symbolic , Nominal</i>

Table 4.6 (Continued): Alkassasbeh dataset features.

<i>Feature No</i>	<i>Feature Name</i>	<i>Type</i>
15	<i>PKT IN</i>	<i>Continuous , Numeric</i>
16	<i>PKTOUT</i>	<i>Continuous , Numeric</i>
17	<i>PKTR</i>	<i>Continuous , Numeric</i>
18	<i>PKT DELAY NODE</i>	<i>Continuous , Numeric</i>
19	<i>PKTRATE</i>	<i>Continuous , Numeric</i>
20	<i>BYTE RATE</i>	<i>Continuous , Numeric</i>
21	<i>PKT AVG SIZE</i>	<i>Continuous , Numeric</i>
22	<i>UTILIZATION</i>	<i>Continuous , Numeric</i>
23	<i>PKT DELAY</i>	<i>Continuous , Numeric</i>
24 \	<i>PKT SEND TIME</i>	<i>Continuous , Numeric</i>
25	<i>PKT RESEVED TIME</i>	<i>Continuous , Numeric</i>
26	<i>FIRST PKT SENT</i>	<i>Continuous , Numeric</i>
27	<i>LAST PKT RESEVED</i>	<i>Continuous , Numeric</i>

Features were extracted after doing some steps in the following:

1. Gathering and examination: In this step, all traffic was collected and checked from NIDS.
2. Data preprocessing: Excessive and duplicate records are removed.
3. Extraction of Feature: Parameters were extracted from the traffic of the collected traffic network, and each feature was allocated to the first column and then used as a vector in the new dataset.
4. Statistical calculations: using statistical equations to extract the rest of the features.

4.5 Summary

In this chapter, we have reviewed a systematic approach to the dataset of IDS/DDoS using both types of dataset regarding the number of attacks, features and the number of records within each dataset as well as how and generated. These datasets are employed to evaluate the performance of the advanced methods described in the following chapters.

CHAPTER FIVE

DESIGN AND METHODOLOGY

5.1 Introduction

DDoS attacks are problems since networks use and high-speed networks are increasing in everyday life. Therefore, it was urgent to find a way to operate efficiently to detect DDoS attacks. In this chapter, a workflow methodology implemented of the suggested a new model for feature selection and reduces dimension by using Mutual information and Singular Value Decomposition to select most relevant features, reduce processing time and data complexity of choosing the most powerful features of DDoS attack. We will describe the theoretical explanation of these techniques also the dataset that simulated the actual reciprocal network traffic. We use BPNN algorithm for detecting the DDoS attacks to diminish the false positive value. The relevant features of the NSL-KDD and Alkansasbeh datasets are selected to enhance the classification detection accuracy.

5.2 Proposed System Architecture

The purpose of the suggested method is to reduce feature sets considerably while maintaining and improving classification accuracy by using back propagation neural networks. In this chapter, using combine the method of feature selection and dimension reduction called (FSDR) for selecting essential features that influence in DDoS attacks. In literature studies, it has been found that most feature selection and dimensionality reduction methods are biased towards considering attributes with many different values. Despite, attributes with quite small information value to seem to receive unfair choice [69]. We use BPNN algorithm to evaluate the approach for detecting the DDoS attack to diminish the false positive value. The proposed method includes three stages. Figure (5.1) shows the entire system flow of the proposed model.

Proposed Framework

1. Phase 1

- a) (Data Collection and Pre-processing)
- b) Selecting NSL-KDD dataset or Alkansasbeh dataset.
- c) Dataset Preparation, Labeling Dataset, and normalization.

2. Phase2

(Our algorithm FSDR) Feature selection

- a) Apply the mutual information for feature scoring.
- b) Apply Normalization on new feature scoring by empirical cumulative distribution functions to expand variance of feature scoring.
- c) Determine threshold depend on uncertainty value to select the subset of features.

3. Phase 3

Dimension reduction

- a) Apply developed SVD with new features space.
- b) Apply PCA for comparison
- c) Apply SVD for comparison
- d) Apply datasets without using dimension reduction methods.

4. Phase 4

(Verification of DDoS attacks Detection)

- a) Training and testing by using NNs Using the Back-propagation Algorithm.
- b) Evaluating results of both datasets

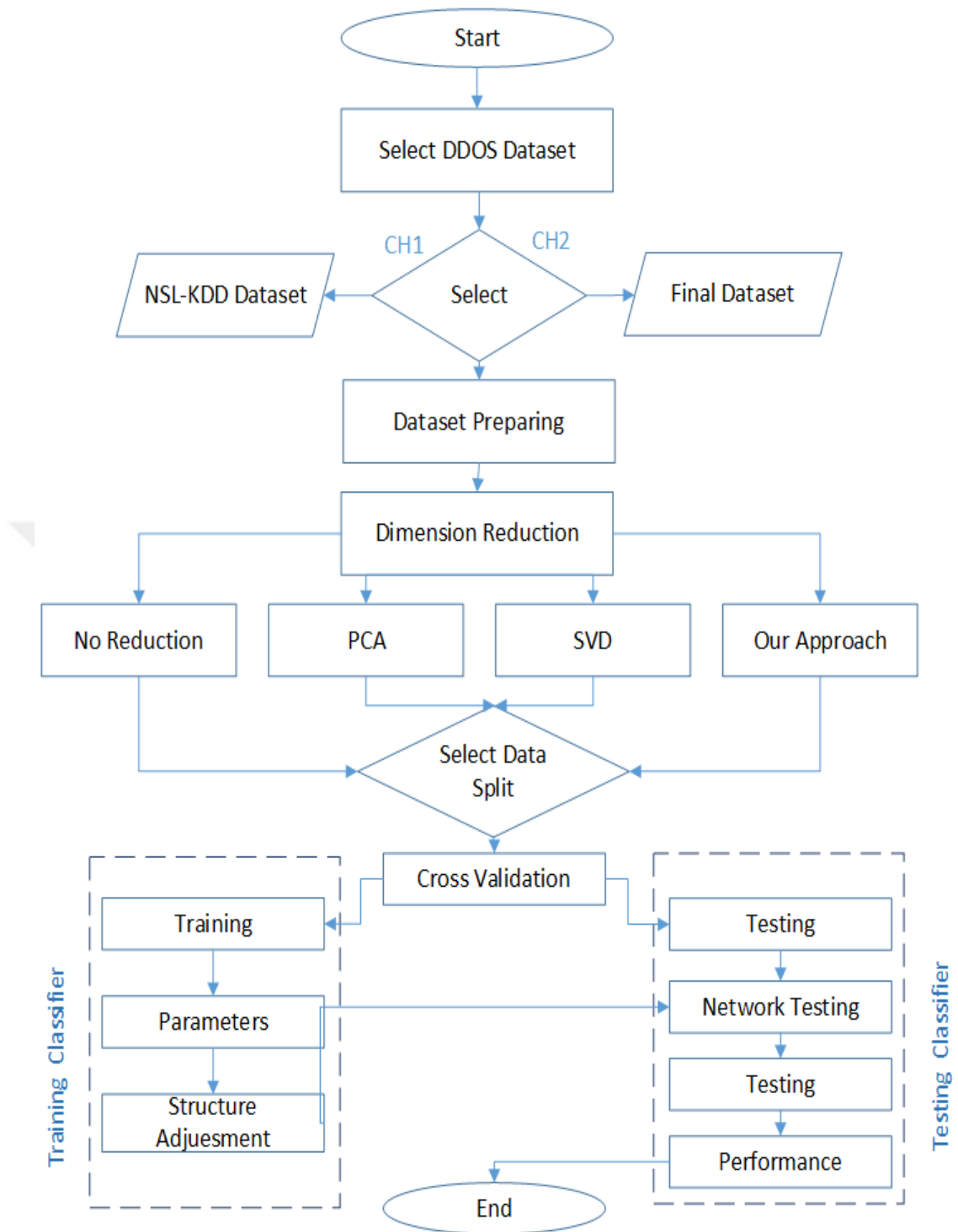


Figure 5.1: The component architecture of the proposed work.

5.2.1. Preprocessing and Labeling Dataset Stage

5.2.1.1 The datasets

In this search, we used two types of the dataset. Firstly, NSL-KDD benchmark dataset. The focus of this thesis on DOS attacks just because it occupies the most

significant proportion of the attack and the most common where the total percentage was about 78.3%. DOS attack to the rest of the attacks occurs in a network of six kinds of attacks. The four most important types of DoS attacks were selected out of six as Neptune, Teardrop, Smurf and Back as well as normal were extracted from the complete NSL KDD dataset. The dataset contains 41 features and the selected part of the dataset consists of a training set containing 113051 records and a test set of 15404 documents. First dataset NLS-KDD comprises four files. It is detailed in chapter IV.

Secondly, the new dataset has been established by Alkasasbeh et al. [1]. They provided a set of data containing modern types of an attack targeting different OSI layers such as Application layer and Network layer. It is the presence of modern types of DDoS attacks as HTTP Flood, SIDDoS, UDP Flood, and Smurf. This set of data contains four types of attack in addition to normal traffic and also comprises 27 features. The dataset includes 2160668 records of the normal and attack samples. The details of the dataset are mentioned in chapter 4. We randomly selected 250000 records to test them in our model between normal traffic and four kinds of attack.

5.2.1.2 Preprocessing and labelling dataset

Data preprocessing comprises of a sequence of steps to convert raw data acquired from data extraction into a clean and neat dataset before statistical analysis. Preprocessing is aimed at evaluating and enhancing the quality of data to enable regular statistical analysis. It includes preprocessing data such as cleaning, integration, conversion, and reduction data [70]. Preprocessing stage, the size of the data indicates that needs to reduce the size and preprocess data to change the data laboring with the selected classification algorithm. The NSL-KDD and Alkasasbeh datasets as input datasets include several features. However, they are in different formats. Some are in the numeric format. Others are in nominal format. Therefore, datasets of these various formats must turn into similar formats and extract from the next stage. The types of nominal features were mentioned in the previous chapter. It will be counted in sequence with the two original datasets. They conduct the conversion processes the following steps:

1. Input NSL-KDD dataset /Alkasasbeh dataset
2. Calculate the probability of (protocol, service, flags, node name from and name to.)

3. Replace nominal value of numeric
4. Generate a numeric CSV file
5. Read a CSV file and normalize use Min-Max normalization
6. Generate a normalized Mat and CSV file.
7. Output NSL-KDD Dataset /Alkasasbeh dataset.

We also convert label classes into numeric to be handled in MATLAB files as either normal or an attack with one definite attack type as in the following Tables (5.1).

Table 5.1: Class labeling of NSL-KDD dataset.

<i>Type</i>	<i>Discerption</i>	<i>Label</i>
<i>Normal</i>	<i>data with no attack</i>	<i>1</i>
<i>Neptune</i>	<i>(DoS)</i>	<i>2</i>
<i>teardrop</i>	<i>(DoS)</i>	<i>3</i>
<i>Smurf</i>	<i>(DoS)</i>	<i>4</i>
<i>Back</i>	<i>(DoS)</i>	<i>5</i>

Table 5.2: Class labeling of Alkasasbeh dataset.

<i>Type</i>	<i>Discerption</i>	<i>Label</i>
<i>Normal</i>	<i>data with no attack</i>	<i>1</i>
<i>UPD-Flood</i>	<i>(DDoS)</i>	<i>2</i>
<i>Smurf</i>	<i>(DoS)</i>	<i>3</i>
<i>SIDDoS</i>	<i>(DDoS)</i>	<i>4</i>
<i>HTTP-Flood</i>	<i>(DDoS)</i>	<i>5</i>

5.2.1.3 Normalization

The third stage of data processing, it is performed by a preprocessing module and data for a ready machine learning. Normalization makes it easy to compare different features with a range specified on a different scale, such as (-1 to 1) or (0 to 1). Also, normalization can accelerate the time needed to train a neural network [71]. We used Normalization in three stages of work. First, the process of converting the nominal features into numeric, we used the min-max equation. Second, in the stage of mutual information for feature scoring using the z-score. Finally, in the pre-classification phase using the data also using Z-score equation. The method uses Min-

Max normalization strategy between 0 and 1 to normalize. It assigns the following equation:

$$\hat{v} = \frac{v - \min A}{\max A - \min A} \quad (5.1)$$

Here A is a related feature. V is a possible value of A within the current time window and V' new is the normalized value. The module chooses appropriate features concerning DDoS detection from Alkansasbeh dataset, or NSL-KDD datasets using the method detailed in IV-B.

In normalization of Z-score, the values of A and A are normalized according to the value of the mean, and standard deviation. Calculation normalizes the value A with v [72].

$$v' = ((v - \bar{A}) / \vartheta_A) \quad (5.2)$$

Here \bar{A} and ϑ_A are the mean and standard deviation, respectively. Each feature produces data with zero mean and unit variance. The standard deviation can be calculated using the following equation:

$$\vartheta_A = \sqrt{\frac{1}{n} \sum_{i=1}^n (A_i - \bar{A})^2} \quad (5.3)$$

The mean can be calculated using the following equation:

$$\bar{A} = \frac{1}{n} \sum_{i=1}^n A_i \quad (5.4)$$

Here, n is the total number of data point in the entire dataset. This normalization method is beneficial when the original minimum and maximum values of A are unknown. Normalization can convert the original data and normalization parameters (mean and standard deviation when z-score normalization is used, minimum and maximum values when min-max normalization is used, normalized in the same way).

The second phase is the dimensional feature reduction and feature selection, which in this step we will propose a development approach for Guide Feature Selection and Dimensionality Reduction (FSDR) approach which combines both steps in the same method. In this stage, we will try to select suitable features from the dataset by using Mutual Information scoring features and then using empirical cumulative distribution function to determine uncertainty threshold value to select more relevant features. Then, it reduces the dataset dimensions as well. We will try to improve and develop the (SVD) function. In the term to satisfy suitable features which are reduced

and extracted as per dataset. This step reduces the dimensionality of the dataset and removed features as well then are given as an input to a next step.

5.2.2 Phase Two: Feature Selection

5.2.2.1 Mutual information (MI)

Mutual information MI is an essential connotation in information theory. Also, it is broadly employed in artificial intelligence (AI). MI indicates the presence of the information shared among two variables and measuring the extent to which one variable is known will be reducing the value of uncertainty for the other. We applied mutual information (MI) for weight the whole feature space. High MI means that the uncertainty is mostly reduced. Low MI means a small reduced in uncertainty, zeroing mutual information among two random variables intends that the variables not be dependent. This relationship is not necessarily linear. The way it works depends on if there is much information shared among the feature and class of labeled data then this is a strong signal that this feature is significant and beneficial in identifying class members from other. Its entropy adopts the idea of the uncertainty concerning random variables. The concept of entropy included discrete variables and continuous variables [73].

If X is a random variable with discrete values, its entropy is expressed as

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \quad (5.5)$$

Where $H()$ is entropy, and $p(x) = \Pr(X = x)$ is the probability density function of X . The key of entropy relies on the distribution of the probability of the random variable.

If conditional entropy is described as the uncertainty decrease in one variable while the other is identified, suppose variable Y is provided, the conditional entropy $H(X; Y)$ of X concerning Y is

$$H(X|Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log p(x, y) \quad (5.6)$$

Where $p(x, y)$ is the joint probability density function and $p(x; y)$ is the following probabilities of X given Further, the joint entropy $H(X, Y)$ of X and Y is

$$H(X, Y) = H(X) + H(Y|X)$$

$$\begin{aligned}
&= H(Y) + H(X|Y) \\
&= \sum_{y \in Y} \sum_{x \in X} p(x, y) \log p(x, y) \quad (5.7)
\end{aligned}$$

The mutual information for discrete random variables can be defined mathematically as:

$$\begin{aligned}
I(X; Y) &= H(X) - H(X|Y) \\
&= H(Y) - H(Y|X) \\
I(X; Y) &= \sum \sum_{xy} p(x, y) \log \frac{p(x, y)}{p(x) p(y)} \quad (5.8)
\end{aligned}$$

Where $p(x, y)$ is the joint probability distribution function of X and Y and $p(x)$ and $p(y)$ are the marginal probability distribution functions for X and Y respectively.

The mutual information for continuous random variables can be defined just changing summation by a definite double integral as:

$$I(X; Y) = \iint_{x, y} p(x, y) \log \frac{p(x, y)}{p(x) p(y)} dx dy \quad (5.9)$$

Here, $P(x, y)$ is the joint probability density function of X and Y, and $P(x)$ and $P(y)$ are the marginal probability density functions of X and Y sequential. In this type of filter, there is a problem the variance of the experimental mutual information in the sample. The existence of the problem may help to let for the issuance of relevant wrong judgments. The features are determined next keeping those. The information overrides a defined threshold. It will be ignored. To strengthen the selection, we should not forget to have some assurance the actual value of mutual information.

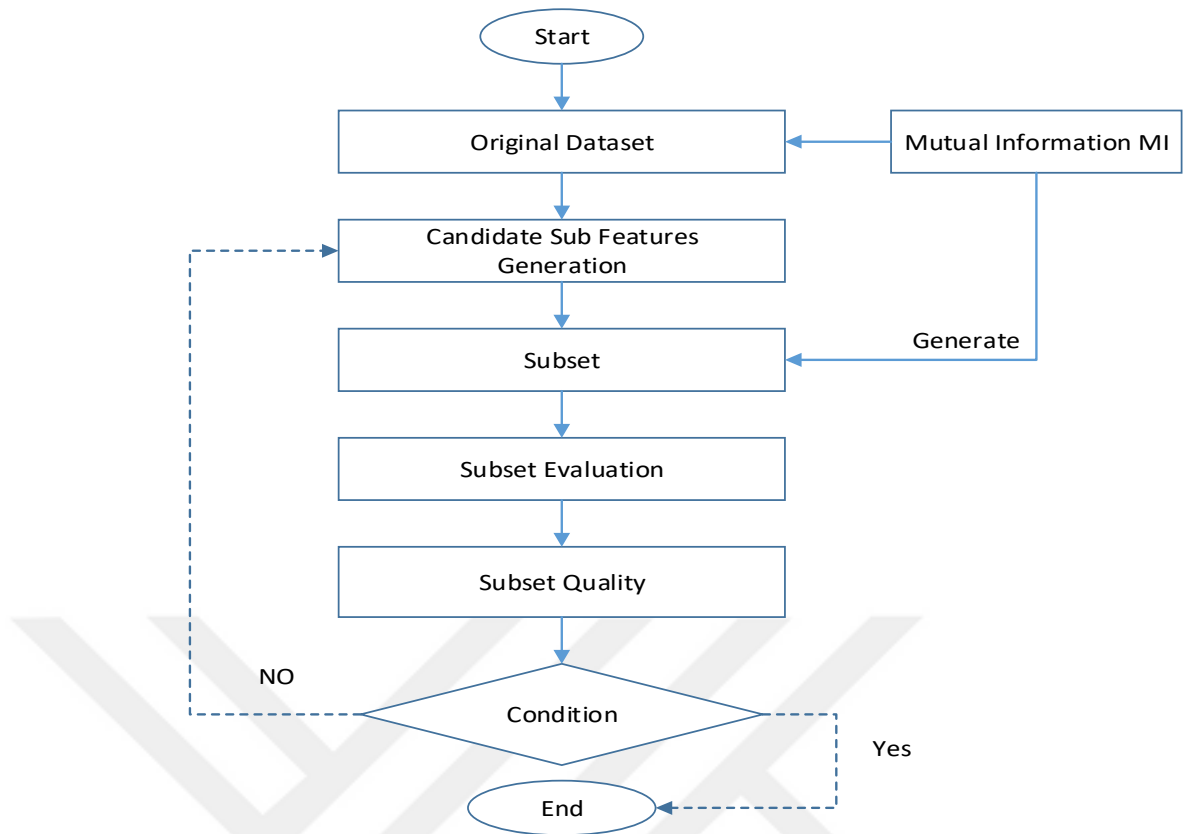


Figure 5.2: Mutual information with feature selection method.

5.2.2.2 Empirical cumulative distribution functions

In statistical methods, empirical distribution function indicates the distribution function correlated by empirical measurement from the sample. The Empirical Cumulative Distribution Function ECDF is a valuable assessment of the cumulative distribution function CDF to a random variable. We use ECDF to expand weights variance of features by normalizing it that have obtained from MI and determent threshold of uncertainty value. That formed the points in the sample. The empirical CDF is calculated by ranking each sample (separately for each variable) and then rescaling the integer rank values to the range (0, 1) .The ECDF is nearly related to the cumulative frequency, and it is determined by getting no assumptions about the underlying distribution. It allocates a probability of $1/n$ to each point of data n . The order arranges the ascending order of the values, and the sum of the allocated probabilities is calculated up to including each observation in a sample. As a result, a

step function increases by one / n for each data is obtained. The ECDF is normally expressed[61] by $\hat{F}_n(x)$ or $\hat{P}_n(X \leq x)$, and is defined as.

$$\hat{F}_n(x) = \hat{P}_n(X \leq x) = n^{-1} \sum_{i=1}^n I(x_i \leq x) \quad (5.10)$$

In fact, to compute the x value this is done in two steps the first step counting the number of data points less than or equal x. The next step is dividing the number of product in the first step by the total number of sample. Remarkable interest facts concerning ECDF. The anticipated value of the ECDF is the actual underlying CDF. This means the ECDF is an unbiased estimator of the CDF of the observations. The ECDF is a constant estimator for the correct CDF at of x value.

The ECDF is beneficial due to resemble an actual CDF well when the size of a sample is large and knows that distribution is useful for statistical reasoning. It can see how fast CDF increases to one Plotting the principal quantiles such as quartiles is beneficial for obtaining the "feel" of the data. The plot of ECDF is a frequently used known CDF distribution when comparing to CDF. It ascertains whether data acquired from one of their common distributions.

5.2.3 Phase 3: Dimension Reduction

5.2.3.1 Enhanced SVD

This part is considered the last of the second stage and after completing the selection of subset features from the original datasets by using MI. We will use SVD's enhanced dimension reduction algorithm for the new set of features. Our model which depend on the regular SVD but our idea is to rank the Eigenvalue which is in diagonal and accelerated Singular Value Decomposition.

Our function produces a diagonal matrix so of the dimension of the rank of X (whole data dimension S) and with non-negativity diagonal elements in decreasing order, and unitary matrices U and V eigenvalue. Where U is eigenvector and V is an eigenvalue.

The basic idea of this development process depends on the method of evaluating the dimensions of the total data in the SVD algorithm. Two-way of the eigenvalue values are arranged from the largest to the smallest and vice versa to reduce the dimensions used to reconstruct the major factors of the variation of the new sub-feature data at ration 96%. The next steps, we will review the work of SVD as following

(A) Produce a diagonal matrix S of a dimension of X means the whole dimension of the dataset.

(B) Do the rank X of the whole of dimension dataset depends on S matrix which is the diagonal matrix

(C) Depend on the non-negativity value which is in a decreasing order of the S matrix. The result is the unitary matrix which is U is eigenvalues of matrix S , and V is eigenvectors of matrix S .

(D) Checking the best approximation that respects the normalization data among all matrices with the Rank not longer than reduced.

Finally in step

(F) Based on the size of the X which is the whole data dimensions firstly, compute the eigenvector $X^T.X$ or $X.X^T$ and secondly convert them to eigenvectors. The suggested enhanced algorithm steps are described in algorithm

Algorithm Steps

In first case my data in $[X^T.X]$

Step (1): determine the Max matrix size .and the eigenvector ratio =0.1

Step (2): compute the S matrix \leftarrow compute (Data $\leftarrow [X^T.X]$)

Step 2.1: check the optimally reduced dimension

Step 2.1.1: compute the diagonal data matrix

Step 2.1.2: Find the max value of the data $X \leftarrow \max$

Step 2.1.3: Find the data size

Step 2.1.4: check the reduced dimension

- a) Compute the eigenvalue
- b) Compute the diagonal of eigenvalue
- c) Sort the value of eigenvalue
- d) Get the index of eigenvalue
- e) Get $[U]$ of eigenvalue matrix

Step 2.1.5: select the max of the eigenvalue

Step 2.1.6: get the index of the eigenvalue

- a) Replace the eigenvalue with the index

Step 2.1.7: compute the unitary of the matrices U

Step 2.1.8: produce diagonal matrices of the dimension of the rank X of the nonnegative diagonal element in declining order S .

- b) Find the minimum half of the eigenvalue
 - c) Compute the unitary matrices V Eigenvector
- Else Step (3): if $\frac{\# \text{ of the feature}}{\# \text{ of the number of the sample}} < 1.0631$
- Do the same steps from (Step 2.1.1 to step 2.1.8).

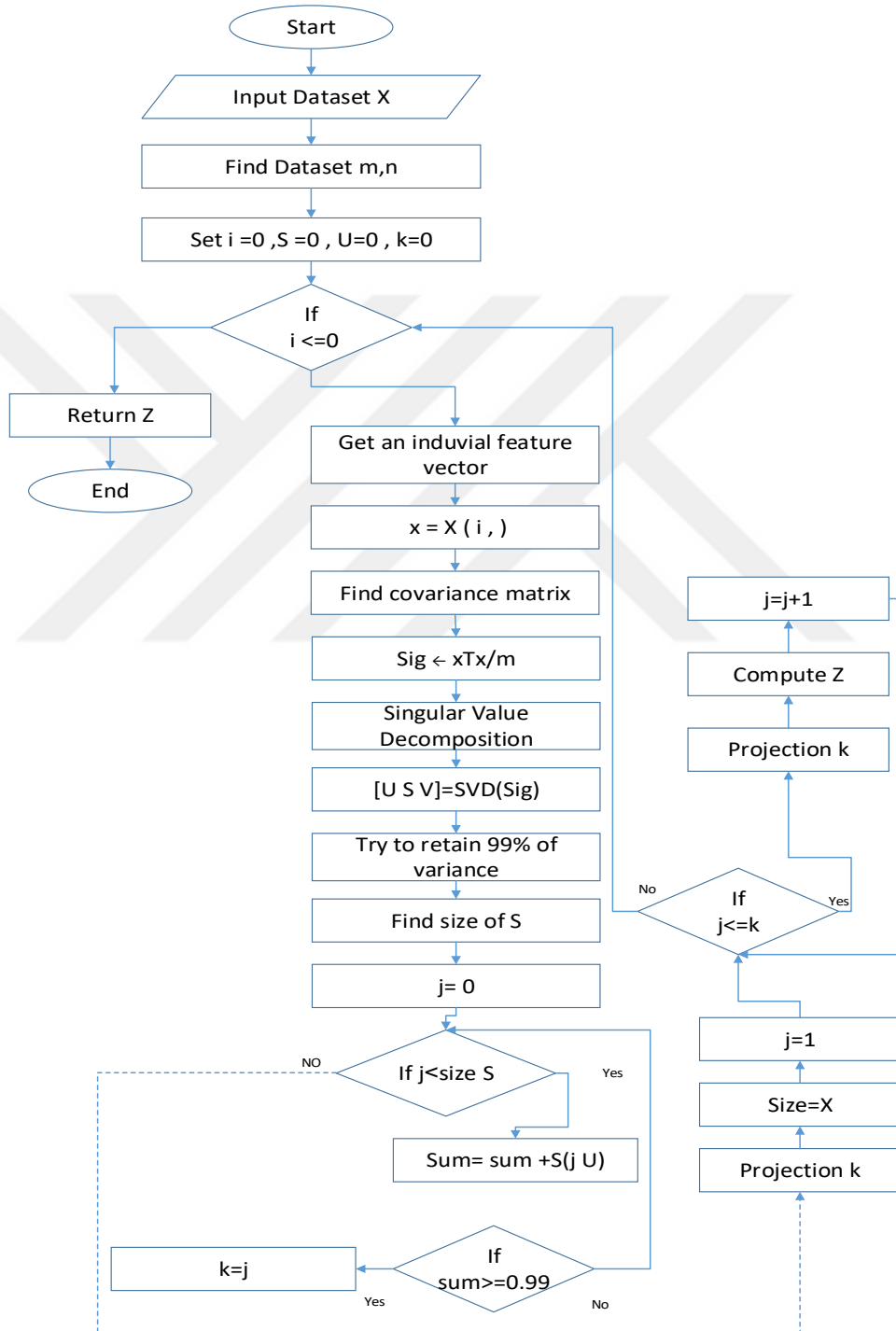


Figure 5.3: Structure of the enhanced SVD algorithm.

5.2.3.2 Singular value decomposition (SVD)

SVD is considered to be one of the most important and common methods factorized matrix. We use the SVD algorithm for comparison with our model. SVD provides an accurate description of any matrix, and As well as facilitate the removal of the insignificant parts of this representation to provide an approximate representation of any number of respectable dimensions.

$$A = U\Sigma V^T \quad (5.11)$$

For any n and m, consider a matrix $A \in \mathbb{R}^{n \times m}$. Let $r = \text{rank}(A)$. The matrices $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{m \times r}$ each of the matrices have orthonormal columns, left and right singular vectors of A sequentially. The U column forms the orthonormal basis of the column span of A and V row forms the orthonormal basis of the row span of A. Here, $\Sigma \in \mathbb{R}^{r \times r}$ is diagonal matrix nonnegative ordinal elements arranged in descending order along the diagonal of the upper /left square block are diagonal matrices with singular values $\sigma_1 \geq \sigma_2 \geq \dots \dots \geq \sigma_r \geq 0$ of A. In the case of A, if $r < \min(m, n)$. That is matrix A is not full rank, and only r singular values are greater than 0. The full rank decomposition of A is generally expressed as follows $A = U_r \Sigma_r V_r^T$. Singular values are always real. When the values of matrix A are real numbers, the values of the matrix U, V are also real numbers. The first k maximum singular values are equal to 0 and only the first k columns of U and V are used. It is generally expressed such as

$$A_k = U_k \Sigma_k V_k^T \quad (5.12)$$

Here, $(u_1, \dots \dots, u_k)$ is a column vector and $((v_1^T, \dots \dots, v_k^T))$ is row of V^T It has relationship SVD between PCA. The computation of SVD is the same as finding eigenvalues and eigenvectors of AA^T and $A^T A$. The eigenvectors of AA^T constitute a column of U, and the eigenvectors of $A^T A$ constitute a column of V. Furthermore, the singular value in is the square root of the eigenvalue from AA^T and $A^T A$. The SVD link to PCA comes from direct linear algebra computation. When the value of each attribute of A is changed so the rate of each feature is equal to zero, then $\text{Cov}(A) = A^T A$. When rewriting the equation mentioned above in the form of $AA^T = Q\Lambda Q^{-1}$. The left part of equation was replaced by the right, $Q\Lambda Q^{-1} = V\Sigma^2 V^T$ Because Q and V are orthonormal, $Q^{-1} = Q^T, V^{-1} = V^T$. Whether the columns of R and Q are arranged so

that the eigenvalues of (R) are in descending order, you can see that the square of Q is the same as V. PCA deletes the average of each variable, but SVD uses the original data (although you can also delete the average before calculating SVD). For sparse data, in particular, it is not always desirable to delete the average of data, several studies have addressed the similarity and difference between PCA and SVD, for instance a good comparative explanation between the two methods

$$AA^T = (U\Sigma V^T)(V\Sigma^T V^T) = U\Sigma^2 U^T \text{ And} \quad (5.13)$$

$$A^T A = (V\Sigma^T U^T)(U\Sigma V^T) = V\Sigma^2 V^T \quad (5.14)$$

It is important understanding why (SVD) is usually indicated to as (PCA). The columns of U and V are recognized as the left and right PC of A. v_1 , the first column of V is A's top right SV and gives a top PC, that represents the direction of considerable variance within A. The i^{th} SV v_1 gives the i^{th} PC which is the direction of greater variance orthogonal to all higher principal components formally:

$$\| Av_i \|_2^2 = v_i^T A^T A v_i = \sigma_i^2 = \max_{\substack{x: \|x\|_2=1 \\ x \perp v_j \forall j < i}} x^T A^T A x \quad (5.15)$$

Where $A^T A$ is the covariance matrix of A Similarly, for the left singular vectors we have:

$$\| u_i^T A \|_2^2 = u_i^T A^T A u_i = \sigma_i^2 = \max_{\substack{x: \|x\|_2=1 \\ x \perp u_j \forall j < i}} x^T A^T A x, \quad (5.16)$$

We used the SVD reduction algorithm for comparison with our model. Here are the steps of the SVD algorithm:

Algorithm Singular Value Decomposition (SVD)

1. Input: Generate Data matrix X
2. Output: New Dimensions C
3. Applying SVD to the matrix X as $X = USV^T$
4. $X \rightarrow$ is a $m \times n$ Matrix
5. $m \rightarrow$ No. of sessions (vectors)
6. $n \rightarrow$ No. of sessions (attributes)
7. $U \leftarrow XX^T$ Orthogonal Matrix of eigenvectors
8. S is a matrix which is diagonal
9. $V \leftarrow X^T X$ Orthogonal matrix of eigenvalues
10. Construct the covariance matrix from this the decomposition by
11. $XX^T XX^T \leftarrow (USV^T)(USV^T)^T = (USV^T)(VSU^T)$
12. $V \leftarrow$ Orthogonal matrix
13. $(V^T V = I), XX^T = US^2 U^T$
14. The square root of eigenvalues of XX^T are the singular value of X

5.2.3.3 PCA Algorithm

PCA is a one of the linear methods that reduce the dimensions of data by analyzing the variance between the factors. PCA investigates the correlation between a couple of attributes to determine that important one and leads to a linear drawing of data into a less dimensionality space so that the contrast of the data is magnified in the low dimensionality space. PCA builds the data correlation matrix and calculates the eigenvector corresponding to the largest eigenvalue component used to reconstruct the majority factors of the variation from the original data.

Suppose we have $m \times n$ is the dimension of the matrix. Which presented a group of observations x_1, x_2, \dots, x_m is $n \times 1$ vectors, wherever every observation is described through a vector of length N . Therefore, the dataset is offered by matrix. Equation (1).

$$X_{M \times N} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} = [x_1, x_2, \dots, x_m] \quad (5.17)$$

The required value determines the average value for every column. This is illustrated in Equation (2).

$$\bar{X} = \frac{1}{M} \sum_{i=1}^M x_i \quad (5.18)$$

The standard deviation is calculated by subtracting each value from the x_i values from the average of all the values per column. These can be calculated using the following equation.

$$\varphi = x_i - \bar{X} \quad (5.19)$$

Covariance and correlation: is a measure of the variance of the strength of two features to determine the percentage of the difference between them. The covariance can be calculated by taking a random sample of two of the variables x, y with the calculation of the average (\bar{x}, \bar{y}) of the variables, in addition to the sample size of m . The standard deviation is calculated by subtracting each value from the x_i values from the average of all the values per column. These can be calculated using the following equation. $\varphi = x_i - \bar{X}$. Covariance equation can be calculated as

$$Cov(x, y) = \frac{1}{m} \sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y}) \quad (5.20)$$

It is worth noting that if the two variables are equal, $x = y$ means the covariance between them when it applied standard deviations to variables x, y . The result of this is equal to the variance and correlation coefficient of x, y . This equivalence indicates the strength of the bond and the direction of the linear relationship between the variables x, y , and the correlation can be calculated using the following equation.

$$Corr(x, y) = \frac{Cov(x, y)}{\sigma_x \sigma_y} \quad (5.21)$$

The covariance matrix of A can be calculated when a matrix is given with a distance of $m * n$ where the m rows represent the data and n columns the attributes represent these data. That is a square matrix formed variations of a single. It can be calculated. If it changes the values of each attribute of A so that the average of each attribute as

$$Cov(A) = A^T A \quad (5.22)$$

Eigenvalues and eigenvectors. Let's $n * n$ is the dimension of the matrix C and a nonzero vector q , the values of λ completing the equation

$$Cq = \lambda q \quad (5.23)$$

The λ is named the eigenvalues of C, the vectors q completing equation above are named eigenvectors. The number of non-zero eigenvalues of C is at generality rank (C). It is the maximum of linear rows and columns linearity from C, respectively. Equation above-mentioned can also be written as

$$(C - \lambda I_n)Q = 0 \quad (5.24)$$

If C has an independent linear eigenvector q_1, q_2, \dots, q_n , the C would produce three matrices

$$C = Q\Lambda Q^{-1} \quad (5.25)$$

Where Λ is a diagonal matrix which diagonal input are the eigenvalues of C sorted by descending order $(\lambda_1, \dots, \lambda_n)$, and Q is an orthogonal matrix due to all the columns in Q, which will be referred to as $q_i = [q_{i1}, \dots, q_{in}]$ are orthogonal, further, a matrix of eigenvectors of C, the i th eigenvectors correspond the i^{th} larger eigenvalue. This equation is called the decomposition of eigenvalue of C. Furthermore, the diagonal of Λ describes the eigenvalues of the covariance matrix, and the columns of Q describe the respective eigenvectors of the covariance matrix, also known as the principal components of the data. When performing eigenvalue decomposition analysis on the square matrix of Cove (A) .After that, the raw data matrix A can be converted to a different matrix $\hat{A} = AQ$, with $Q_i = [q_1, \dots, q_n]$ (See also Equation 9). Every column of \hat{A} is a linear composite of the basic features, \hat{A} columns are the principal components, the new i^{th} feature differences is λ_i , the sum of the variances for all new features is equal to the sum of the variances in the original features. The eigenvalues also describe the variance of the data. For this reason, it is important to rank the eigenvalues and associated eigenvectors in descending order to maximize the variance in the sub dimensioned subspace. After the ranking step of the eigenvectors and eigenvalues. The PCA picks the first d principal components with $d \leq D$.

We used the PCA reduction algorithm for comparison with our model. Here are The steps of the PCA algorithm:

Algorithm Principle Component Analysis (PCA)

Input: Generate Data matrix X (feature of NSL-KDD or FINAL dataset)

Number of principle component d

Output: New Dimensions N

1. Repeat
2. Compute the mean of transactions $\mu \leftarrow \frac{1}{m} \sum_1^m x_i$
3. Subtract the mean from each transaction $X(t) \leftarrow x_i - \mu$
4. Compute the covariance matrix $co(t) \leftarrow \frac{1}{m} X_n X_n^T$
5. From $Co(t)$ compute eigenvectors $u(t)$ of AA^T
6. Consider matrix AA^T as a matrix $M \times M$ matrix
7. Compute the eigenvectors $v(t)$ of AA^T such that :
8. $AA^T \rightarrow \mu_i V_i$
9. $\mu_i V_i \rightarrow AA^T AV_i$
10. Compute the best μ eigenvectors of AA^T : $\mu_i \leftarrow AV_i$
11. Keep only K eigenvectors, (K features with their values)
12. $U \leftarrow Top(eigenvector(C, d)$
13. Until represent every transaction li over the time interval t as a vector $x(t)_i$
14. Return $N \leftarrow U^T X$

5.2.4 Phase Four Neural Network DDoS Detector

The ANN can distinguish current attack types and detect unknown attack types it is very useful to use artificial neural network BP for IDS. Traditional neural networks have some inherent flaws such as easy weaknesses. That breaks down to local minimums, so set up some networks New Neuro IDS for increases durability increasing intelligence and adaptation of IDS This is leading to inclination the evolution of IDS. This section introduces the adopted NN to classify DDoS attack types and normal in the NSL-KDD dataset and Alkasasbeh dataset. Moreover, we present here the expanding our NN and we also use Residual design techniques applied to NN to improve the DoS detection performances and consuming the time of the proposed model.

5.2.4.1 Back-propagation

Backpropagation algorithm method to enable the error derivative calculation between the actual output. The target is derived independently for all weights in the network in multi-layered Perceptron NNs. The backpropagation algorithm is utilized to calculate the necessary modifications after random selection of the network weights. The work of the algorithm can be divided into four stages. The stages include calculating the feed forward the output layer concerning Backpropagation, the hidden layer concerning backpropagation and update the weight. If the value of the error function is as follows. The algorithm is turned off if the error value is function has matured, or decrease is sufficiently [74]. The next step illustrates the mechanism of the ANN algorithm Here X is input dataset, Y is output products, and W is weight values sequentially. θ is a correction required only for hidden layer and output. As well as after each repetition are continuously updated. The e is an error gradient value, and p represents the number of iterations. σ represents error gradient values.

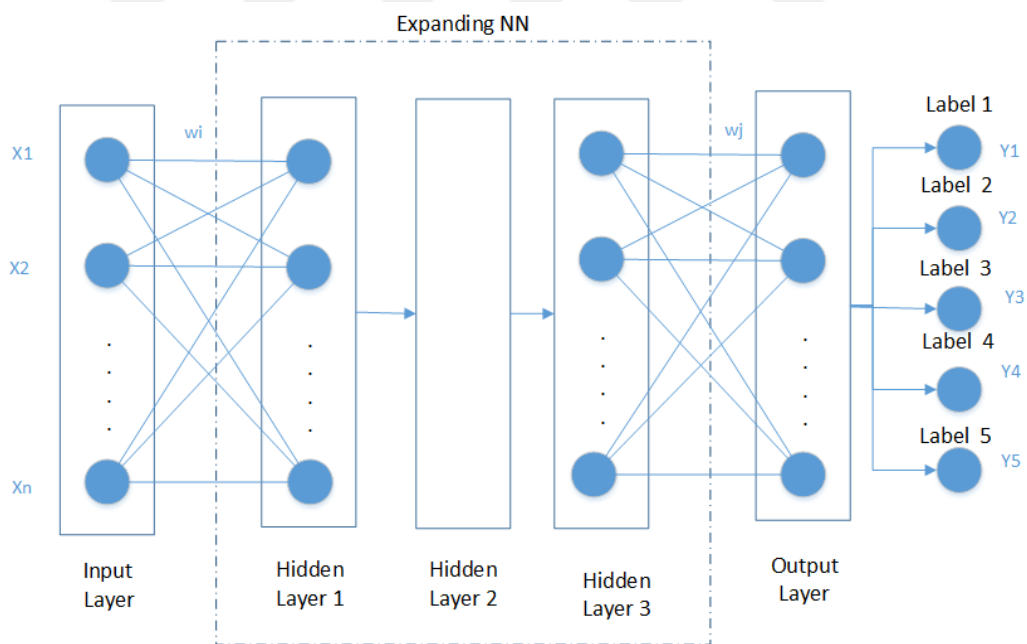


Figure 5.4: Structure of Back Propagation based on Neural Network with multi-layer

It ordinarily utilizes the sigmoid function and represented implying many phases of calculations as follow:

- 1) Determine all levels of weights and threshold levels of the neural network into randomized numbers distributed uniformly within a specified range such as

(-2.4/Fi, 2.4/Fi). Here Fi is the entire neuron (i) number of inputs in the neural network.

2) Compute the hidden layer output of neurons. Calculated using the following formula:

$$y_i(p) = \text{sigmoid} \sum_{i=1}^n [x_i(p) * w_{ij}(p) - \theta_j] \quad (5.26)$$

• Here n is the number of input j neurons in the hidden layer. Sigmoid activation function which is considered non-linear. It is a fit choice for simplifying functions in neural networks $\text{sigmoid}(s) = \frac{1}{1+e^{-s}}$. Where e is the basis of the natural logarithm.

• Compute the real outputs of the neurons at the output layer:

$$y_k(p) = \text{sigmoid} \sum_{i=1}^m [x_{jk}(p) * w_{jk}(p) - \theta_k] \quad (5.27)$$

Here m is the number of input from neuron k at the output layer.

• Compute the error gradient descent to the neurons at the output layer:

$$\sigma_k(p) = y_k(p) * [1 - y_k(p)] * e_k(p) \quad (5.28)$$

Here $e_k(p) = y_{d,k}(p) - y_k(p)$. They $y_{d,k}$ is the required output value.

• Compute weight after correction E

$$\Delta w_{jk}(p) = \alpha * y_i(p) * \sigma_k(p) \quad (5.29)$$

$$\text{After that update } w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p) \quad (5.30)$$

Here α is named rate of learning

• Compute the error gradient descent to the neurons at the output layer:

$$\sigma_j(p) = y_j(p) * [1 - y_j(p)] * \sum_{k=1}^l w_{jk} p * \sigma_k(p) \quad (5.31)$$

• Compute weight after corrections

$\Delta w_{ij}(p) = \alpha * x_i(p) * \sigma_j(p)$ After that update

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p) \quad (5.32)$$

• In this step, the iteration of p is increased, and the amount of the increment is equal to one. Then back to step two and repeat the process until the specified error amount is respect. The number of iteration 500.

The following Figure (5.5) shows the details of the NN in our model.

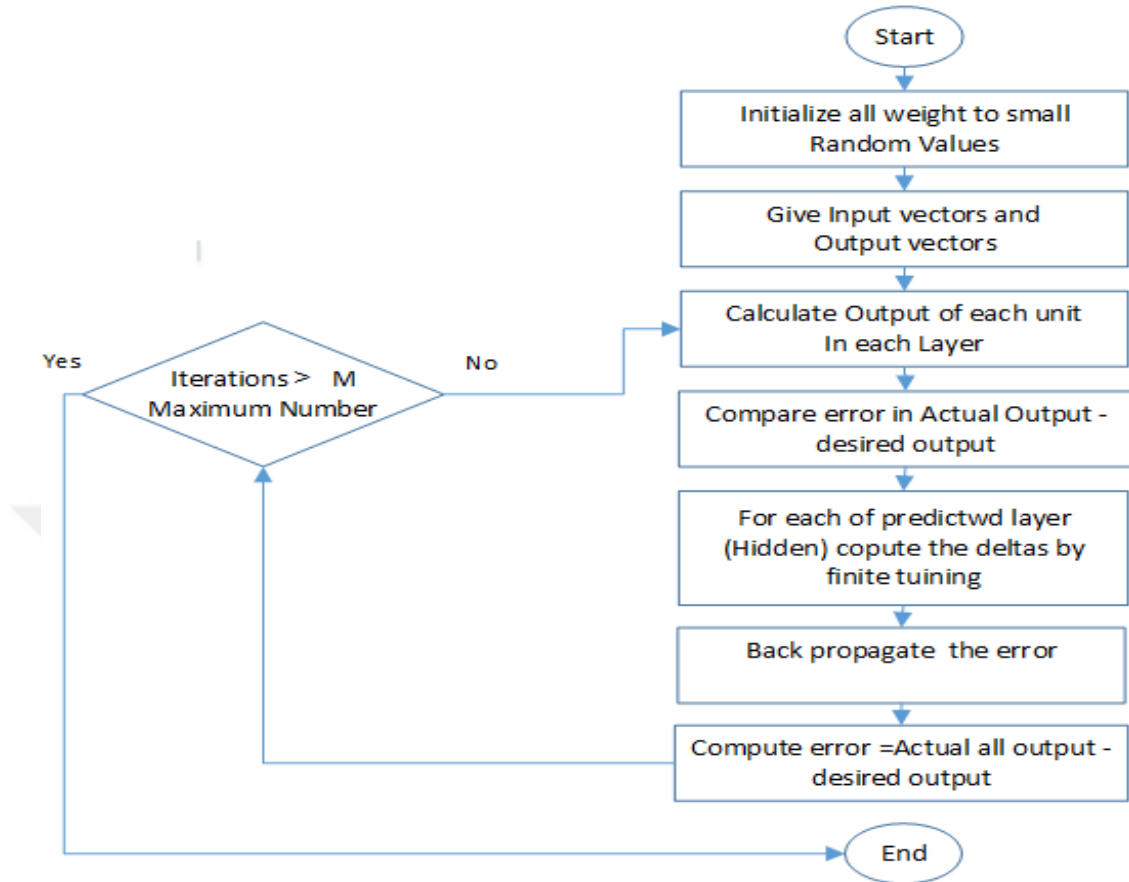


Figure 5.5: Details of the NN in our model.

Gradient descent method GDM is adopted to reduce the mean square error MSE connecting the output layer of the network and the rate of actual error Network efficiency is measured utilizing the coming parameters. *i* Convergence rate. *ii* the epochs number is taken to converge the network. *iii* Calculated MSE.

5.2.4.3 Residual network

Residual Network is a neural network structure that resolves t.he difficulty of fading gradients in a way that is as simple as possible. If it encounters problems sending the gradient signal in the reverse direction let's set a shortcut for each layer in the network and proceed smoothly .Whereby traditional networks the activation can be defined in a layer as follows :

$$y = f(x) \tag{5.33}$$

Here $f(x)$ is convolution matrix multiplication. That is batch normalization, and so on if the signal is sent in the opposite direction the gradient must always pass within $f(x)$ that can initiate problem because nonlinearity involved. Rather at each layer l performs a Residual Network the following

$$y = f(x) + x \quad (5.34)$$

The shortcut is to add x at the end y you can pass gradients directly backward. With stacking certain layers the gradient theoretically oversteps all intermediate layers and can reach the bottom without decreasing. The aim is to give a fine track for the gradient to backpropagate to the initial layer of the network. This affects the learning process accelerated by bypassing vanishing gradient and neuron death [75].

5.3 Training and Testing Stages

All stages of training and testing in classifier BP-ANN for both datasets NSL-KDD and Alkassasbeh were validated by five cross-validations to evaluate the model performance.

5.4 Cross-Validation

The aim cross-validation is to use whole available dataset to the learning stage. The data accomplish this cross-validation. The data is randomly split into K fold. As every fold, the whole dataset is employed however it is a learning set. It is employed as a test set. The mean error for each fold produces a small biased estimator [76]. We applied five-fold cross validation in this work by using MATLAB. The datasets are randomly split into five segments where the layer is represented in roughly the like proportions as in a whole set of data. Each segment is kept in turn, and the learning stage is trained on the remaining four /fifths next the error rate is computed on the standing segment. The learning phase was carried out a total of 5 times on separate various training sets, and subsequently, the average error rates of five resulted in a general error calculation. The consequence of the classifier BP-NN is the part that makes us understand whether our DDoS detection model is it operating correctly or not.

CHAPTER SIX

RESULTS AND DISCUSSION

6.1 Introduction

In this chapter, we will review the general performance results of the BPNN by using NSL-KDD and Alkawasbeh datasets based on the training and testing. We empirically evaluate Guided Features Selection Dimension Reduction (FSDR) by comparing it with some of the dimension reduction methods based on two datasets NSL-KDD and Alkawasbeh. We essentially observe the difference between the returned features subsets related to different dimension reduction methods. Hence, we respectfully present the results of NN-BP classifier, since the classification performance is the ultimate evaluation of a criterion. As one of the basic methods of the datasets of the proposed model in the selection of features and the reduction of dimensions using MI and developed SVD and their effect to detect DoS and DDoS attacks. We will review the results of four experiments using the SVD algorithm, the PCA algorithm, and the two sets of data without using the reduction methods. These algorithms are used to reduce the space feature of 41 and 27 and the features of both datasets and the classification of four types of DDoS attacks for both datasets. The results were done by using the MATLAB 2014a, program language to implement our proposed approach.

6.2 Environment

The experiments have been performed on a portable computer with i7-6700HQ 2.6 GHz CPU and 8 GB RAM. All phases of the proposed model and the algorithms used in the comparison have been implemented using MATLAB 2014a.

6.3 MATLAB

MATLAB[77] is a tool utilized for computation of numerical, programming and visualization. It can also be employed with multiprogramming languages like Python, C ++, and others. It also implements a less complicated program for creating models, data analysis, and algorithm development to perform classification and visualization missions. MATLAB has many easy tools in use that minimize the computational overhead time and improve operational effectiveness. We applied MATLAB2014a in all stages of our model of data preparation and algorithm classification using multi-class NN

6.3 Performance Measurements

There are several ways to measure the efficiency of the performance of classification algorithms, which will be reviewed in[78, 79] as follows:

1) True positives (TP) is intrusions true diagnosed (detected), true negatives (TN) means of non-intrusions correctly diagnosed (detected). A positive of true is a probability of issuing an alert when there is an intrusion. The true positive rate (TPR) is calculated using the following equation

$$TPR = \frac{TP}{TP + FN} \quad (6.1)$$

2) A false negative (FN) happens when the intrusions result is incorrectly prophesied for negative when it is positive (not detected)

$$FNR = 1 - TPR \quad (6.2)$$

3) A false positive (FP) happens when the non-intrusions result is incorrectly prophesied positive when it is negative. The false positive rate is measured as $FPR = 1 - TNR$ or it is calculated using the following equation as

$$FPR = \frac{FP}{FP+TN} \quad (6.3)$$

4) A Recall describes the percentage of the total all the samples in a database that required to be classified as positive denoted so and is computed as

$$TPR = \frac{TP}{TP+FN} \times 100\% \quad (6.4)$$

5) A Precision measures of the rate how many of the returned correct classifying in the classifier, among all the detection. They are defined by:

$$TPR = \frac{TP}{TP+FP} \times 100\% \quad (6.5)$$

6) The F-measure is described as the balance between the average of the precision and the recall, F-Score is a scale of the test accuracy, and the value of F increases in proportion to the improvement in efficiency, Recall, the higher value of F-Score denotes that the model works better in positive class. It is calculated as

$$F = \frac{2 \times \text{precision} \times \text{Recall}}{\text{precision} + \text{Recall}} \quad (6.6)$$

7) Accuracy, the most popular metric for classifier evaluation, it evaluates the overall leverage of the algorithm by assessing the probability success rate. It is the number of true classifications divided by the total number of classifications, it is calculated as

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6.7)$$

Furthermore, the error rate is an assessment of the probability of data misclassification as stated by the model prediction, it is calculated as

$$\text{The error rate} = 1 - \text{Accuracy}. \quad (6.8)$$

6.4 Confusion Matrix

Confusion matrix is an evaluation method applied to all types of classification problems. The size of matrix relies on the product of the diverse classes being classified. It is used to visually distinguish the information at actual class labels with forecasted class labels of the classifier. The confusion matrix can be illustrated in the following Table (6.1).

Table 6.1: Instance of Confusion matrix.

		Actual Class	
		Anomaly	Normal
Predicted Classless (Expectation)	Anomaly	TP Correctly Classifieds Anomaly	FP Incorrectly Classifieds Anomaly
	Normal	FN Incorrectly Classifieds Normal	TN Correctly Classified as Normal

Whole various performance evaluation measures are obtained from TP, FN, FP, and TN.

1. True Positive (TP): classified data will be positive classes are correctly classified Positive class.
2. False positives (FP): classified data will be negative classes are incorrectly classified positive category.
3. False negatives (FN): classified data will be positive classes are misclassified in a negative class.
4. True Negative (TN): classified data will be negative classes are correctly classified negative class.

The purpose concerning the detection algorithms is to diminish the FP and FN due to denote errors and also data erroneously classified. In the opposite direction must escalate TPs and TNs. As some attacks, the detection algorithm performs well, and some of them do not operate properly to detect the various types of attacks the algorithm must adjust according to that particular kind for improved performance to detect the attack.

6.5 Experiment and Results

All experiments were carried out and coding using MATLAB. We show the overall performances results in Accuracy, Recall, Precision and the F-Measure ratio of the BPNN classifier. Two different datasets NSL-KDD dataset and Alkansasbeh dataset were training and testing on the classifier. We named datasets on tables and figures about NSL-KDD dataset Data1 and Alkansasbeh dataset Data2. Four cases were used as the training, and testing dataset which are non-reducing dimensions, PCA, SVD and our model were suggested. We have categorized these datasets into two sections binary class and multiclass. We stated the building model time to BPNN classifier for each case.

6.5.1 Experimental One

In the first case, we use our model to select features and reduce dimensions with BPNN classifier to detect DDoS attacks in NSL-KDD dataset. We display the results of our model where the mutual information scoring is used to do a descending order

of 41 features according to the values of the information exchanged and selection 17 of the best features using the empirical CDF estimator as shown in Figure (6.1) below. The Table (6.2) below shows the order and sequence of features in the original dataset.

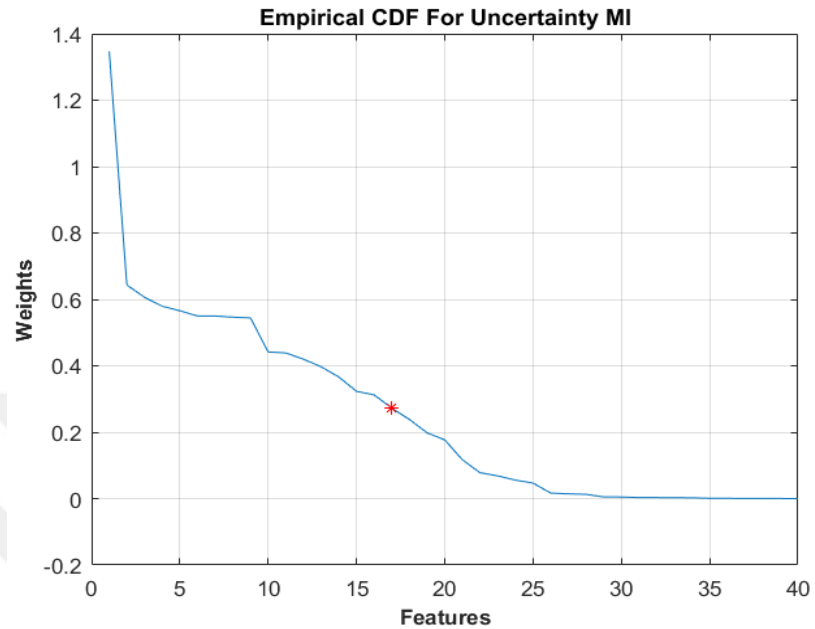


Figure 6.1: Empirical CDF for uncertainty MI value with Data 1.

Table 6.2: Numbers of features selected by MI with Data 1.

<i>NO</i>	<i>Feature Name</i>	<i>NO in Original dataset</i>
<i>1</i>	<i>dst_host_srv_error_rate</i>	<i>39</i>
<i>2</i>	<i>error_rate</i>	<i>27</i>
<i>3</i>	<i>dst_host_count</i>	<i>32</i>
<i>4</i>	<i>service</i>	<i>3</i>
<i>5</i>	<i>diff_srv_rate</i>	<i>30</i>
<i>6</i>	<i>num_outbound_cmds</i>	<i>20</i>
<i>7</i>	<i>srv_error_rate</i>	<i>26</i>
<i>8</i>	<i>protocol_type</i>	<i>2</i>
<i>9</i>	<i>srv_diff_host_rate</i>	<i>31</i>
<i>10</i>	<i>urgent</i>	<i>9</i>
<i>11</i>	<i>count</i>	<i>23</i>
<i>12</i>	<i>is_guest_login</i>	<i>22</i>
<i>13</i>	<i>dst_host_diff_srv_rate</i>	<i>35</i>
<i>14</i>	<i>dst_host_same_src_port_rate</i>	<i>36</i>
<i>15</i>	<i>dst_host_srv_count</i>	<i>33</i>
<i>16</i>	<i>dst_host_same_srv_rate</i>	<i>34</i>
<i>17</i>	<i>dst_host_error_rate</i>	<i>40</i>

The new features space data that was selected in the previous stage that used in the enhanced SVD algorithm for dimension reduction in the multi selective eigenvalue which influenced the reconstruction value as 96 %. In this process distinguishes a fitting low-dimensional description of original data for four best potential features which are listed in the Table (6.3) below.

Table 6.3: Names of reduced features at the developed SVD with Data1.

<i>NO</i>	<i>Name of Features Selected</i>	<i>No of Feature in Origin dataset</i>	<i>score</i>
<i>1</i>	<i>dst_host_srv_error_rate</i>	<i>39</i>	<i>1.347300</i>
<i>2</i>	<i>error_rate</i>	<i>27</i>	<i>0.643267</i>
<i>3</i>	<i>dst_host_count</i>	<i>32</i>	<i>0.606313</i>
<i>4</i>	<i>service</i>	<i>3</i>	<i>0.579425</i>
<i>Time for MI and DR /sec</i>		<i>2.77</i>	

Table (6.4) and Table (6.5) present all the performance metrics of our model on NSL-KDD dataset after dimensionality reduction. The performance metrics of two different tests on the same dataset in multiclass (Normal, Neptune, Teardrop, Smurf, and Back) and binary class (Normal, abnormal) are obtained. It is obvious that the average accuracy of the classifier in multi-class is 99.10% and in binary class 97.92% respectively. It is possible to note that the consuming time in the data testing using BPNN classifier was 29.76 sec.

Table 6.4: Performance measurement of our model with multi-class Data 1.

	Recall	Precision	F-M	FA	Accuracy
Normal	96.01	99.83	97.88	3.91	95.94
Neptune	100.00	96.24	98.09	0.00	100.00
Teardrop	99.97	99.76	99.86	0.04	99.96
Smurf	100.00	99.95	99.97	0.00	100.00
Back	99.65	99.98	99.81	0.37	99.63
Average	99.12	99.15	99.12	0.86	99.10
Testing Time Elapsed in NN			29.76 seconds		
Training Time Elapsed in NN			324.65 seconds		

Table 6.5: Performance measurement of our model with binary class Data 1.

	Recall	Precision	F-M	FA	Accuracy
Normal	96.01	99.83	97.88	3.91	95.94
Up normal	99.90	98.98	99.43	0.10	99.90
Average	97.95	99.41	98.66	2.00	97.92

Figure (6.2) shows on the left the error curve of training, either on the right showing the error curve of testing result with respect the iteration number rely on our model in FSDR by using BPNN classifier with four features.

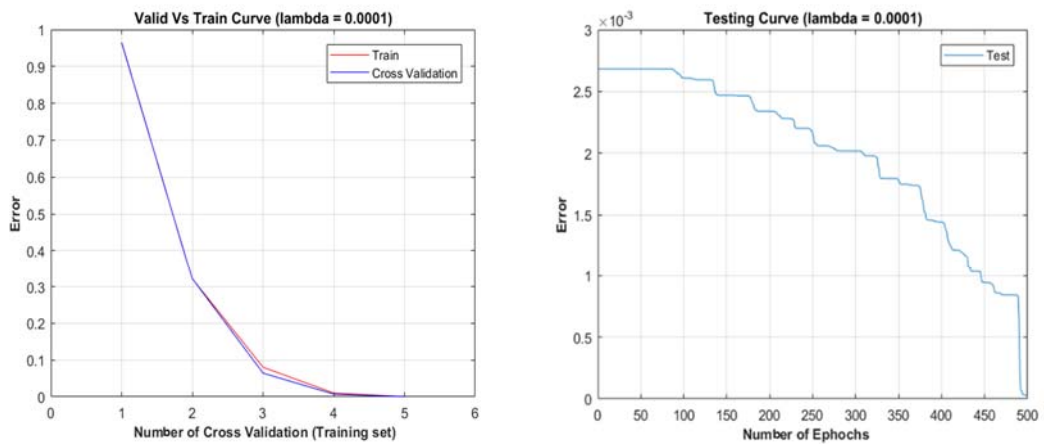


Figure 6.2: Training and testing error curve based on BPNN and FSDR.

In the second case, we use our model to select features and reduce dimensions with BPNN classifier to detect DDoS attacks in Alkasasbeh dataset. We display the results of our model where the mutual information scoring are used to do a descending order of 27 features according to the values of the information exchanged and selection 8 of the best features using the empirical CDF estimator as shown in Figure (6.3) below. The Table (6.6) below shows the order and sequence of features in the original dataset.



Figure 6.3: Empirical CDF for uncertainty MI value with Data 2.

Table (6. 6) Numbers of features selected by MI with Data 2.

NO	Feature Name	NO in Original dataset
1	PKT_DELAY	23
2	FID	9
3	PKT_ID	3
4	PKT_SIZE	7
5	PKT_SIZE	2
6	PKT_OUT	16
7	FLAGS	8
8	UTLIZATION	22

The new features space data that was selected, that used in the enhanced SVD algorithm for dimension reduction in the features ranking which influenced the reconstruction value as 96%. In this process distinguishes a fitting low-dimensional description of original data for six best potential features which are listed in the Table (6.7) below.

Table 6.7: Names of reduced features at the developed SVD with Data 2.

NO.	Name of Features Selected	No of Feature in Origin dataset	Score
1	PKT_IN	15	0.406132
2	PKT_OUT	16	0.388425
3	PKT_DELAY	23	0.373565
4	SEQ_NUMBER	10	0.320811
5	PKT_DELEAY_NODE	18	0.320811
6	FID	9	0.308251
Time for MI and DR		4.749726 sec	

Table (6.8) and Table (6.9) present all the performance metrics of our model on Alkaskasbeh dataset after dimensionality reduction. The performance metrics of two different tests on the same dataset in multiclass (Normal, UDP-Flood, Smurf, SIDDoS, and HTTP-Flood) and binary class (Normal, abnormal) are obtained. It is obvious that the average accuracy of the classifier in multi-class is 98.09% and in binary class 98.80% respectively. It is possible to note that the consuming time in the data testing using BPNN classifier was 115.21sec. Figure (6.4) shows on the left the error curve of training, either on the right showing the error curve of testing result with respect the iteration number rely on our model in FSDR by using BPNN classifier with six features. It is possible to note that the consuming time in the data testing using BPNN classifier was 115.21 sec.

Table 6.8: Performance measurement of our model with multi-class Data 2.

	Recall	Precision	F-M	Detection	FP	Accuracy
Normal	100.00	96.36	98.15	100.00	0.00	100.00
UDP-Flood	99.30	96.41	97.84	99.99	0.70	99.29
Smurf	98.54	99.83	99.18	99.62	1.80	98.17
SIDDoS	93.30	99.55	96.33	99.62	6.93	92.97
HTTP-Flood	99.99	99.22	99.61	100.00	0.01	99.99
Average	98.23	98.28	98.22	99.85	1.9	98.09
Testing Time Elapsed in NN			115.21 sec			
Training Time Elapsed in NN			924.92 sec			

Table 6.9: Performance measurement of our model with binary class Data 2.

	Recall	Precision	F-M	FA	Accuracy
Normal	100.00	96.36	98.15	0.00	100.00
Up normal	97.78	98.75	98.24	2.27	97.61
Average	98.89	97.56	98.19	1.14	98.80

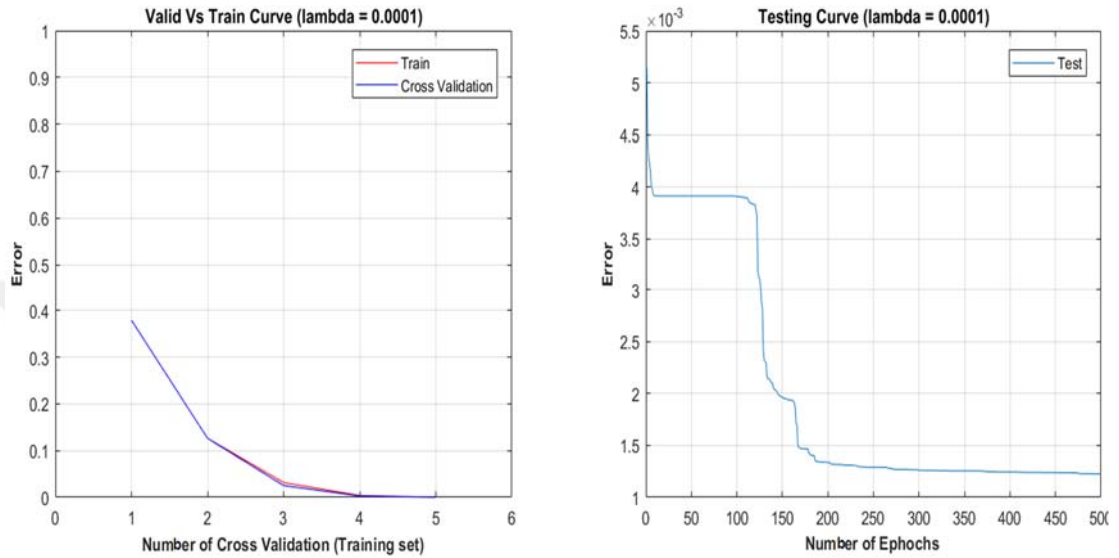


Figure 6.4: Training and testing error curve based on BPNN and FSDR.

6.5.2 Experimental Two

In the second experiment of this study, the PCA algorithm was used to reduce the size of the NSL-KDD dataset, and a DoS attack was detected using the BPNN classifier. In the first case, we will use trial and error. That examine the PCA by using five various attempts to reduce the dimensions are (4, 5, 10, 20, 35) from the original (41) feature space. Table (6.10) and Table (6.11) present all the performance metrics of two different tests for the same data are set to multiclass (Normal, Neptune, Teardrop, Smurf, Back) and binary class (Normal, abnormal). It is clear that the average highest accuracy of multi-class classifiers is 95.16 % and 93.98%, respectively, when potential ten features are used and the time consumed in the data testing that is using BPNN classifier was 34.38 sec.

Table 6.10: Performance measurement of 5 cases DR by PCA with multi-class Data 1.

NO Feature	Recall	Precision	F-M	F. A	Accuracy	Training Time	Testing Time
4	92.40	94.71	93.13	8.96	91.04	370.29	30.79
5	92.58	93.13	92.59	7.58	92.32	391.67	31.07
10	95.39	95.63	95.36	4.83	95.16	482.49	34.38
20	92.24	92.61	92.14	8.36	91.61	570.80	55.18
35	91.04	92.94	91.31	9.05	90.91	1265.20	122.92

Table 6.11: Performance measurement of 5 cases DR by PCA with binary class Data 1.

	Recall	Precision	F-M	F. A	Accuracy
4	93.75	92.65	92.89	6.19	92.81
5	88.75	94.49	91.24	10.22	88.18
10	94.77	89.92	91.90	4.86	93.98
20	92.46	92.97	92.54	8.09	90.96
35	93.37	85.06	88.08	5.99	93.27

The goal of the PCA is to skip the originally large number of attributes into a miniature set which characterizes the more prominent part of the information included in the original attributes. The highest results were obtained using in the testing PCA in the NSL-KDD dataset to detect DoS attack types (Normal, Neptune, Teardrop, Smurf, Back) at ten features, which are indicated in detail in the Table (6.12) and the Figure (6.5) below respectively.

Table 6.12: Highest performance measurement of DR by PCA with multi-class Data 1.

	Recall	Precision	F-M	F. A	Accuracy
Normal	99.47	93.77	96.54	0.53	99.47
Neptune	96.81	88.79	92.63	3.96	95.92
Teardrop	97.73	98.60	98.16	2.26	97.69
Smurf	96.65	97.58	97.11	3.41	96.48
Back	86.29	99.40	92.38	12.11	86.23
Average	95.39	95.63	95.36	4.45	95.16

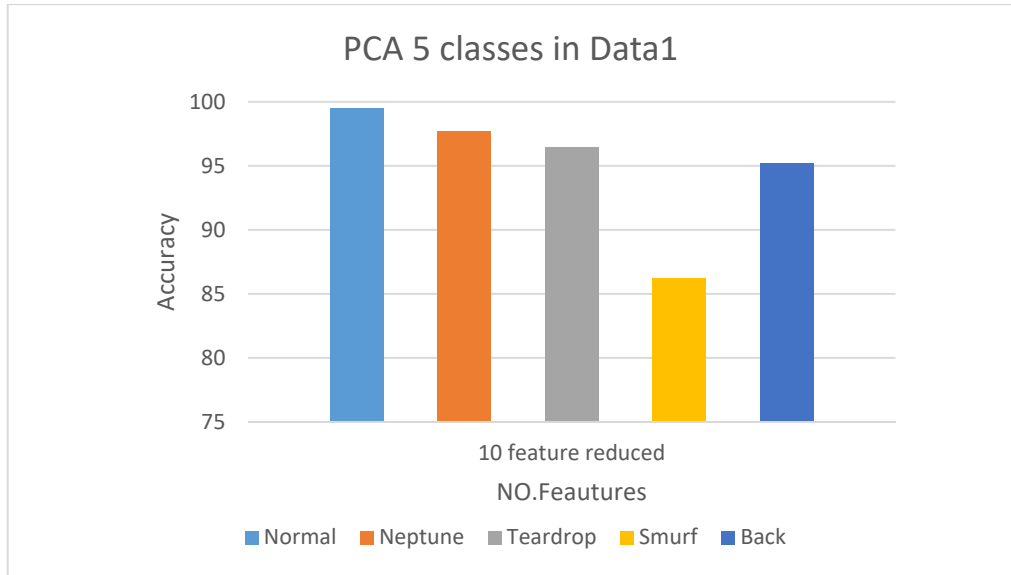


Figure 6.5: Accuracy rate of BPNN with ten selected features using PCA Data 1.

After that, the experiment of this study, the PCA algorithm was used to reduce the size of the Alkawasbeh dataset, and a DDoS attack was detected using the BPNN classifier. In the second case, we will use trial and error to examine the PCA by using five various attempts to reduce the dimensions are (5, 6, 10, 20) from the original (41) feature space. Table (6.13) and Table (6.14) present all the performance metrics of two different tests for the same data are set to multiclass (Normal, UDP-Flood, Smurf, SIDDOS, and HTTP-Flood) and binary class (Normal, abnormal). It is clear that the average highest accuracy of multi-class classifiers is 93.68% and 94.46%, respectively, when potential ten features are used and the time consumed in the data testing that is using BPNN classifier was 151.02 sec.

Table 6.13: Performance measurement of 5 DR by PCA with multi-class Data 2.

	Recall	Precision	F-M	F. A	Accuracy
Normal	93.89	87.44	90.55	6.23	93.77
UDP-Flood	98.64	89.90	94.07	1.38	98.60
Smurf	82.18	97.15	89.04	17.91	82.07
SIDDOS	97.41	97.41	97.41	2.36	97.31
HTTP-Flood	96.77	98.45	97.60	3.36	96.63
Average	93.78	94.07	93.73	6.31	93.68

Table 6.14: Performance measurement of 5 DR by PCA with multi-class Data 2.

NO Feature	Recall	Precision	F-M	F. A	Accuracy	Training Time	Testing Time
5	89.60	90.74	89.27	11.53	89.45	655.43	36.04
6	89.46	89.53	91.43	8.33	89.84	668.91	44.24
10	93.78	94.07	93.73	5.67	93.68	713.92	151.02
20	93.39	94.14	93.47	6.07	93.34	896.98	170.42

After seeing the best result in reducing the features using PCA, which the highest results were obtained using the BPNN in testing the Alkasasbeh dataset. Detection DDoS attack types include (Normal, UDP-Flood, Smurf, SIDDoS, and HTTP-Flood) at ten features, which are indicated in detail in the Table (6.15) and the Figure (6.6) below respectively.

Table 6.1: Highest performance measurement of DR by PCA with multi-class Data 1.

	Recall	Precision	F-M	F. A	Accuracy
5	96.40	85.62	89.50	7.11	92.86
6	92.12	81.63	87.18	11.4	88.56
10	94.50	89.76	91.75	5.12	93.46
20	93.82	91.58	92.54	5.75	93.71

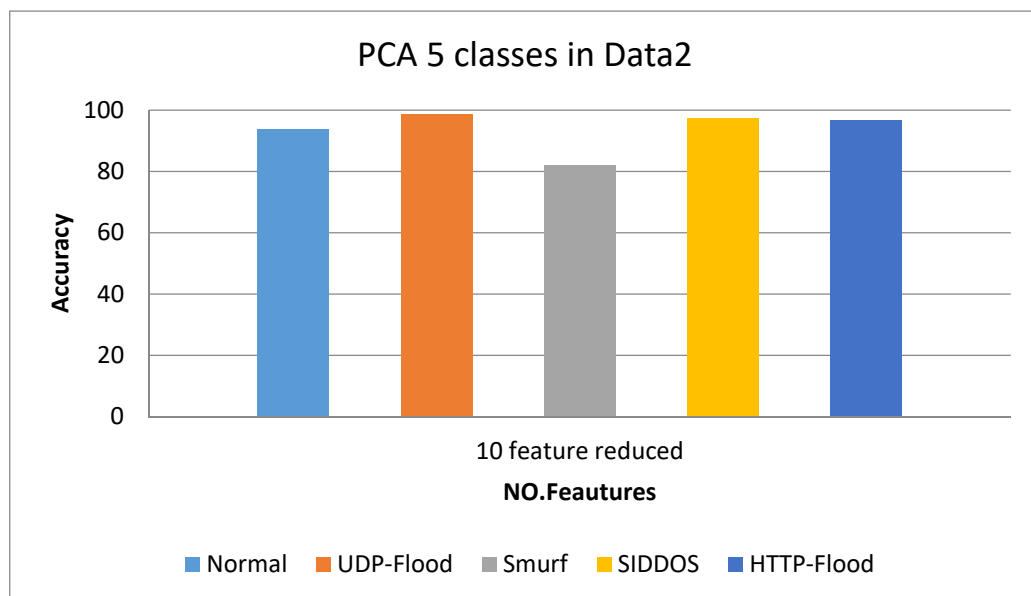


Figure 6.6: Accuracy rate of BPNN with 10 selected features using PCA Data 2.

6.5.3 Experimental Three

In the second experiment of this study, the SVD algorithm was used to reduce the size of the NSL-KDD dataset, and a DoS attack was detected using the BPNN classifier. In the first case, we will use trial and error to examine the SVD by using five various attempts to reduce the dimensions are (4, 5, 10, 20, 35) from the original (41) feature space. Reduce dimensions for some features that we define in the SVD algorithm.

This happens after calculating the mean data in the matrix for each feature where there is no indicator to choose the important features. Table (6.16) and Table (6.17) present all the performance metrics of two different tests for the same data are set to multiclass (Normal, Neptune, Teardrop, Smurf, Back) and binary class (Normal, abnormal). It is clear that the average highest accuracy of multi-class classifiers is 95.07% and 95.55%, respectively, when potential 20 features are used and the time consumed in the data testing using BPNN classifier was 41.75 sec.

Table 6.16: Performance measurement of 5 DR SVD with multi-class Data 1.

NO Feature	Recall	Precision	F-M	F. A	Accuracy	Training Time/Sec	Testing Time
4	92.40	94.71	93.14	7.56	91.04	303.64	24.58
5	93.43	95.50	93.94	5.48	93.26	332.20	30.07
10	94.71	95.00	94.71	5.10	94.48	550.21	31.34
20	95.63	95.62	95.60	4.62	95.07	680.27	41.75
35	90.00	91.91	90.31	9.66	89.12	891.93	61.77

Table 6.17: Performance measurement of 5 DR by SVD with binary- class Data 1.

NO Feature	Recall	Precision	F-M	F. A	Accuracy
4	93.75	92.65	92.89	6.19	92.81
5	95.60	90.42	92.37	3.72	94.41
10	95.22	94.46	94.74	4.61	95.08
20	95.64	95.97	95.79	4.68	95.55
35	93.34	94.07	93.62	6.22	93.20

The purpose of the comparison using SVD is to show the reduction of the dimension space and remove the noise. Taking into consideration the performance of

the system and not affected significantly when cutting features. The above highest results were obtained using the BPNN in the testing NSL-KDD dataset to detect DDoS attack types (Normal, Neptune, Teardrop, Smurf, Back) at 20 features, which are indicated in detail in the Table (6.18) and the Figure (6.7) below respectively.

Table 6.2: Highest performance measurement by SVD with multi-class Data 1.

	Recall	Precision	F-M	F. A	Accuracy
Normal	95.66	96.54	96.10	4.78	95.02
Neptune	89.25	94.47	91.78	9.92	89.02
Teardrop	96.44	93.24	94.81	4.69	95.14
Smurf	96.83	94.81	95.81	3.68	96.20
Back	99.97	99.06	99.52	0.03	99.97
Average	95.63	95.62	95.60	4.62	95.07

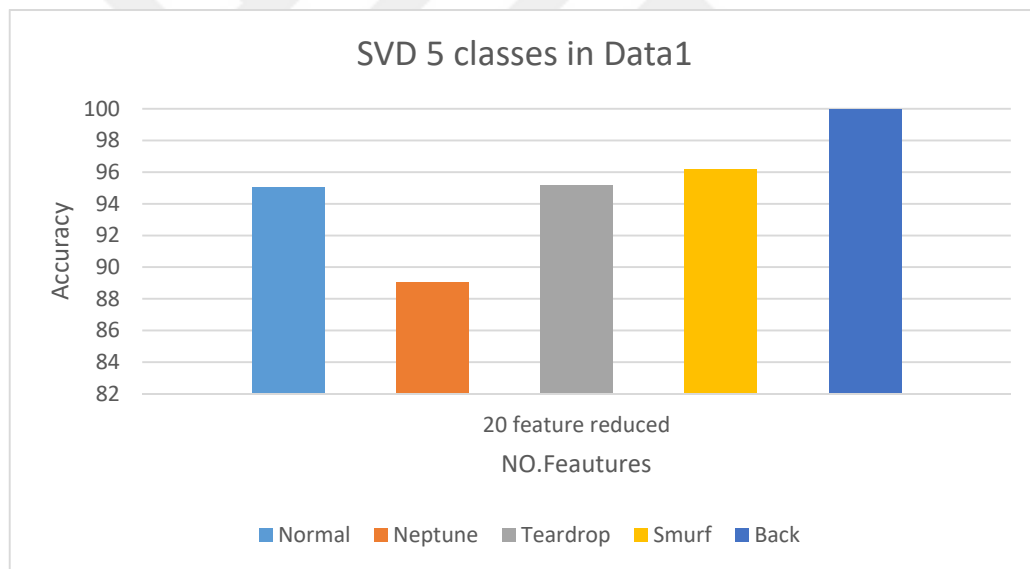


Figure 6.7: Accuracy rate of BPNN with 20 selected features by SVD Data 1.

In the second experiment of this study, the SVD algorithm was used to reduce the size of the Alkansasbeh dataset, and a DDoS attack was detected using the BPNN classifier. In the second case, we will use trial and error to examine the SVD by using five various attempts to reduce the dimensions are (5, 6, 10, 20) from the original (41) feature space. Table (6.19) and Table (6.20) present all the performance metrics of two different tests for the same data are set to multiclass (Normal, UDP-Flood, Smurf, SIDDoS, and HTTP-Flood) and binary class (Normal, abnormal). It is clear that the

average highest accuracy of multi-class classifiers is 93.34% and 94.46%, respectively, when potential ten features are used and the time consumed in the data testing using BPNN classifier was 122 Sec.

Table 6.3: Performance measurement of 5 cases DR by SVD with multi-class Data 2.

	Recall	Precision	F-M	F. A	Accuracy
Normal	88.34	89.98	88.64	11.27	88.31
UDP-Flood	88.98	87.31	88.14	9.94	88.87
Smurf	94.67	94.77	94.81	5.75	94.89
SIDDoS	96.94	95.59	96.26	2.98	96.93
HTTP-Flood	97.78	99.07	98.42	2.37	97.58
Average	93.14	93.14	93.14	6.86	93.34

Table 6. 20: Performance measurement of 5 cases DR by SVD with multi-class Data 2.

NO Feature	Recall	Precision	F-M	F. A	Accuracy
5	93.14	93.14	93.14	7.48	92.52
6	92.15	81.69	87.21	6.48	88.59
10	89.76	91.75	91.45	5.12	94.46
20	93.82	91.58	92.54	5.75	93.71

After seeing the best result in reducing the features by using SVD, which the highest results were obtained using BPNN in the testing the Alkasasbeh dataset too. Detection DDoS attack types include (Normal, UDP-Flood, Smurf, SIDDoS, and HTTP-Flood) at ten features, which are indicated in detail in the Table (6.21) and the Figure (6.8) below respectively.

Table 6.21: Highest performance measurement in DR by SVD with multi-class Data 2.

NO Feature	Recall	Precision	F-M	F. A	Accuracy	Training Time	Testing Time
5	92.35	94.43	93.79	8.65	92.18	1020.68	91.07
6	89.51	89.58	91.44	9.32	89.88	1061.70	98.05
10	93.39	94.14	93.47	6.07	93.34	1246.36	122.00
20	91.09	91.48	91.05	9.00	90.99	1860.089	276 .24

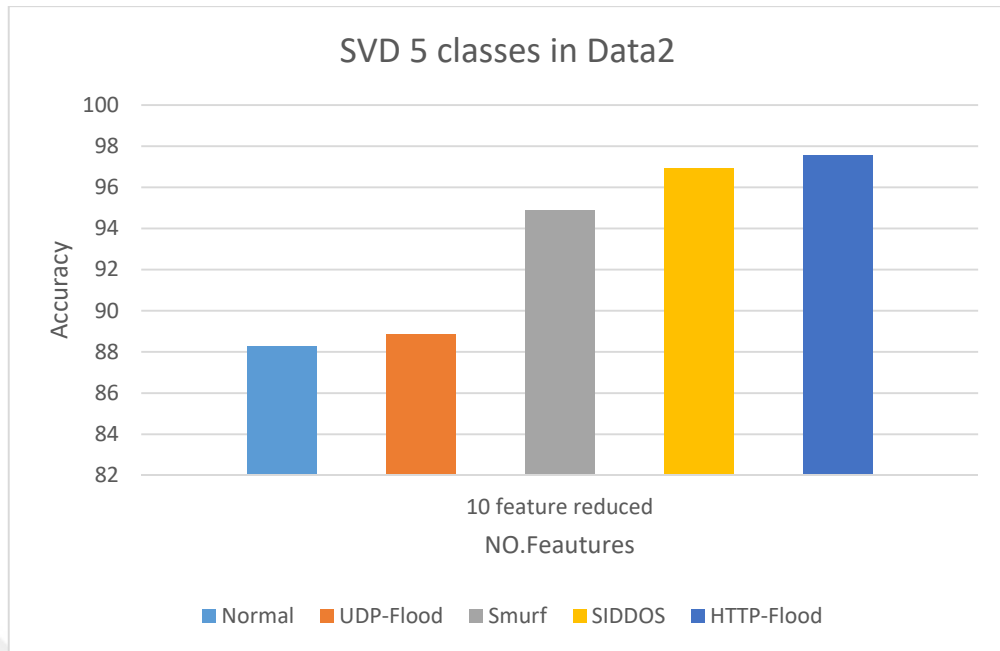


Figure 6.8: Accuracy rate of BPNN with ten selected features by SVD Data 2.

6.5.4 Experimental Four

In our recent experience in this chapter, the NSL-KDD dataset was tested using all features space in the dataset (41) without methods of reducing the dimensions, where was detected of DoS attack using the BPNN classifier. Table (6.22) and Table (6.23) present all the performance metrics of two different tests for the same data are set to multiclass and binary class (Normal, abnormal). It is clear that the average highest accuracy of multi-class classifiers is 94.97% and 94.51%, respectively, while the time consumed in the data testing using BPNN classifier was 230.68 sec ,the results will be compared with our proposed model and other methods

Table 6.22: Performance measurement with multi-class Data 1 all features.

	Recall	Precision	F-M	FA	Accuracy
Normal	94.66	86.94	90.63	5.95	93.73
Neptune	99.82	95.34	97.53	0.18	99.82
Teardrop	97.72	96.71	97.21	2.46	97.53
Smurf	87.38	99.22	92.93	11.26	87.32
Back	97.14	99.77	98.44	3.46	96.44
Average	95.34	95.59	95.35	4.98	94.97
Testing Time Elapsed in NN			230.68 seconds		
Training Time Elapsed in NN			1692.99 seconds		

Table 6.4: Performance Measurement with binary-class Data 1 all features.

	Recall	Precision	F-M	F. A	Accuracy
Normal	94.66	86.94	90.63	5.95	93.73
Abnormal	95.52	97.76	96.53	4.33	95.28
Average	95.09	92.35	93.58	5.14	94.51

In the second case, the Alkansasbeh dataset was tested using all features space in the dataset (41) without methods of reducing the dimensions, where was detected of DDoS attack using the BPNN classifier. Table (6.24) and Table (6.25) present all the performance metrics of two different tests for the same data are set to multiclass (Normal, Neptune, Teardrop, Smurf, Back) and binary class (Normal, abnormal). It is clear that the average highest accuracy of multi-class classifiers is 92.44 % and 93.64% respectively, while the time consumed in the data testing using BPNN classifier was 804.74 sec.

Table 6.24: Performance measurement with multi-class Data2 all features

	Recall	Precision	F-M	F. A	Accuracy
Normal	95.62	83.03	88.89	4.19	95.62
UDP-Flood	83.63	91.22	87.26	14.15	83.54
Smurf	96.73	98.89	97.80	3.23	96.67
SIDDoS	95.24	95.16	95.20	4.84	94.93
HTTP-Flood	92.74	97.43	95.03	7.97	91.46
Average	92.79	93.15	92.84	6.88	92.44
Testing Time Elapsed in NN			804.74 sec		
Training Time Elapsed in NN			2964.16 sec		

Table 6.25: Performance measurement with binary-class Data2 all features.

	Recall	Precision	F-M	F. A	Accuracy
Normal	95.62	83.03	88.89	4.19	95.62
Up normal	92.09	95.68	93.82	7.55	91.65
Average	93.85	89.36	91.35	5.87	93.64

6.6 Results Discussion

There are four sets of experiments implemented by this chapter, which are our model FSDR experiment, PCA experiment, SVD experimental and non-dimension reduction experiment. All experiments are implementing in Neural Network BPNN classifier for testing two datasets NSL-KDD and Alkasasbeh dataset for detection DoS and DDoS attacks in both multi-class datasets and the binary class dataset. The performance results of the Neuronal Neuron Network with both the multi-layer and the binary classes were obtained in two sets of datasets with the highest accuracy of 99.1, 97.92, 98.09, and 98.80 with our FSDR compared to the other three trials shown in Table (6.26) and Table (6.27), respectively.

Table 6.5: Comparison accuracy and time the multi - class Data 1 cases in multi-class Data 1.

Reduction Approach	Accuracy In Multi	Accuracy In Binary	Testing Time/Sec	No. of Features
Full Data (Without reduction)	94.97	94.51	230.68	Full features (41)
PCA (With better features)	95.16	93.98	34.38	Select (10/41)
SVD (With better features)	95.07	95.55	41.75	Select (20/41)
Ours model (FSDR)	99.10	97.92	29.76	4

Table 6.27: Comparison accuracy and time FSDR with other cases in binary-class Data 1.

Reduction Approach	Accuracy In Multi	Accuracy In Binary	Testing Time/Sec	No. of Features
Full Data (Without reduction)	92.44	93.64	804.74	Full features (27)
PCA (With better feature)	93.68	93.46	151.02	Select (20/27)
SVD (With better features)	93.34	94.46	122	Select (10/27)
Ours model (FSDR)	98.09	98.80	115.21	6

The results indicate that the higher classification accuracy is achieved by a diminutive number of features during the suggested method is employed as feature selection and reduced dimensionality of big dimensional datasets. That is accomplished by use of MI with the ECDF estimator is the first guide in selecting the best features of the total dataset, where the descending order of the original group values is performed according to the highest exchange information among the variables. The uncertainty threshold value is usually set to a higher value so that the

nonnegative value and not very small in the mutuality of information between the two variables above the threshold and can be selected to subset features. Where ECDF determines the uncertainty values to ignore negative and non-useful values based on MI's work which finds an optimal subset of all attributes and removes irrelevant and redundant attributes. That identify 17 features out of 41 in the first dataset and 8 out of 27 for the second dataset.

Since objects in the MI feature selection dataset are not ideal for classification. So we used another guide to reduce dimensions is the improved SVD algorithm, SVD. That uses four features of mapped data instead of all 17 features and six features of the newly selected in early phase instead of all eight features, based on reconstruction that means maintaining 96% variance ensures that not a lot of the information is dropped. SVD, one may determine which eigenvalues are higher important. Figure (6.9) and Figure (6.10) show the effect of the number of dimensions that have been reduced on the resolution rate in the DoS and DDoS attack detectors.

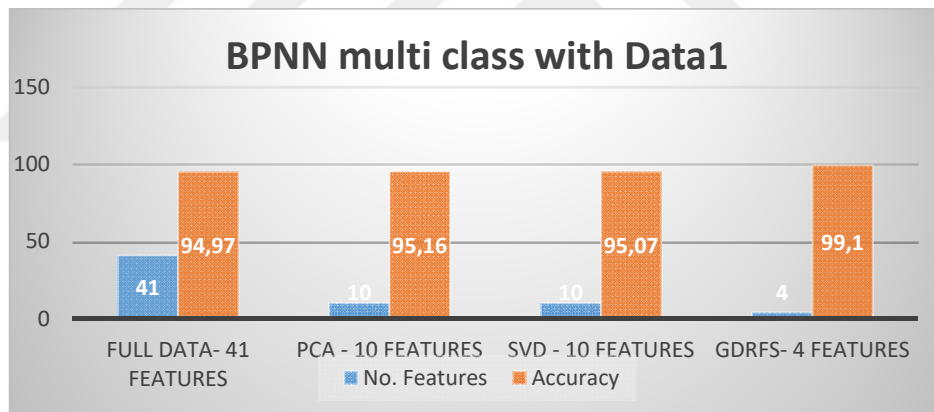


Figure 6.9: Comparison between (FSDR) and others results with multi-class Data 1.

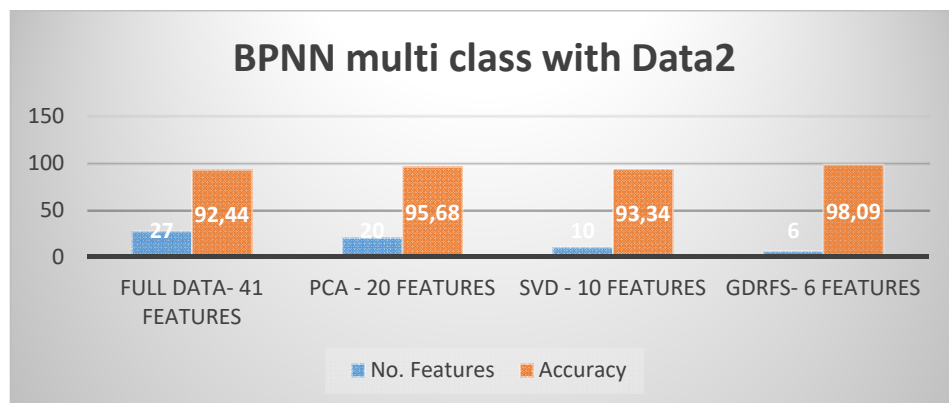


Figure 6.10: Comparison between (FSDR) and others results with multi-class Data 2.

Increased time is an unacceptable factor in intrusion detection systems as we need a near real-time solution for today's high-speed connections. The computational overhead of anomaly-based IDSs are significantly reduced. It makes them viable for real-time deployment in high-speed networks. With our model applied in both datasets, which took the least time to complete the classification, which is 29.76 seconds and 115 seconds, respectively, compared to the remaining three experiments as shown in Figure (6.11) and Figure (6.12).

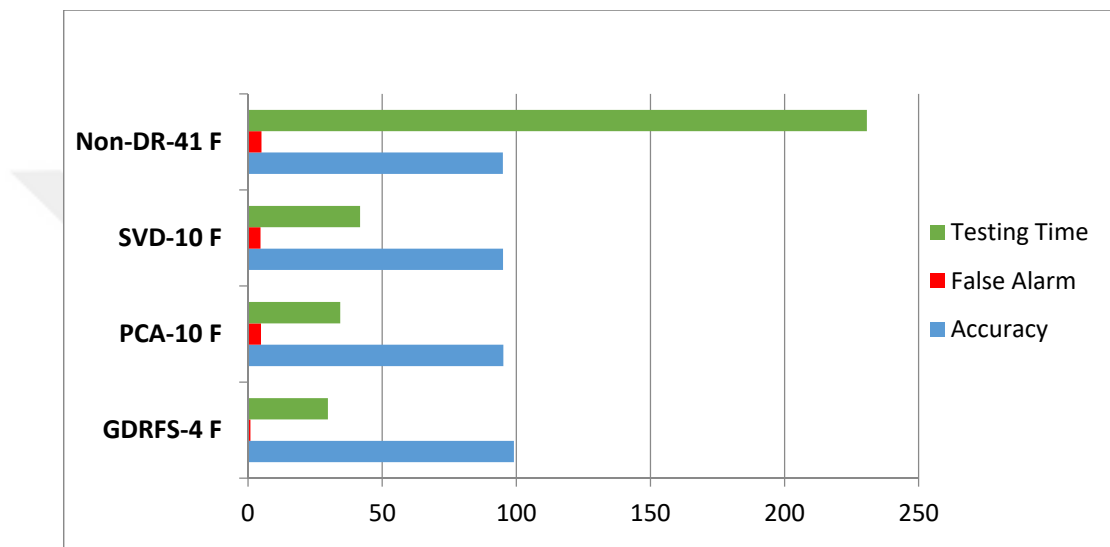


Figure 6.11: Comparison (FSDR) and others results with multi-class Data1 consuming time in testing.

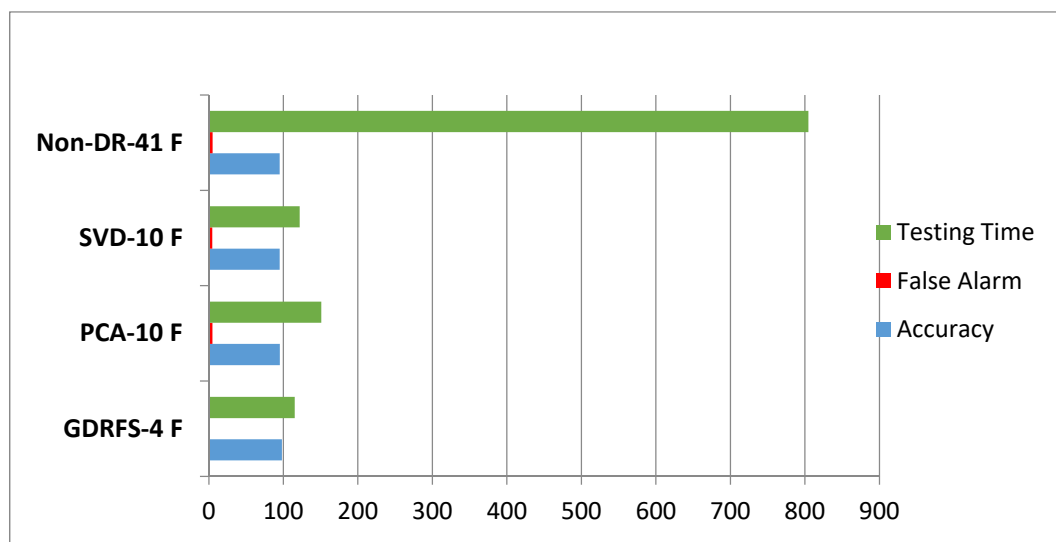


Figure 6.5: Comparison (FSDR) and others results with multi-class Data1 consuming time in testing.

CHAPTER SEVEN

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

For traffic network analysis, we need a preprocessing stage to clean up the data and convert it to an appropriate format so that effective IDS is done. In this thesis, we introduced an optimal feature selection and dimensionality reduction with back-propagation neural network algorithm for detecting DoS and DDoS attacks in two datasets. Four kinds of DoS attacks named (Neptune, Teardrop, Smurf, Back) on the NSL- KDD dataset. The other dataset contains modern types of attack DDoS distributed between the layers of Application and Network and contains four types of DDoS attack (Smurf, UDP-FLOOD, SIDDoS, and HTTP-Flood). Some of the feature selection methods have a specific bias to considering attributes having many different values. However, attributes with very low information value to seem to receive unfair profits. Moreover, in the feature selection phase, MI algorithm was used to rank all datasets with scoring for each feature by descending order. Besides, empirical cumulative distribution function (ECDF) was used for expanding the variance between weights MI and then selecting subset features to depend on the uncertainty value in MI theory. The Negative scoring values are ignored because they are useless and positive values are chosen. This is considered the first guide to selecting a partial set of features in the early stage.

On the dimensionality reduction side, there are many ways to minimize dimensions in the dataset such as PCA, SVD etc. In our research, SVD was adopted because it works analyzing matrices efficiently. We enhanced SVD algorithms. This part is considered the second stage and after completing the selection of subset features from the original datasets by using MI. We will use SVD's enhanced dimension reduction algorithm for the new set of features. Our model, which depends on the

regular SVD, but our idea is to rank the Eigenvalue which is in diagonal in two cases and accelerated Singular Value Decomposition. Used to reconstruct the significant factors of the variation of the new sub-feature space data at ration 96%. We find the essential features, the first benchmark of the NSL - KDD dataset has reduced the dimensions from 41 to 4 and in addition to the new dataset of DDoS attacks created by Alkawasbeh and its features reduced from 27 to 6.

We obtained highest classification accuracy by the Neural Network BPNN classifier to testing two datasets NSL-KDD and Alkawasbeh dataset for detection DOS and DDoS attacks in both multi-class datasets and the binary class dataset. The performance results of the Neuronal Neuron Network with both the multi-layer and the binary classes were obtained in two sets of datasets with the highest accuracy of 99.1%, 97.92%, 98.09%, and 98.80%. We compared the FSDR method with the PCA and SVD reduction methods as well as the two sets of data tests with the full set of features.

7.2 Future Works

The suggested method could produce effective feature selection and dimensionality reduction for IDS. Our method proved to have the highest accuracy in detecting DDoS attacks while reducing the time spent on the test. It is an efficient way to identify DDoS attacks while dealing with a large volume of traffic network contains many features. In the future work, our method will be tested with other classification algorithms. We will implement our model with real-time to improve the work of IDS.

REFERENCES

1. Alkasassbeh, M., et al., *Detecting Distributed Denial of Service Attacks Using Data Mining Techniques*. International Journal of Advanced Computer Science and Applications, 2016. 7(1).
2. Networks, A. *Arbor Networks' 12th Annual Worldwide Infrastructure Security Report Finds Attacker Innovation and IoT Exploitation Fuel DDoS Attack Landscape*. 2016 [cited 12th; Available from: <https://www.arbornetworks.com/arbor-networks-12th-annual-worldwide-infrastructure-security-report-finds-attacker-innovation-and-iot-exploitation-fuel-ddos-attack-landscape>].
3. Balkanli, E., A.N. Zincir-Heywood, and M.I. Heywood. *Feature selection for robust backscatter DDoS detection*. in *Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th*. 2015. IEEE.
4. Haddadi, F., et al. *Intrusion detection and attack classification using feed-forward neural network*. in *Computer and Network Technology (ICCNT), 2010 Second International Conference on*. 2010. IEEE.
5. Han, J., J. Pei, and M. Kamber, *Data mining: concepts and techniques*. 2011: Elsevier.
6. Bhuyan, M.H., et al., *Detecting distributed denial of service attacks: methods, tools and future directions*. The Computer Journal, 2013. 57(4): p. 537-556.
7. Hoque, N., D. Bhattacharyya, and J. Kalita. *Denial of Service Attack Detection using Multivariate Correlation Analysis*. in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*. 2016. ACM.
8. Özçelik, İ. and R.R. Brooks. *Cusum-entropy: an efficient method for DDoS attack detection*. in *Smart Grid Congress and Fair (ICSG), 2016 4th International Istanbul*. 2016. IEEE.
9. Lee, K., et al., *DDoS attack detection method using cluster analysis*. Expert Systems with Applications, 2008. 34(3): p. 1659-1665.

10. Al-Mamory, S.O. and Z.M. Ali, *Using DBSCAN Clustering Algorithm in Detecting DDoS Attack*.
11. Hsieh, C.-J. and T.-Y. Chan. *Detection DDoS attacks based on neural-network using Apache Spark*. in *Applied System Innovation (ICASI), 2016 International Conference on*. 2016. IEEE.
12. Barati, M., et al. *Distributed Denial of Service detection using hybrid machine learning technique*. in *Biometrics and Security Technologies (ISBAST), 2014 International Symposium on*. 2014. IEEE.
13. Oo, T.T. and T. Phyu. *Analysis of DDoS Detection System based on Anomaly Detection System*. in *International Conference on Advances in Engineering and Technology (ICAET'2014)*. Singapore. 2014.
14. Purwanto, Y. and B. Rahardjo. *Traffic anomaly detection in DDoS flooding attack*. in *Telecommunication Systems Services and Applications (TSSA), 2014 8th International Conference on*. 2014. IEEE.
15. Yadav, S. and S. Subramanian. *Detection of Application Layer DDoS attack by feature learning using Stacked AutoEncoder*. in *Computational Techniques in Information and Communication Technologies (ICCTICT), 2016 International Conference on*. 2016. IEEE.
16. Keerthika, M.R., et al., *Cluster-Based DDoS Detection Method in Data Mining*.
17. Kato, K. and V. Klyuev. *Large-scale network packet analysis for intelligent DDoS attack detection development*. in *Internet Technology and Secured Transactions (ICITST), 2014 9th International Conference for*. 2014. IEEE.
18. Sasikala, S., S.A. alias Balamurugan, and S. Geetha, *Multi filtration feature selection (MFFS) to improve discriminatory ability in clinical data set*. Applied Computing and Informatics, 2014.
19. Liao, Q., et al. *Feature extraction and construction of application layer DDoS attack based on user behavior*. in *Control Conference (CCC), 2014 33rd Chinese*. 2014. IEEE.
20. Buragohain, C., et al., *Anomaly-based DDoS Attack Detection*. International Journal of Computer Applications, 2015. 123(17).

21. Robinson, R.R. and C. Thomas. *Ranking of machine learning algorithms based on the performance in classifying DDoS attacks*. in *Intelligent Computational Systems (RAICS), 2015 IEEE Recent Advances in*. 2015. IEEE.
22. Fadlil, A., I. Riadi, and S. Aji, *A Novel DDoS Attack Detection Based on Gaussian Naive Bayes*. *Bulletin of Electrical Engineering and Informatics*, 2017. 6(2): p. 140-148.
23. Jia, B., et al., *A DDoS Attack Detection Method Based on Hybrid Heterogeneous Multiclassifier Ensemble Learning*. *Journal of Electrical and Computer Engineering*, 2017. 2017.
24. Harbola, A., J. Harbola, and K.S. Vaisla. *Improved Intrusion Detection in DDoS Applying Feature Selection Using Rank & Score of Attributes in KDD-99 Data Set*. in *Computational Intelligence and Communication Networks (CICN), 2014 International Conference on*. 2014. IEEE.
25. Osanaiye, O., et al., *Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing*. *EURASIP Journal on Wireless Communications and Networking*, 2016. 2016(1): p. 130.
26. Khan, S., et al., *Feature Selection of Denial-of-Service Attacks Using Entropy and Granular Computing*. *Arabian Journal for Science and Engineering*, 2017: p. 1-10.
27. Cbsnews. *"WannaCry" ransomware attack losses could reach \$4 billion*. 2017 [cited November Available from: www.cbsnews.com/news/wannacry-ransomware-attacks-wannacry-virus-losses
28. Floridi, L., *The Unsustainable Fragility of the Digital, and What to Do About It*. *Philosophy & Technology*, 2017. 30(3): p. 259-261.
29. Anderson, J.P., *Computer Security Technology Planning Study. Volume 2*. 1972, Anderson (James P) and Co Fort Washington PA.
30. The Statistics Portal, *Cyber crime: average company loss in selected countries 2017*.
31. Kabiri, P. and A.A. Ghorbani, *Research on intrusion detection and response: A survey*. *IJ Network Security*, 2005. 1(2): p. 84-102.
32. Garcia-Teodoro, P., et al., *Anomaly-based network intrusion detection: Techniques, systems and challenges*. *computers & security*, 2009. 28(1): p. 18-28.

33. Jyothsna, V., V.R. Prasad, and K.M. Prasad, *A review of anomaly based intrusion detection systems*. International Journal of Computer Applications, 2011. 28(7): p. 26-35.
34. Lau, F., et al. *Distributed denial of service attacks*. in *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*. 2000. IEEE.
35. Mansfield-Devine, S., *The growth and evolution of DDoS*. Network Security, 2015. 2015(10): p. 13-20.
36. Geoff Huston, A., *Network Service Models - The Internet Protocol Journal*, . June 2013
37. Peng, T., C. Leckie, and K. Ramamohanarao, *Survey of network-based defense mechanisms countering the DoS and DDoS problems*. ACM Computing Surveys (CSUR), 2007. 39(1): p. 3.
38. Kumar, S. *Smurf-based distributed denial of service (ddos) attack amplification in internet*. in *Internet Monitoring and Protection, 2007. ICIMP 2007. Second International Conference on*. 2007. IEEE.
39. Lemon, J. *Resisting SYN Flood DoS Attacks with a SYN Cache*. in *BSDCon*. 2002.
40. Senie, D. and P. Ferguson, *Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing*. Network, 1998.
41. Bora, G., et al., *OSI reference model: An overview*. International Journal of Computer Trends and Technology (IJCTT), 2014. 7(4): p. 214-218.
42. Specht, S.M. and R.B. Lee. *Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures*. in *ISCA PDCS*. 2004.
43. Mansfield-Devine, S., *DDoS: threats and mitigation*. Network Security, 2011. 2011(12): p. 5-12.
44. Garg, D. *Ddos mitigation techniques-A survey*. in *International Conference on Advanced Computing, Communication and Networks*. 2011.
45. Srivastava, A., et al., *A recent survey on DDoS attacks and defense mechanisms*. Advances in Parallel Distributed Computing, 2011: p. 570-580.

46. Fürnkranz, J., D. Gamberger, and N. Lavrač, *Foundations of rule learning*. 2012: Springer Science & Business Media.
47. Jeon, B. and D.A. Landgrebe, *Partially supervised classification using weighted unsupervised clustering*. IEEE Transactions on Geoscience and Remote Sensing, 1999. 37(2): p. 1073-1079.
48. Haykin, S.S., *Neural networks: a comprehensive foundation*. 2001: Tsinghua University Press.
49. Anderson, J.A., *An introduction to neural networks*. 1995: MIT press.
50. Saad, R.M., et al., *An intelligent icmpv6 ddos flooding-attack detection framework (v6iids) using back-propagation neural network*. IETE Technical Review, 2016. 33(3): p. 244-255.
51. Data, M.C., *Secondary Analysis of Electronic Health Records*. 2016: Springer.
52. Han, J., M. Kamber, and J. Pei, *Data Preprocessing*. Data mining: concepts and techniques. San Francisco: Morgan Kaufmann, 2006: p. 47-97.
53. Liu, H. and H. Motoda, *Feature selection for knowledge discovery and data mining*. Vol. 454. 2012: Springer Science & Business Media.
54. Roweis, S.T. and L.K. Saul, *Nonlinear dimensionality reduction by locally linear embedding*. science, 2000. 290(5500): p. 2323-2326.
55. Jolliffe, I.T., *Principal Component Analysis and Factor Analysis*, in *Principal component analysis*. 1986, Springer. p. 115-128.
56. Tufféry, S., *Data mining and statistics for decision making*. Vol. 2. 2011: Wiley Chichester.
57. Demmel, *Applied numerical linear algebra*. 1997.
58. Kotsiantis, S.B., I. Zaharakis, and P. Pintelas, *Supervised machine learning: A review of classification techniques*. 2007.
59. Guyon, I. and A. Elisseeff, *An introduction to variable and feature selection*. Journal of machine learning research, 2003. 3(Mar): p. 1157-1182.

60. Kohavi, R. and G.H. John, *Wrappers for feature subset selection*. Artificial intelligence, 1997. 97(1-2): p. 273-324.
61. Van der Vaart, A.W., *Asymptotic statistics*. Vol. 3. 1998: Cambridge university press.
62. Stolfo et al. *KDD Cup 1999 Data*. 1999 October 28, 1999; Available from: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.
63. M. Tavallae. (*NSL-KDD*) *A Detailed Analysis of the KDD CUP 99 Data Set*. 2009; Available from: <http://www.unb.ca/cic/datasets/nsl.html>.
64. al, T.e., *A Detailed Analysis of the KDD CUP 99 Data Set*. 2009.
65. Stolfo, S.J., et al. *Cost-based modeling for fraud and intrusion detection: Results from the JAM project*. in *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*. 2000. IEEE.
66. MIT, L.L.a. *DARPA INTRUSION DETECTION EVALUATION 1998*; Available from: <https://www.ll.mit.edu/ideval/data/1998data.html>.
67. Tavallae, M., et al. *A detailed analysis of the KDD CUP 99 data set*. in *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*. 2009. IEEE.
68. Mouhammd, A. *Dataset- Detecting Distributed Denial of Service Attacks Using Data Mining Techniques*. International Journal of Advanced Computer Science and Applications 2016; Available from: https://www.researchgate.net/publication/292967044_Dataset-_Detecting_Distributed_Denial_of_Service_Attacks_Using_Data_Mining_Techniques.
69. Strickland, J., *Predictive modeling and analytics*. 2014: Lulu. com.
70. Malley, B., D. Ramazzotti, and J.T.-y. Wu, *Data Pre-processing*, in *Secondary Analysis of Electronic Health Records*. 2016, Springer. p. 115-141.
71. Krizhevsky, A., I. Sutskever, and G.E. Hinton. *Imagenet classification with deep convolutional neural networks*. in *Advances in neural information processing systems*. 2012.
72. Al Shalabi, L., Z. Shaaban, and B. Kasasbeh, *Data mining: A preprocessing engine*. Journal of Computer Science, 2006. 2(9): p. 735-739.

73. Estévez, P.A., et al., *Normalized mutual information feature selection*. IEEE Transactions on Neural Networks, 2009. 20(2): p. 189-201.
74. Haykin, S.S., et al., *Neural networks and learning machines*. Vol. 3. 2009: Pearson Upper Saddle River, NJ, USA:.
75. Ebrahimi, M.S. and H.K. Abadi, *Study of Residual Networks for Image Recognition*.
76. Sterlin, P., *Overfitting prevention with cross-validation*. Supervised Machine Learning Report, 2007. 83.
77. The MathWorks, I., Natick, Massachusetts, United States. *MATLAB and Statistics Toolbox Release 2014a*,. 2014 November 2017]; Available from: <https://www.mathworks.com/company/newsroom/mathworks-announces-release-2014a-of-the-matlab-and-simulink-product-families.html>.
78. Srinivasulu, P., et al., *Classifying the network intrusion attacks using data mining classification methods and their performance comparison*. International Journal of Computer Science and Network Security, 2009. 9(6): p. 11-18.
79. Shyu, M.-L., et al., *A novel anomaly detection scheme based on principal component classifier*. 2003, Miami Univ Coral Gables FL Dept Of Electrical And Computer Engineering.

CURRICULUM VITAE

Name : Saif Abdul Fattah Abdul Khaleq AL-HELLI
Date of Birth : 09/04/1984
Title : Senior Programmers.
Address : 213/Harithiya - Al-Kindi Street, Baghdad, Iraq
Tel : 009647901390422-05393010042
Email : saif.it.84@gmail.com



Occupation Employee in the Department of Information Technology-Section of Networks and Internet -Ministry of construction, Housing, and Public Municipalities-Haifa St Baghdad.

Specialize: In the network management area, servers and information security.

EDUCATION AND QUALIFICATIONS

From (Oct. 1, 2001) to (July. 10, 2005): BSc in Computer Science, Al - Mamoun University College.