

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

**A NOVEL INTRUSION DETECTION MODEL BASED ON TF.IDF AND C4.5
ALGORITHMS**



MASTER THESIS

Khaldoon AWADH

DEPARTMENT OF INFORMATION TECHNOLOGY

MASTER THESIS PROGRAM

NOVEMBER 2017

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

**A NOVEL INTRUSION DETECTION MODEL BASED ON TF.IDF AND C4.5
ALGORITHMS**



MASTER THESIS

Khaldoon AWADH

1406050024

DEPARTMENT OF INFORMATION TECHNOLOGY

MASTER THESIS PROGRAM

Supervisor: Asst. Prof. Dr. Ayhan AKBAŞ

Khaldoon AWADH, having the student number 1406050024 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: "A NOVEL INTRUSION DETECTION MODEL BASED ON TF.IDF AND C4.5 ALGORITHM".

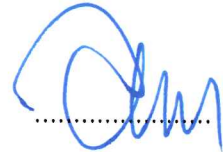
Supervisor: Asst. Prof. Dr. Ayhan AKBAŞ
University of Turkish Aeronautical Association



Jury Members: Assoc. Prof. Dr. Fırat HARDALAC
Gazi University



Asst. Prof. Dr. Tansel DÖKEROĞLU
University of Turkish Aeronautical Association



Asst. Prof. Dr. Ayhan AKBAŞ
University of Turkish Aeronautical Association



Thesis Defense Date: 8 November 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.



8.11.2017

Khaldoon AWADH

ACKNOWLEDGEMENTS

I would like to thank my family for their continuous support. Also many thanks for all of my instructors, especially my supervisor Asst. Prof. Dr. Ayhan AKBAŞ for his guidance and assistance.

Thank you THK.

November 2017

Khaldoon AWADH

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ACRONYMS	x
ABSTRACT	xi
ÖZET	xiii
CHAPTER ONE	1
1. INTRODUCTION	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Objective and Questions of the Research	3
1.4 Methodology	3
1.5 Contributions	4
1.6 Organization of the Thesis	5
CHAPTER TWO	6
2. LITERATURE SURVEY	6
2.1 Introduction	6
2.2 Related Works	6
2.3 Summary	11
CHAPTER THREE	13
3. BACKGROUND OF IDS AND DATA MINING	13
3.1 Introduction	13
3.2 Network Security	13
3.2.1 Intruder	14
3.2.2 Intrusion	14
3.2.3 Confidentiality	15
3.2.4 Integrity	15
3.2.5 Availability	15
3.2.6 Firewall	15
3.3 Network Activities	16
3.4 Network Vulnerabilities Sources	17
3.4 A History of Intrusion Detection System	18
3.5 Types of Attacks against Network Security	19
3.6 Intrusion Detection System (IDS)	22
3.6.1 Network Intrusion Detection System (NIDS)	22
3.6.2 Host Intrusion Detection Systems (HIDS)	23
3.6.3 Hybrid of HIDS and NIDS	24
3.7 General Intrusion Detection System Model	24
3.8 IDS Alarms	25
3.9 Intrusion Detection Techniques	25

3.9.1	Misuse Technique Based IDS	26
3.9.2	Anomaly based IDS.....	26
3.9.3	Hybrid Based IDS.....	27
3.9.3.1	Hybrid anomaly misuse sequence detection system.....	27
3.9.3.2	Hybrid misuse anomaly sequence detection system.....	28
3.9.3.3	Hybrid parallel detection system.....	28
3.9.4	Scan Detection Based IDS.....	29
3.9.4.1	Horizontal scan technique.....	29
3.9.4.2	Vertical scan technique.....	29
3.9.5	Coordinated Scan Technique.....	29
3.9.6	Profiling Modules Based IDS.....	30
3.10	Data Mining	31
3.11	Data Preprocessing.....	31
3.12	Preprocessing With IDS.....	33
3.13	Information Retrieval	33
3.14	Machine learning	33
3.14.1	Supervised Learning.....	34
3.14.2	Unsupervised Learning.....	34
3.15	Machine Learning Algorithms for IDS.....	34
3.15	Artificial Neural Network.....	35
3.15.1	Advantages and Disadvantages	36
3.15.2	Decision Tree Algorithm.....	37
3.15.3	Advantages and Disadvantages	37
3.15.4	Naive Bayes.....	38
3.15.5	Advantages and Disadvantages	39
3.16	Evaluation	39
CHAPTER FOUR.....		40
4. DATASET FOR IMPLEMENTATION AND EVALUATION THE MODEL.....		40
4.1	Introduction.....	40
4.2	UNSW-NB15 Description	40
4.3	The Nine Categories of Attacks in UNSW-NB15 Dataset.....	41
4.4	UNSW-NB15 Features Description.....	44
CHAPTER FIVE.....		46
5. MODEL METHODOLOGY.....		46
5.1	Introduction.....	46
5.2	Model Architecture	46
5.3	Implementing Tools	49
5.3.1	Visual Studio C#.....	49
5.3.2	Weka.....	49
5.3.3	Microsoft Excel	49
5.4	Dataset Selection.....	49
5.5	Customize Dataset	50
5.6	Dataset Preprocessing	51
5.6.1	TF.IDF	51
5.6.1.1	TF-IDF weighting	52
5.6.1.2	TF.IDF dataset processing	52
5.6.2	Attributes Reduction.....	55
5.7	Machine Learning Classifiers	57
5.7.1	C4.5 Decision Tree	58

5.7.1.1 Pruning.....	60
5.7.1.2 C4.5 (C4.5) algorithm.....	60
5.7.2 Multi-Layer Perceptron (MLP).....	61
5.7.1 Backpropagation and Error Redaction.....	62
5.7.3 Naive Bayes.....	63
5.8 Training and Testing Stages.....	65
5.9 Cross Validation.....	65
CHAPTER SIX	67
6. RESULTS AND DISCUSSION	67
6.1 Introduction.....	67
6.2 Environment.....	67
6.3 IDS Model Performance Evaluation.....	67
6.4 Evaluation Indicators.....	68
6.5 Results.....	70
6.5.1 Results of the First Experiment.....	70
6.5.1.1 Decision tree C4.5 (J48).....	70
6.5.1.2 Multilayer perceptron.....	71
6.5.1.3 Naïve bayes.....	72
6.5.2 Results of the Second Experiment.....	73
6.5.2.1 Decision tree C4.5 (J48).....	73
6.5.2.2 Multilayer perceptron.....	74
6.5.2.3 Naïve bayes.....	75
6.6 Results Discussion.....	76
6.7 Comparisons With Previous Works.....	79
CHAPTER SEVEN	81
7. CONCLUSIONS AND FUTURE WORK	81
7.1 Introduction.....	81
7.2 Conclusion.....	81
REFERENCES	83
CURRICULUM VITAE	89

LIST OF TABLES

Table 3.1 : Advantages and disadvantages in applying ANN with IDS.	37
Table 3.2 : Advantages and disadvantages in applying C4.5 with IDS.	38
Table 3.3 : Advantages and disadvantages in applying NB with IDS.	39
Table 4.1 : Normal and attack categories and their frequencies in dataset.	43
Table 4.2 : Features description of UNSW-NB15	44
Table 5.1 : Class name and relevant features	56
Table 5.2 : The new set of UNSW-NB15 dataset.	57
Table 5.3 : J48 algorithm setting by using Weka software.	61
Table 5.4 : MLP algorithm setting by using Weka software.	63
Table 5.5 : NB algorithm setting by using Weka software.	65
Table 6.1 : Representation of confusion matrix.	68
Table 6.2 : Performance evaluation with C4.5 (J48) with multi class datasets.	71
Table 6.3 : Performance evaluation with MLP with multi class datasets.	72
Table 6.4 : Performance evaluation with NB with multi class datasets.	73
Table 6.5 : Performance evaluation with C4.5 (J48) with binary class datasets.	74
Table 6.6 : Performance evaluation with MLP with binary class datasets.	75
Table 6.7 : Performance evaluation with NB with binary class datasets.	75
Table 6.8 : Highest and lowest detection accuracy and the increase in accuracy.	76
Table 6.9 : Comparison between our work and Amar Agrawal et al. with Hossein Gharaee and Hamid Hosseinvand.	80

LIST OF FIGURES

Figure 3.1 : The location of the IDS and its activities.....	14
Figure 3.2 : The types of network attacks	20
Figure 3.3 : Position of NIDS and HIDS.	23
Figure 3.4 : Anomaly misuse sequence detection system.....	28
Figure 3.5 : Misuse anomaly sequence detection system.....	28
Figure 3.6 : Shows a general description of the parallel detection system.	28
Figure 3.7 : Shows description Profiling Modules in intrusion detection system.	30
Figure 3.8 : Perceptron a simple network structure of ANN.	36
Figure 3.9 : Structure of MLP.	36
Figure 5.1 : Model architecture.....	48
Figure 5.2 : Rate of normal records to attack records.	50
Figure 5.3 : TF.IDF processing steps.	54
Figure 5.4 : Pseudo code of C4.5 algorithm.....	60
Figure 6.1 : C4.5 (J48) with multi class segmented dataset of 4000 records.....	71
Figure 6.2 : MLP with multi class segmented dataset of 5000 records.	72
Figure 6.3 : NB with multi class segmented dataset of 50 records.	73
Figure 6.4 : C4.5 (J48) with binary class segmented dataset of 1000 records.....	74
Figure 6.5 : MLP with binary class segmented dataset of 4000 records.....	75
Figure 6.6 : NB with binary class segmented dataset of 1000 records.	76
Figure 6.7 : Comparison between C4.5 and other classifiers with multi class datasets.....	77
Figure 6.8 : Comparison between C4.5 and other classifiers with binary class datasets.....	78
Figure 6.9 : Each classifier with different segmented dataset and the number of detecting classes.....	78

LIST OF ACRONYMS

ML : Machine Learning
MLP : Multilayer Perceptron
NB : Naïve Bayes
FPR : False Positives Rate
IDS : Intrusion Detection System
TPR : True Positive Rate
TCP : Transmission Control Protocol



ABSTRACT

A NOVEL INTRUSION DETECTION MODEL BASED ON TF.IDF AND C4.5 ALGORITHMS

Khaldoon Ali Hammood AWADH

Master, Department of Information Technology

Thesis Supervisor: Asst. Prof. Dr. Ayhan AKBAŞ

November 2017, 90 pages

In recent years, the use of Machine Learning and Data Mining technologies has been very effective in improving performance of Intrusion Detection System (IDS). These techniques have proven effective solution in distinguishing malicious network packets. One of the most important problems that researchers face with is how to transform data into a form that can be handled effectively by Machine Learning Algorithms. In this thesis, we present an IDS model based on decision tree C4.5 algorithm with transforming simulated UNSW-NB15 dataset as a preprocessing operation to convert data types to an efficient and normalized form for machine learning to achieve high detection performance results. It uses term frequency – inverse document frequency (TF.IDF) to evaluate the importance of dataset items. The model has been tested and evaluated with randomly selected 250000 records of the UNSW-NB15 dataset, then dividing it to various segment sizes as 50, 500, 1000, 4000 and 5000 records, each segment size was divided into two subsets of multi and binary class datasets. We have compared the performance of decision tree C4.5 algorithm with Multilayer Perceptron, and Naive Bayes in Weka software. Finally, we compared our model results with existing models. Our proposed method significantly improves the accuracy of classifiers and decreases the incorrectly

detected instances and that have been achieved with 10 folds cross validation. The increase in accuracy reflects the efficiency of transforming dataset with TF.IDF of various segment sizes.

Keywords: Intrusion Detection System (IDS), TF.IDF, Data mining, Machine Learning, Network Security, Cross Validation, decision tree C4.5, MLP, Naive Bayes, Weka.



ÖZET

TF.IDF VE C4.5 ALGORİTMALARINA DAYALI YENİ BİR İZİNSİZ GİRİŞ TESPİT MODELİ

AWADH, Khaldoon Ali Hammood

Yüksek Lisans, Bilgi Teknolojileri Anabilim Dalı

Tez Danışmanı: Yrd. Doç. Dr. Ayhan AKBAŞ

Kasım 2017, 90 sayfa

Son yıllarda, Bilgisayarla Öğrenme ve Veri Madenciliği teknolojileri, IDS (İzinsiz Giriş Tespit Sistemi) performansını arttırmada çok etkili olmaktadır. Bu teknolojiler, zararlı ağ programlarını ayırt etmede kanıtlanmış etkili çözümlere sahiptir. Araştırmacıların sıklıkla karşılaştığı önemli sorunlardan biri de Bilgisayarla Öğrenme Algoritmaları aracılığıyla verileri etkili bir biçimde ele alınacak şekilde nasıl dönüştürüleceğidir. Bu tezde, daha yüksek tespit performansı sonuçları elde etmek amacıyla, bilgisayarla öğrenmede etkili ve normalleştirilmiş bir forma dönüştürmek için bir ön işlem operasyonu olarak, simülasyonu yapılmış UNSW-NB15 veri setini dönüştürmek suretiyle, karar ağacı C4.5 algoritmasına dayalı olarak bir IDS modeli sunmaktayız. Bu model, veri seti kalemlerinin önemini değerlendirmek amacıyla terim frekansı-evrik doküman frekansı (TF.IDF)'ni kullanmaktadır. Model, rasgele seçilen 250.000 UNSW-NB15 veri seti kaydı ile test edilmiş ve değerlendirilmiştir. Daha sonra, 50, 500, 1000, 4000 ve 5000 şeklinde çeşitli segment boyutlarına, her bir segment boyutu ise çoklu ve tekli sınıf veri setlerine ait iki alt sete bölünmüştür. Karar ağacı C4.5 algoritmasının performansını, Weka yazılımında MultilayerPerceptron(MLP) ve NaiveBayes ile karşılaştırdık. Son olarak, model sonuçlarımızı mevcut modellerle karşılaştırdık. Önerdiğimiz yöntem, sınıflayıcıların doğruluğunu anlamlı bir biçimde arttırmış, yanlış olarak tespit edilen

olayları azaltmış olup 10 kat apraz doęrulama elde edilmiřtir. Doęruluktaki artış, eřitli segment boyutlarındaki TF.IDF ile veri setinin etkin bir biimde dnüşürölmesini yansıtmaktadır.

Anahtar kelimeler: IDS (İzinsiz Giriř Tespit Sistemi) , TF.IDF, Veri madencilięi, Bilgisayarlı Öğrenme, Ağ Güvenlięi, apraz Doęrulama, Karar ağacı C4.5, MLP, NaiveBayes, Weka



CHAPTER ONE

INTRODUCTION

1.1 Introduction

Today's networks represent the backbone of every modern-day application such as transportation, healthcare, banking and so on. A significant quantity of information is increasing day by day along the increasing of attacks and threats on networks from interior and exterior include the purpose of gaining illegal access permission to disclose sensitive data or resources. Intrusion detection system is a monitoring, detection and analyzing opposition traffic behavior in the network system. Intrusion is described as a work violates the computer or network confidentiality, integrity or availability. These violations represent a direct threat to the security of the network, for example, unauthorized monitoring, unauthorized system misusing, unauthorized modification or deletion of information, denial of services or etc. Last thirty years there are a lot of researches done in about IDS by the use of different technique, however, no one of these tactics is universally dependable because they did not reach to the satisfactory result with a hundred percent accuracy intrusion discovery. Several techniques have been implemented in building IDSs for different network types such as host and network IDS as well as and different IDS techniques such as anomaly and misuse IDS techniques proved the efficiency of malicious activity detection. Nevertheless, cybercrime is still considered in its infancy and a lot of things can be exploited from the development of other recently sciences [1]. Among all these approaches we intend to use data mining and machine learning algorithms together with simulation dataset to find the superior results of detection. We will examine the UNSW-NB15 intrusion dataset within our research, as well as real time captured dataset. This data set is a hybrid of intrusion data collected from real modern normal and abnormal activities (contemporary

synthesized attack activities) of the network traffic [2]. This dataset is more modern and more efficient than KDD98, KDDCUP99 and NSLKDD which are common, older and less feature datasets because they were generated before more than a decant. UNSW-NB15 is present day and efficient, but more Complex due to the fact of its features is larger and with more types of attacks. Using supervised training schema of Machine Learning Algorithms is a proper strategy to solve intrusion detection problems because we process a huge volume of data, it is greater suitable for discovering malicious traffic and has capability for instruction and discovering the anomaly attacks. In our approach we will build a suitable preprocessing model for machine ML algorithms. The converted dataset will be experimented and evaluated by using Decision Tree, Artificial Neural Networks and Naïve Bayes algorithms in classification mode, which are one of the known algorithms in applicable and powerful with IDSs.

1.2 Motivation

IDSs refer to securing the network system from the intruders' activities. On the opposite it means accessing the network is restricted by centralizing control what is called network administrator. Today's IDSs will appear as a need in various communicating fields with the development of communication networks forced on finding active malicious detection techniques and analyzing the efficiency in cyber systems implementation and pointing the limitations, strengths and weaknesses. The motivation is we have noted that the most of the previous researches used old fashion simulated dataset like KDD99 dataset for building IDSs to face continuously renewed attacks. However, very few researches carried out with UNSW-NB15 dataset. Our approach is an attempt to explore the potential of developing IDS with UNSW-NB15 dataset by using data mining, information retrieval and ML techniques with an important factor consideration of the system should provide accurate detection results. IDSs require more deal of human effort for the purpose of improving the effectiveness of performance. In this research, we continue to progress towards the goal in building an IDS model, including results that encounter the evolution of intruders' behavior.

1.3 Objective and Questions of the Research

The main objective of this work is to build an optimized intrusion detection model by using data mining techniques and ML through builds a suitable dataset processing approach. Because of the complexity of the recent UNSW-NB15 dataset, unconventional methods must be found for converting inconsistent data format into a form that enables the ML to differentiate between normal and attack types effectively. The converted dataset will be evaluated by using different ML techniques in the classification stage of distinguishing between normal and anomaly activities in the network system and analyzing the results of this model. In designing this model some questions will be answered and discussed in detail:

1. What are the threats and malicious behavior from intruders against network system?
2. What are the IDS functions and the efficient techniques have been used in intrusion detection?
3. What are the proper techniques of data mining can be used in data preprocessing stage with a recent simulated UNSW-NB15 dataset?
4. What is the proper algorithm of ML technique can be used in detecting anomaly behavior?
5. Is multi class or binary class classification approach better with our preprocessing model?

1.4 Methodology

In order to perform this research, some procedures of implementation and experimentation must be applied sequentially as bellow:

1. Studying and analyzing the UNSW-NB15 dataset and its data types identify the positive and negative characteristics.
2. Training and testing sets will be selected randomly as a subset of the UNSW-NB15 dataset.
3. Customize two copies of training and testing sets, one for multi class and another for binary class detection.

4. Apply a fixed number division to a selected subset of a number of sections, each of which represents a document. For example, divide overall subset into segments of 200 records.
5. Customize some copies of training and testing sets for different segment size.
6. TF-IDF text mining technique will be applied to convert the structured UNSW-NB15 subset to an effective form.
7. Applying attributes selection to reduce the size of dataset.
8. Multi class dataset with Different number of segment size will be evaluated with Multi Layer Perceptron, Decision Tree and Naïve Bayes classification algorithms.
9. A binary class dataset with Different number of segment size will be evaluated with the same classification algorithms.
10. The results will be analyzed and conclusions will be stated.

1.5 Contributions

This research contributes to important insights in an intrusion detection realm as mentioned in the following points:

1. A novel approach of using TF.IDF text mining method to convert structured UNSW-NB15 dataset as a data preprocessing stage to a suitable and normalized dataset for ML algorithms.
2. Experiment multi class subset with different segment sizes will be evaluated with Multi Layer Perceptron, Decision Tree and Naïve Bayes classification algorithms.
3. Experiment binary class subset with different segment sizes will be evaluated with the same classification algorithms.
4. The results from steps number two and three will be compared.

As summarized from the literature these methods have advantages over the other methods in anomaly intrusion detection.

1.6 Organization of the Thesis

The remainder is organized into six chapters further as follows:

Chapter 2 introduces the literature reviews and related works with IDSs and data mining and ML have been used. Chapter 3 highlights general principles of network security, background of IDS and data mining techniques. Chapter 4 describes the UNSW-NB15 dataset. Chapter 5 describes the implementation of our model with TF.IDF approach, multi class subset, and binary class subset evaluating them with Multi Layer Perceptron, Decision Tree and Naïve Bayes classification algorithms. Chapter 6 presents the results of our model and compares the results of different experiments. Chapter 7 reviews the concluding of implementing our model and future studies.

CHAPTER TWO

LITERATURE SURVEY

2.1 Introduction

The intrusion detection systems (IDSs) have been modeled through using single or hybrid machine learning (ML) algorithms with data mining techniques along various types of simulation benchmark datasets. These algorithms and techniques had been promoted and analyzed for improving the performance; this was in sync with the study and analysis of the benchmark datasets. Correct prediction rate and error prediction rate are the main indicators of IDS performance efficiency. The most well known and optimal researches will be presented in this chapter to obtain the related work with our research. The purpose of the presentation is to find out the techniques used previously and how to utilize from the results to find an effective system.

2.2 Related Works

Tahir Mehmood and Helmi B Md Rais [3], presented a comparison between various supervised ML algorithms with using simulation benchmark KDD99_10% dataset. The comparison focused on detecting anomaly data intrusion. ML algorithms such as support vector machine, naive Bayes, J.48, and decision table were experiment for anomaly detection. All algorithms were experimented using Weka 3.7 application tools environment. The main idea of comparison in this research is classifying the KDD99 data set into four main attack classes in addition to normal class. Each one of ML algorithms was applied with one class in each time. The comparison showed that accuracy of J.48 decision tree is higher among all other algorithms and has low misclassification rate. The reason of superiority in J.48 decision tree is it yields a good result in the existence of the redundant features.

Finally, Overall results review there is not an individual algorithm has a high detection rate for all KDD99 dataset classes. In this comparison performance was evaluated by three measures are true positive rate, false positive rate, and precision.

Nour Moustafa and Jill Slay [2], presented a modern benchmark UNSW-NB15 dataset for research community and evaluated it with exist common benchmark data set KDDCUP99, that was generated more than a decade ago. The reason of founding anew dataset was the existing benchmark KDDCUP99 and other datasets are not representing the universal representation of the modern direction of malicious network traffic and new attack scenarios. The existing IDS simulation datasets such as KDD98, KDDCUP99 and NSLKDD were limited by a few numbers of attacks and information of packet behaviors. UNSW-NB15 was created by establishing the synthetic environment with IXIA tool and it contains additional features than those exist in KDDCUP99. UNSW-NB15 is a comprehensive dataset for IDS by had supplemental contemporary data of malicious attacks and it is considered helpful as a modern benchmark dataset for researchers.

Amar Agrawal et al. [4], proposed a hybrid an IDS including two stages first one, misuse detection model by using a Binary Tree Classifier for detecting only known attacks and the second stage for anomaly detection model based on SVM Classifier which detected patterns that deviate from normal behavior. The Binary Tree consists of several classifiers specialized in detecting attacks with high accuracy. Combining Binary Tree classifier and specialized classifiers will increase accuracy of the misuse detection model. The proposed hybrid IDS has been experimented and evaluated using benchmark UNSW-NB15, KDD Cup 99 and NSL-KDD dataset. Each classifier tested with number of times according to the number of classes in that dataset to analyze and study the dataset specifications. The model built by choosing the best classifier algorithm for each class. This model was implemented by using WEKA API with Java programming. The proposed model was compared with ZeroR, Naive Bayes and J48 algorithms. The highest accuracy was obtained with decision tree J48 87.52% with UNSW-NB15 dataset, 98.47% with NSL-KDD dataset and 98.95% with KDD Cup 99 while the proposed mode results 88.55% with UNSW-NB15 dataset, 98.80% with NSL-KDD dataset and 99.92% with KDD Cup 99.

Hossein Gharaee and Hamid Hosseinvand [5], presented an intrusion detection approach. They implemented a new feature selection model to reduce the dimension of UNSW-NB15 and KDD CUP 99 datasets which they were used in these experiments through using the discriminating properties of Genetic algorithm. They used support vector machine as detection algorithm. The combination of genetic algorithm for feature selection and SVM for detection returned a good discovering accuracy. They used all attack types in the KDD CUP 99 dataset, but the only selected 6 from 9 attacks of the UNSW-NB15 dataset, in addition to normal class then it has 10 classes. This work selected the active attributes that related to each class in both datasets by using genetic algorithm. Accuracy, TPR and FPR were obtained by using SVM individually with each class. Experimental results were 99.05 % for normal traffic, 99.95% of DOS attack, 99.06% of PROBE attack, 98.25% of R2L attack and 100% of U2R attack with KDD CUP 99. In addition the results of UNSW-NB15 Dataset were 97.45 % with Normal class, 96.39 % Fuzzers attack, 91.55 % Reconnaissance attack, 99.45 % Shellcode attack, 91.24 % DoS attack, 79.19 % Exploits attack and 97.51 % Generic attack.

Nour Moustafa and Jill Slay [6], compared the efficiency of the benchmark data set UNSW-NB15 and KDD99 to remove what is not associated to or unimportant features in recognizing between normal and malicious data records. Author developed a feature selection technique to remove inappropriate features to decrease the computational time in intrusion detection system. Naive Bayes algorithm and expectation maximization clustering algorithm have been used to estimate the accuracy and false alarm rate. Outcomes showed that the preciseness of the KDD99 was greater than the UNSW-NB 15, and the false alarm rate concerning to the KDD99 dataset is lower than UNSW-NB15. However, the evaluation criteria of the UNSW-NB15 dataset show that the decision engine algorithms were not able to detect several records categories, because the similarities in record values.

Tanya Garg and Surinder Singh Khurana [7], presented a comparative performance of different classification algorithms by using benchmark NSL-KDD dataset. These classifiers were evaluated by WEKA Knowledge Analysis environment with using 41 features form dataset. Approximately 94,000 records from whole NSL-KDD dataset was specified for training phase and over 48,000 records for the testing dataset. Garrett's Ranking approach was applied to rank

dataset classifiers. They tested 45 classifiers and evaluated them by ROC area, FPR, accuracy, Kappa, Mean Absolute error, Recall, Precision and training time. Finally the results ranked in a table. The results of evaluation classifiers show Equal importance was given to all the classification algorithms. Using another ranking method can change the ranking with the same evaluation vectors.

Vrushali D. Mane and S.N. Pawar [8], proposed a model of anomaly IDS based on ANN algorithm. This paper targets to identify attacks with the support of supervised and back propagation ANN algorithm. In this research experimented only 17 attributes of 41 the whole attributes in the KDD 99 dataset. Only 10% of the records from the KDD 99 dataset were used. All classifications were applied on the binary of attack and normal basis. The KDD 99 training set consists of more than 4,900,000 instances which represent connection packets, each of which is labeled as normal or attack. ANN takes long time in days for training and testing if entire KDD dataset is inputted to it. So using 10% is the best choice to improve the performance and accuracy. The results obtained performances of IDS with reducing features increases the detection accuracy until (98.0%) and training phase and testing phase take less time. Also, normalization has been generally increased the results of testing it is compared with results without normalization. In addition, feature reduction approach improves the efficiency and decreasing the false alarm rate.

Datta H.Deshmukh et al. [9], depicted a comparative performance of different classification algorithms of Naive Bayes, AD Tree and NB Tree by using benchmark NSL-KDD 99 dataset. Weka as data mining tool was used for classification analyzing the results. Firstly, they described the classification mechanism of each algorithm. The dataset was processed by feature selection which removes the redundant or irrelevant attributes from the training and testing dataset to prevent reduction in classification accuracy and unnecessary incrementing in computational costs, min-max normalization gives all features an equal weight between minimum and maximum value in each attribute and discretization which is the process of converting the continuous domain of a feature into a nominal domain with a finite number of values. The results show the proposed model improved the accuracy of classifiers detection with higher TP Rate of all the classifiers. Finally they concluded Naive Bayes classifier is efficient because it is simple, robust and effective as

compared to other classifiers tested in this paper. They tested classifiers and evaluated them by ROC area, FPR, accuracy, Recall, Precision and training time.

Varsha Singh and Shubha Puthran [10], viewed various data mining methods of classification and clustering to improve the detection rate and reduce the error detection rate. They studied different classification algorithms as individually and hybrid with clustering algorithms with different data sets such as NSL_KDD, KDD99, GureKDD and Kyoto 2006+ to determine the suitable classifier with IDS. They concluded the Decision Tree is better in prediction than another tested classifiers. Pruning Decision Tree C4.5 accuracy was 92% with KDD 99 and it resulted 98.45% with NSL_KDD. For the hybrid models Naive Bayes with K-means the detection rate was 92.12%. Therefore, Decision Tree C4.5 with K-means the detection rate was 99.6%. They mentioned that the Hybrid Algorithm Classification and Clustering gives efficient results of higher detection accuracy with reducing the false positive and false negative rate.

Yanjie Zhao [11], discussed the statistical methods to normalize the dataset attributes before implementing the classification. The Infrastructure of training set or testing set is a pattern of data constricted attributes. The classification mechanism should be compatible with attribute structure to obtain effective detection. Authors put two questions before applying attribute normalization, first one is attribute normalization essential to detection rate, and second which method of attribute normalization is the best. For these two questions, four schemes of attribute normalization Mean range (0, 1), Statistical normalization, Ordinal normalization and Frequency normalization were experimented with three classifiers SVM, k-NN and PCA. Only 34 numeric features of the benchmark KDD Cup 1999 dataset were used for evaluation. The experimental results depicted the importance of attribute normalization in improving detection performance.

Dipali Gangadhar Mogal et al. [12], proposed a new model of IDS based on Central Points of attribute values with a prior algorithm for selecting high ranked features and removing irrelevant features which causes high FAR, in the preprocessing stage. Then Logistic Regression and Naive Bayes algorithms were used. The evaluation of this model was done by using the KDD and a new benchmark UNSW-NB15 datasets. KDD dataset is the benchmark a decade ago for IDS and the author noted it is becoming outdated because it cannot cover modern

normal and malicious attack activity behaviors. The proposed system divided each dataset to training and testing parts after preprocessing was implemented to prepare the datasets for ML algorithms. The experimental results depict that, the preprocessing reduced the processing time and improved the evaluation of the Naive Bayes algorithm. The system has improved the detection accuracy and decrease the FAR by reducing the processing time.

Seyyed Mohammad Hossein Dadgar et al. [13], presented a novel text mining approach for classifying news text. The proposed model implements a collection of data news processing in sequence. First step was text preprocessing because the news text is unstructured text consists of useful and useful data because data news documents were collected from different news resources and should be cleaned from useless items such as punctuations, exclamations, semicolons and dates, etc. Second step, was text transformation from upper case lower case to avoid homologous word appearing in the document. The third step was tokenizing by separating words from their sentences. Forth step, the weight of each word is calculated with TF-IDF equation which is one of the most famous text mining algorithms. After preparing data into a structured form as a dataset support vector machine (SVM) algorithm was implemented to group news documents. The proposed system used the BBC dataset and 5 groups of 20Newsgroup dataset, all groups were evaluated. The classification precision was 97.48% and 94.93% for the BBC and 20Newsgroup datasets respectively.

2.3 Summary

From the literature review, it is found that many researchers employed data mining and machine learning algorithms to solve intrusion detection problem. Different strategies had been used by placement single machine learning algorithm or using hybrid machine learning techniques. Different ML classifiers can be used in modeling IDS such as SVM, MLP, J48 and Naive Bayes algorithms. Implementing any classifier algorithm with IDS has advantages and disadvantages. In papers above, the researchers identified the most important points that lead to the weakness or increase efficiency of the model supposed to be built. Some researchers suggested new methodologies to raise the performance of intrusion system. Various types of data set or combining more than one data set was such as benchmark UNSW-NB15,

KDD Cup 99 and NSL-KDD dataset evaluate Intrusion Detection model, however the benchmark NSL KDD 1999 Data set is the most commonly use in researches because it is simplify the comparison task between previously proposed works but the new UNSW-NB15 dataset is more efficient and simulate the modern normal and malicious behaviors. From the rates results of implementation algorithms, we found that some machine learning algorithms gave the best results and they have proved that they are the best in term of intrusion detection system such as MLP, J48 and Naive Bayes algorithms. They tested classifiers performance and evaluated them by FPR, accuracy, Recall, Precision and training time.



CHAPTER THREE

BACKGROUND OF IDS AND DATA MINING

3.1 Introduction

In this chapter, general specification and basic principles of intrusion detection system components will be reviewed. This chapter starts with a general description of network security attacks and the main components of IDS Model. Define the IDS and explain the various types of intrusion detection techniques. Section three reviews the detection components of IDS by using data mining and machine learning, and then the machine learning techniques and its properties will be reviewed. Before going into deep details about the types and techniques of the IDS, general definitions and concepts will be mentioned.

3.2 Network Security

Network security refers to the protecting network systems from outside and inside intrusion. This title refers to searching in finding the mechanisms and methods in the area of designing a protected network and detection system from anomaly attacks as well as known attacks in real time. The limitation to design a full secure network system is the regeneration and increasing sophistication in the methods of attack and techniques which lead to increase the size of identify intrusion data and the high cost of computation [14]. Intrusions targets to harm security in computer or network system by compromising one or more of the confidentiality, availability or integrity the main principles of secure system, they are usually represented as a triangle of system security. Before entering in this chapter and its details, take some important definitions about the main principles of intrusion detection system such as intruder, intrusion, firewall and network activities.

3.2.1 Intruder

The intruder is unauthorized person, party or device attempt to access the network system or computer system from inside or outside to sabotage the system or to exploit sensitive information [14-16]. The intruders can be recognized into three classes:

1) Masquerader: Unauthorized person attempts to use the network system and its information as a legitimate user. The masquerader is likely to start his attack from outside the system.

2) Misfeasor: A legitimate person attempts to access data, programs, or resources, but his authentication doesn't have a privilege to use them. The misfeasor is likely to start his attack from inside the system.

3) Clandestine user: A person who exploits his privileges by supervises the control of the system to gain privet users. The clandestine is likely to start his attack from outside or inside the system.

3.2.2 Intrusion

Intrusion is the act of intruder which leads to gain unauthorized network information or sabotage the system or its information [17]. Intrusion refers to attack; this term contains two concepts intrusion indicates to the attack from outside the network system or computer system and misuse describes attacks from inside the network system or computer system but they perform the same damage [14]. Figure 3.1 shows the location of the IDS and its activity against inside and outside malicious activities.

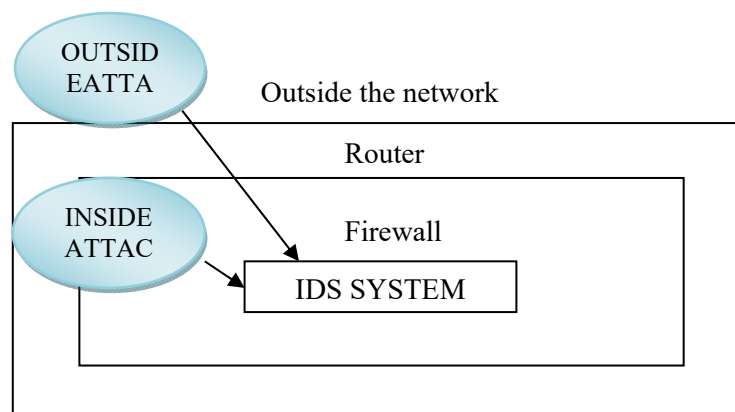


Figure 3.1: The location of the IDS and its activities.

3.2.3 Confidentiality

It refers to securing private information from restrictions or disclosing from unauthorized parties [18]. The goal of confidentiality protects personal privacy and proprietary data in computer system or network system from intentional or accidental unauthorized attempts to read or disclose data [15, 19].

3.2.4 Integrity

Assuring only authorized parties can make changes on data such as modification, deletion, creation or distraction. It is divided into two types: data integrity and system integrity, the goal of both of them, should keep the legitimate users of computer system or network system free from unauthorized manipulation from intentional or accidental unauthorized attempts to alter or destroy the data and system information or configuration [15, 19].

3.2.5 Availability

Assuring authorized parties able to access quickly and completely at any time to the system for obtaining information or services [18]. The goal of availability protects legitimate computer system or network system users from intentional or accidental unauthorized attempts to block data or system services. There are different authorized actions against availability such as deleting data, hide data, interruption of service or any action cause unavailability of service what is known as denial of service [15, 19].

3.2.6 Firewall

It is a hardware or software or combination of them uses rules and filters to secure network system from possible malicious activities as the first line of defense against external network [20]. It is placed between the network and the outside world that represents other networks. The firewall must have some properties such as all network traffic pass through the firewall, only authorized packets pass to the network and it is protected from penetration or compromise. Therefore firewalls individually do not provide full network protection from inside and outside. Intrusion detection

system is a complementary part of the network protection system exactly when intruders starts its attack from inside the network or succeed in passing the firewall from outside the network [18, 21]. At first sight IDS and firewall look similar but they have some different concepts as [22]:

1. A firewall stops attempts intrusion from happening by recognizing the unauthorized or suspicious traffics.
2. A firewall represents a gate and the traffic going through it in each direction must pass by checking the metadata packets.
3. The security policy of firewall orders authorized traffic packets to which direction will pass.
4. Firewalls restrict passing of incoming and outgoing network traffic packets in private networks.
5. The firewall restricts and limits the access between connected networks to prevent intrusion, but does not detect attacks incoming from inside the network, the intruder may impersonate as an authorized person to enter the network and exploit the resources.
6. An IDS monitors and evaluates the traffic packets and signals an alarm against suspicious or intrusion packets.
7. An IDS watches traffic packets that incoming and outgoing form and to the network.
8. The network IDS protects the system from the intruder's malicious packets that are designed to be able to bypass the firewall's filters and rules.

3.3 Network Activities

Networks are a vital and important part of all life facilities. Attacks against those network systems could come from the inside or outside the network, from intruders or authorized attackers, in all possibilities of attack against networks, each attack is sent as a network packet, so the network activities provide a good resource to be analyzed to detect possible intrusion. Network packets contain data and a variety of characteristics that can be analyzed to determine intrusion information such as source of address/port, destination of address/port, the packet size, the communication protocol name and the content of the packets. An example for Denial-of-Service (DOS) attack, an intruder could send a huge number of individuals

TCP connection requests to overload the limitation of the host, in another side the reaction of the host after reaches to the threshold of limitation, the host will deny any incoming TCP connection request. These packets represent the abnormal behavior or malicious packets that must be prevented from accessing the network. The characteristics of packets that cause Denial-of-Service (DoS) attack can be recorded and analyzed by intrusion detection system to prevent such as this attack happening in the future [19].

3.4 Network Vulnerabilities Sources

Vulnerabilities are weaknesses in the design flaws, incorrect implementation and poor security management of a network system in other words the weaknesses is existed in the software or hardware polices or procedures of network foundation which is vulnerable and exploitable by intruders. These titles above are the main network vulnerabilities sources [14, 23]:

1. Design flaws: Network vulnerabilities, security suffering from the poor of designing software and hardware, network system; however hardware considered low more vulnerable than software. Poor of designing software classified as the common source of network vulnerabilities because there are many reasons for example the complexity and difficulty of programming even for professionals, neglecting audit and examination, misunderstanding concepts of developing a secure system, the spreading of entrusted software by small in-house producers.

2. Incorrect implementation: Another important source of vulnerabilities in network System is incorrect implementation. It is the compatibility of combining two or more hardware interface device or software applications in network system. The poor implementation is happening when adding or replacing an interface with the carelessness of the specifications or the configuration of the network system. Such vulnerabilities results from poor of experience, insufficient training or deliberate acts of sabotage.

3. Poor security management: Use of improper technique, security policies and evaluation of the effectiveness of policies, controls or monitoring from the network administrator who is the responsible for administrative network security process as defenses from network vulnerabilities. Security management also involves to constraint the access to the network information, network documentation and

resources by using firewalls and reliable Strong encryption system to secure the network users from attacks against terms of integrity, confidentiality and availability. Poor security management is a result of poor administration over great challenges of network vulnerabilities or it means that the administrator failed in having strong and secure network system.

3.4 A History of Intrusion Detection System

The point of starting the history of intrusion detection is back to 1980, the first one proposed by James Anderson in his seminal paper which prepared for a government organization. He proposed in his paper to monitor and analyze the traffic packets to, understand user behavior to prevent security threats also he provided the foundation of the intrusion detection system model. His work represents the start of host based intrusion detection (HIDS) and intrusion detection and prevention system in general. In 1983, Dr. Dorothy Denning, joined in a government project to analyze government mainframe user behavior and create a profile of their activities. One year later, he developed the first model for intrusion detection system. In 1988, the first Internet worm had been released which disabled Sun and VAX workstations. In the same year, three Intrusion Detection Systems (IDSs) was created. In 1990, Heberlein was the first developer of Network Security Monitor (NSM), after that known as a network IDS when analyzed massive amounts of information traffic to detect suspicious network user's behavior. His work generated a new interested security field for investment, and he introduced the first schema of hybrid intrusion detection. Host based intrusion detection was also developed by SAIC called Computer Misuse Detection System (CMDS). In this year the Cryptologic Support Center of United States Air Force Submitted Automated Security Measurement System (ASIM) to monitor network traffic. ASIM was the first system had been developed by merging both of hardware and software tools to build network IDS. In 1991, the scalability, maintainability and efficiency as vectors of IDSs were founded in intrusion detection ID researches. Two concepts were brought in this year Network Anomaly Detection and Intrusion Reporter (NADIR) and Distributed Intrusion Detection System (DIDS) these systems were able to detect attacks against a set of hosts by collecting and analyzing the data from multiple hosts. In 1994 Mark Crosbie and Eugene Spafford proposed autonomous agents and the NetRanger produced as the first commercial

network IDS product. RealSecure a network intrusion detection system had been developed by ISS for marketing. In 1998, Ross Anderson and Abida Khattak proposed information retrieval technique in a system of intrusion detection. Wenke Lee and Salvatore Stolfo used data mining techniques to construct automatically intrusion detection model system. In 1999, a mobile agent based on intrusion detection system (IDS) was proposed. In the same year, the open source intrusion detection system called Snort system version 1.0 have been released. In 2000, the intrusion detection system as an information security term was extended to include wireless ad hoc network [19, 24].

3.5 Types of Attacks against Network Security

There are many different types of attacks that challenge network security. These attacks are increased by new attacks are constantly being born because of the continuous development and expansion of network systems. Network attacks activities cause impairment in network performance, disclosure of network information, viruses, spyware on network users, and control the network sources, etc. the main classification of main types of network attacks can be categorized into two types as shown in Figure 3.2 below [14-16, 19]:

Active attacks: This type of attack is easier to detect, but its effects are great on the network and network users. As a result, they can be easier to detect, but at the same time they can be much more devastating. Examples of this attack a denial of service (DoS) attack, spoofing attacks or man in the middle attack

1. Denial of Service attack: It is preventing the legitimate users from accessing the network system information services or network system resources services. There is a limitation of providing services in each network system. The intruder sends multiple time excessive messages requesting services to the network system of authenticate requests which are invalid return addresses, cause preventing legitimate users from the service and blocking the service from working. In this type of attack the intruder can prevent the user from accessing websites by floods the web site server by overloading requests, email sending a lot of spam email messages, online accounts by denial of service attacks the bank web site server or etc.

2. Spoofing attack: In spoofing attack the intruder pretends he is an authorized person in network system. For example, the third party C sends a request to the

network A and they pretend they are B as authenticated person. This type can be performed in different scenarios like spoofing email when an intruder sends an email with pretending it is from another one or spoofing packet by sending network packets pretending the address is from another one. Another example is the DNS spoofing.

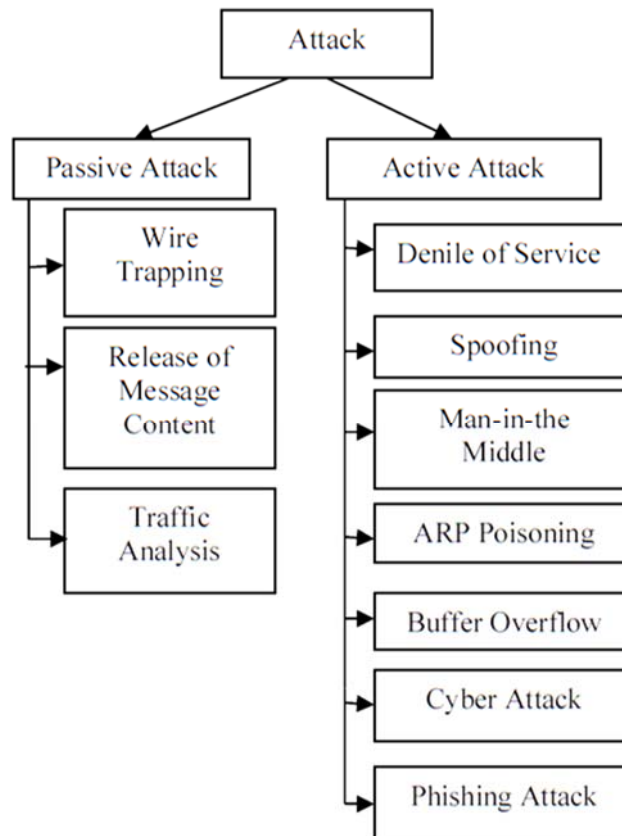


Figure 3.2: The types of network attacks [17].

3. The man in the middle attack: In this attack, the attacker take place between two network users or between two network systems and continuously monitoring the communication between them such as the third party C takes place between A and B, C pretends B when communicate with A and pretends A when communicate with B. The results of this attack, the connections between two parties are disclosures and the messages are relayed. This attack can be used as modification attack through receiving a legitimate message from the source and make insertion, deletion or alteration and reply message to the target.

4. ARP poisoning attack: In this type the intruder sends spoofed Address Resolution Protocol (ARP) messages on to a network system. This attack causes modification or stops the traffic. The intruder changes the Media Access Control

(MAC) address of the network interface by fabricating the ARP by insertion, deletion, alteration or creation and reply packets. Because Address Resolution Protocol has been changed, the target network cannot distinguish it or detect it as an attack.

5. Buffer overflow attack: This attack is a special case of the penetration of memory. It occurs when a process attempts to write data to a fixed size, location of memory, the buffer, starts saving the data overload, because the buffer contains a fixed amount of blocks, the extra data overwrite the data in main memory addresses of a network system, that causes breakdown the network system.

6. Cyber Attack: This type of attack targets the information, it represents any aggressive behavior from a person or machine against a network system to gain private network users information or network information system. An example of this attack, a deceptive site is designed by the attacker, who claims it is the victim's bank, and asks him some personal information, confirms the use of the site by the user's name and password to steal the victim's information. Cross site scripting technique is a one of attack methods by injecting a malicious Java code in a URL internet browser that able to change the accessing to the malicious site instead of accessing to the real site.

7. Phishing attack: This attack can be done by individual intruder or groups of intruders. They target the very sensitive private information such as passwords, address, credit card details.

Passive attack: When an intruder intercepts network traffic through the network directly or remotely. As a result, they can be more difficult to detect, but at the same time they can be less devastating. Passive attack can be done by one of these techniques traffic analysis, Eavesdropping, and Monitoring.

1. Wiretapping attack: Third party monitors the network traffic between two users or between two networks by using the secret connection to gain information, this type of attack is difficult to be detected because the monitoring has no significant impact on network performance by network users.

2. Release of message content attack: Network traffic includes some secret information. Attacker can exploit these contents by reply or delay the messages.

3. Traffic Analysis attack: Attacker analyzes the incoming and outgoing network traffic to gain information such as location, communication hosts, frequency and length of messages, the date of the message, etc.

3.6 Intrusion Detection System (IDS)

Intrusion detection system definition is a tool purposed for detecting unauthorized activities and alerts the system or network administrator from person or device attempting to compromise a computer system or network system [25]. It is a security device or a security software application, but most forms of the intrusion detection systems today are software application. Intrusion detection system function dedicated to monitoring the abnormal behavior in the network environment to build completely secure system in response to the increasing difficulty of attacks. Services from IDS summarized by monitor identity and produce reports for third parties of abnormal activities to prevent security breaches in the future. All packets that manipulate with network from outside or from inside are captured and scanned by the IDS to detect suspicious traffic in different techniques, exactly which has the sufficient ability to cross the firewall's filters. There are different techniques can be used for detection such as statistical algorithms, specification algorithms, immunity algorithms and machine learning algorithms. There are three types of Intrusion detection system network based intrusion detection system (NIDS), host based intrusion detection system (HIDS) and Hybrid of HIDS and NIDS [25].

3.6.1 Network Intrusion Detection System (NIDS)

Network intrusion detection systems observe overall the network and work by monitoring the exchanging network packets between the network and other networks or computers [26]. The system of detecting malicious examining network packet traffic from outside the network by matching with its own intrusion database to build the decision of generating alert, in the same time omitting normal packets with superintendent limitation. NIDS does not eliminate the use of fundamentals network protection fundamentals such as firewalls, encryption, and methods of verification and authentication, but NIDS is complementary to them, Figure 3.3 shows general

description of NIDS. The best example of network based IDS is the SNORT, it is a widespread software for network security.

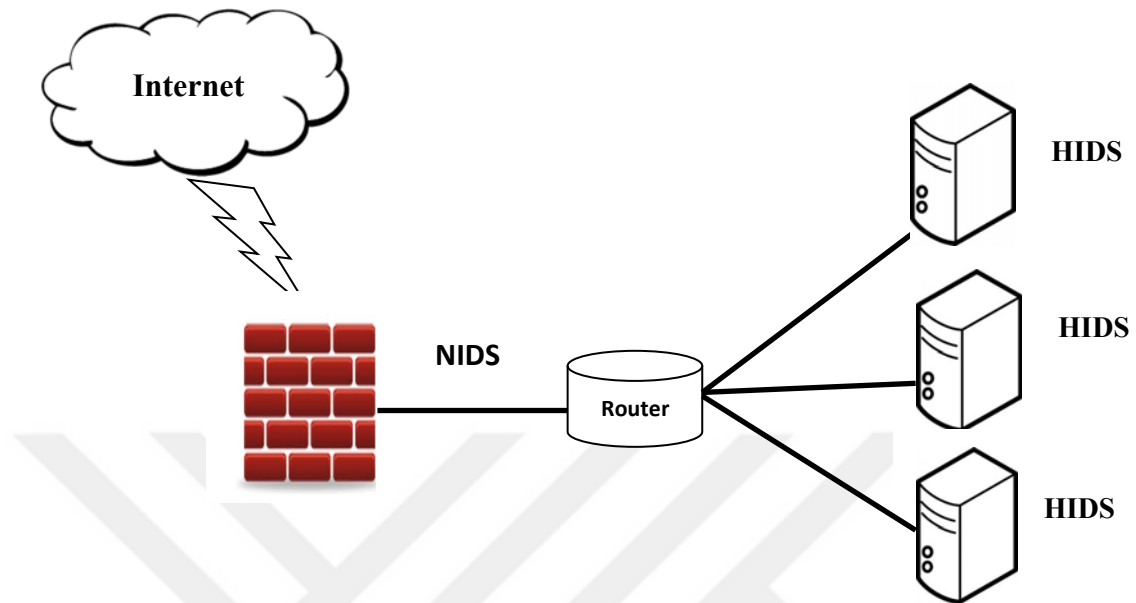


Figure 3.3: Position of NIDS and HIDS.

The management of NIDS is hidden at the network for the purpose of preventing intruder from ability to discover where is located at the network [27]. Packet is a single logical packet segment in the network. As we mentioned, packets are under surveillance that going inside the network or out of the network. Each network should has one or more software program for collecting network packets called sensor, which should is copied into a router or a switch. IDS cannot read packets before transform them into a readable form by sensors pretreatment. Sensor software program can be copied into a router or switch [25, 26].the restriction of NIDS is Attributed to the location among network and it has to observe all network packets.

3.6.2 Host Intrusion Detection Systems (HIDS)

Host based IDS typically collects and monitors exchanging packets inside a single network (a host) or a device represent a component in the network. It located as a software application and it does not monitor network entirely. Known skills concerning HIDSs contain log analysis, event correlation, integrity checking, policy enforcement, root kit detection, and alerting. Host intrusion detection system can

consider the status of the system such as its information in the RAM, file system or file record, wherefore monitors interior anything or anyone tries to treat the system [14]. The system of malicious detection exams interior packet with its own intrusion database to decide raises alert to the administrator or not, in the same time omissions normal behavior.

3.6.3 Hybrid of HIDS and NIDS

According to the ongoing developments of intrusions against IDSs led up to find a hybrid intrusion detection system to confront the challenges of modern attack. Combining host and network IDS is performed perform keep made by means of compile IDS sensor locations and merge the report attacks to increase the qualification of security system in a part of network or whole network [25].

3.7 General Intrusion Detection System Model

There are several models to design modern intrusion detection systems IDSs but most of them matches with the general model that proposed by Dorothy E. Denning in 1987. These models have different detection techniques, however they are similar in collecting and analyzing data. General model includes six main components [19]:

1. Subjects are initiators of user activity such as traffic packets in the single host or network should include both types of normal users and malicious behavior packets.
2. Objects are hardware and software, computer network system resources managed by the detection system, such as files, commands and devices.
3. Audit records are the updating data generated by the network system to perform in response to user's requests or accessing by subjects to objects, for instance, user login, command execution and file access.
4. Profiles are some structures that describe the behavior of the user's activities that categorized previously as subjects in forms of statistical metrics and models and observed activities obviously. Profiles can be generated by the

administrator manually or with expert knowledge or generate it automatically by employing single or multi machine learning algorithms.

5. Anomaly records are alarms triggered to the network administrator when detecting abnormal behaviors from a network user performs are device a device.

6. Actions can be taken according to the activity rules when some conditions are satisfied. Example of actions can include updating profiles, detecting abnormal behavior, detecting suspicious intrusion anomalies and generating alarms report to the network administrator.

3.8 IDS Alarms

The main objective of designing IDS is to output reports of malicious or suspicious activities. Network administrator receives these reports in different forms such as emails, alarm sound or any configured type to alarm the administrator. Alarms caution administrator, but reports includes the malicious or suspicious activity description, usually these reports include timestamp, attack type if it is known and some other information concerning with traffic packet. Reports help administrator in taking decisions about network packets such as prevents true malicious packets or they are just suspicious but they are normal activities (false positive alarm) [14]. The quality of alarms relies on the efficiency of intrusion detection system. There are two active types of alarms in detection systems. The first one is the miss alarms on true attacks and the second one false alarm on normal behavior wastes operator's time and reduces the operator's trust in the IDS[19].

3.9 Intrusion Detection Techniques

IDS's detection techniques include five types misuse technique known as signature technique, an anomaly technique, hybrid technique, scan detection technique and profiling models technique but essential the first two techniques misuse and anomaly represent the main principles of detection technique[19]. Generally networks use these techniques according to the type of network and the type of attack or what is known as malicious network packets and what information was fed into intrusion detection system.

3.9.1 Misuse Technique Based IDS

Misuse technique is an analysis engine in host based IDS and network based IDS, it is known as signature technique. This technique depends on matching cached reciprocal network packet signature together with an existing event pattern in the intrusion detection system. A signature is an identification data associated with the attacks that monitored previously and saved in the IDS recognize as signature databases or rule sets, for example it may contain a segment of virus code. A characteristic that distinguishes this technique is the high regulation and it return high accuracy rates verses known attacks, however, there are some disadvantages with this technique, it does not have ability to discover unknown or new attacks and it cannot exploit attack information in future. New attacks represent the weak point in this technique because it was not found in intrusion detection database. Signature databases or rule sets in misuse technique generally have parts of attack information such as:

1. Unique signature bytes.
2. Operation type.
3. Protocol name.
4. IP port requested.
5. IP addresses.
6. Reaction result, such as allow, deny, alert, log, disconnect.
7. Begin time.
8. End time.
9. Total duration

Often intrusion detection system comes with some of the attack and it is possible to update the signature databases, but increasing number of signature database items increase computation time and with the continual appearance of new types and methods of attack, it becomes more complicated.

3.9.2 Anomaly based IDS

Anomaly technique is an analysis engine for discovering new and unknown attacks in host based IDS and network based IDS, it is known as behavior based IDS. This technique has provided a significant progress in the field of network security

with intrusion detection system because it is able to treat the threats that cannot be solved in the signature technique. Anomaly IDS technique is based on the creation of a profile contains normal and abnormal behavior in a feasible Pattern that that can be compared with network packets. The advantage that distinguishes this technique is the discovering of new and unfamiliar attacks which are not found in the profile and the second advantage is the possibility of updating the profile automatically by anomaly attacks information and it will be a part of existing patterns can be used in detection in the future. However, there are some disadvantages of this technique; it is suffering from a less accuracy than signature technique and it is expensive in terms of computation. Because this technology is promising for the network security, advancement, a large number of researches have been achieved in this field by using different approaches for the purpose of raising the accuracy and efficiency in the detection of IDS. Basically, these approaches are techniques (algorithms) categorized as statistical algorithms, specification algorithms, immunity algorithms and machine learning algorithms. For our research, we will build our proposed system by employing machine learning algorithm artificial neural network for classification to produce the training set which contains normal and abnormal behavior of the simulation network dataset.

3.9.3 Hybrid Based IDS

Detection systems usually use signature technique and an anomaly technique with a presence of drawbacks in both approaches as mentioned above. A hybrid technique was developed from integrating the advantages of both technologies. Running these two systems together can combine the ability of detecting unknown or new attack behavior with the high regulation and high accuracy but it is not always successful because it needs cognitive awareness of how design the intrusion detection system by using both techniques. It divided into three types of hybrid detection [28]:

3.9.3.1 Hybrid anomaly misuse sequence detection system

This model design was succeed in reducing the false alarm rate (FAR) by excluding suspicious alarms in anomaly detection part that was not classified as an

alarm by the misuse detection part [28]. Figure 3.4 shows general description of anomaly misuse sequence detection system.

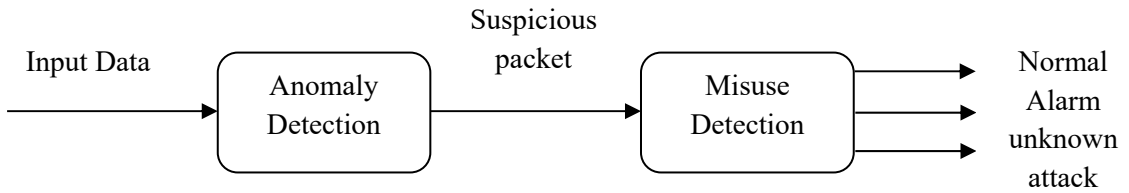


Figure 3.4: Anomaly misuse sequence detection system.

3.9.3.2 Hybrid misuse anomaly sequence detection system

This hybrid model design was succeed in Possess the ability of detecting new attacks by anomaly detection part that overtook from misuse detection part [28]. Figure 3.5 shows general description of misuse anomaly sequence detection system.

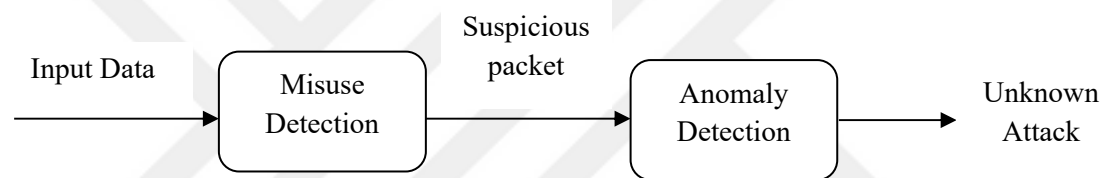


Figure 3.5: Misuse anomaly sequence detection system.

3.9.3.3 Hybrid parallel detection system

It is a complex detection system model, it achieves high detection performance employs signature technique and anomaly techniques in parallel with signature and anomaly network packets are trained in intrusion detection system in parallel at the same time [28]. Figure 3.6 shows general description of parallel detection system.

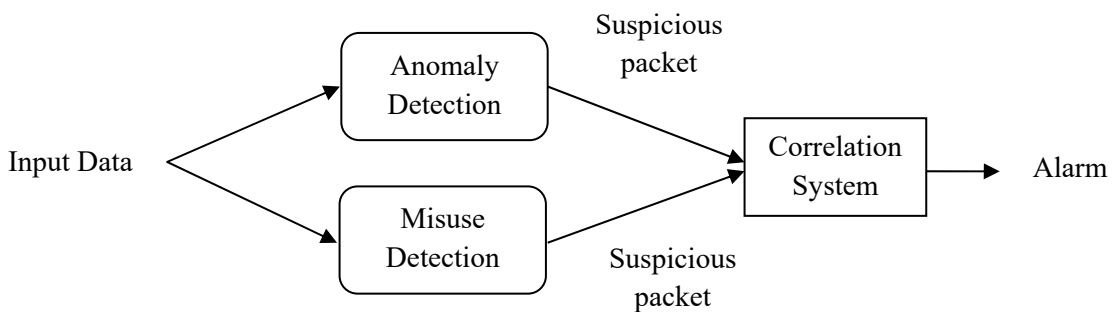


Figure 3.6: Shows a general description of the parallel detection system.

3.9.4 Scan Detection Based IDS

This detection technique works as proactive predictor of attacks and able to deal with network packets in blocks or stream form. It uses real time scanning method against intruders when they try to send scanned pulses to discover the network component structure and services such as destination of internet protocols and the source internet protocols to find the vulnerabilities in network system. Usually intruders send messages to network system ports and analyzing the response, reflection of these ports to determine the access port can be exploited in attacks. These accessible ports provide insufficient information to complete the attack. Scan detection technique generates alerts before the attack happened, therefore it can prevent attackers continuing in its penetration missions, however, this technique practically is suffering from limitation in low accuracy scan detection and high rate of false positive detection. Intruders scan categorized into three technique types horizontal scan technique, vertical scan technique, and coordinated scan technique [28].

3.9.4.1 Horizontal scan technique

Horizontal scan technique represents the common resent technique of attacks. In this technique intruder scans overall ports and services belong to particular network system to find a single or a number of ports can be exploited as a loophole in target to gain insufficient information to complete the mission of attack [28].

3.9.4.2 Vertical scan technique

This technique type on the contrary of horizontal scan technique, intruder scans specific ports and services to find particular group in a single host in network system can be exploited as a loophole to gain information [28].

3.9.5 Coordinated Scan Technique

It is also known as distributed scan, a single intruder attempts to scan specific ports or services by using various internet protocol addresses to find the desired port and host in network system.

3.9.6 Profiling Modules Based IDS

Profiling detection technique provides data mining and machine learning algorithms for generating a profile contains patterns of normal and malicious behavior for detecting anomalous and known attacks. Data mining or machine learning algorithms performs grouping of similar network normal and malicious behavior and find the relationship between them to build a detecting learned pattern. This technique has two problems profiling model, the incremental amount of network traffic and the limitation in detecting the type of behavior by using learned pattern. This model can be represented in four steps, data collection, data preprocessing, generating profiles, and alarm reporting. Figure 3.7 shows description Profiling Modules in intrusion detection system.

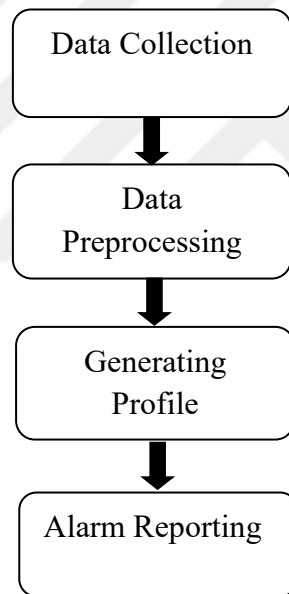


Figure 3.7: Shows description Profiling Modules in intrusion detection system.

This model Collects network traffic data online or offline as in proving the efficiency of classification models. Then choose the effective properties that characterize the network packet than another, this preprocessing operation known as feature selection. In the third step, data mining or machine learning algorithm can classify network traffic data and generating profile behavior, according to associated features, but on a condition that all network packets have the same features. Finally the alarm malicious behavior is reported to the network administrator.

3.10 Data Mining

It is a processing of a large amount of data to extract a useful hidden data. Data mining is a term from a larger term of knowledge discovery in databases (KDD), and it is a stage from data processing stages and represents the link that combines a range of disciplines that deal with data such as information retrieval, statistics, machine learning and pattern recognition [29]. There are four full known fundamentals data mining processing [30]:

1. Association Pattern Mining: It is a task of exploring specific characteristics as a new data by applying a statistical operation on database items individually or jointly.

2. Outlier Detection: It is a task of extracting unusual, useful information from massive structured or unstructured data.

3. Data Clustering: It is a task of grouping the data according to the similarity of data items. The relationships between the data in the same group are used to recognize the new data. The data items in each group are very similar at the same time they are dissimilar than other groups.

4. Data Classification: It is a task of grouping the data in databases according to the value of a particular feature. The relationships between the data in the same group are used to recognize the new data.

3.11 Data Preprocessing

In order to implement the data mining process the data preprocessing is an important multistage preparation phase includes a number of individual steps to qualify data before achieving data mining main processing. The number of steps in data preprocessing depends on the type, size and the description of data. Generally, this phase starts after data collection, below following steps of preprocessing [30]:

Feature extraction: It builds structured data from raw data by analyzing data and abstracts the important features that are relevant to accomplishing the task. Generally the nature of raw data before extraction tends to be incomplete, noisy, and inconsistent, these drawbacks must be removed in the next data processing stage by using data cleaning.

Data cleansing: It is known as data cleaning, it is used to purge the extracted data by treating or dropping records which contains missing, noisy or inconsistent which they may cause breakdown the accuracy. Missing values is, an empty value in a part or entirely attribute of features extracted. Data noise is a random error or variance in a measured variable led to appearing extreme values in the same attribute. Inconsistent in the data is inconsistent data representations and inconsistent use of codes [31].

Data Integration: Data integration is the merging of data from different data resources. Integration must avoid redundancies, inconsistencies and irrelevant attributes in the generated dataset. The correct implementation of data integration can improve the accuracy of dataset [31].

Feature Selection: Also it is known as variables selection, data reduction or attributes selection [14]. The term of features selection is commonly used with machine learning, and generally with data mining applications to describe methods and techniques used to minimize the size of dataset features to a convenient subset of features size [32, 33]. Feature selection elements irrelevant attributes to reduce data set space and transforming data set to another new one to increase accuracy and efficiency by reducing training time [30]. In addition it reduces overfitting of data by reducing redundant data which leads to build decisions depend on noisy data. There is a need of using feature selection to reduce the size of the data, particularly with intrusion detection system simultaneously with the huge increasing in the volume of alternating data packets over the network and the continued appearance of new types of attack. Datasets became huger and that led make detection exhausts computational resources and delay the detection of intrusions [14].

Transformation: It is a data mining process means converting nominal attributes or the combination of letters and numbers by Special mathematical method to numeric form.

Normalization: It is one of data Transformation process types [31]. The benefit of normalization appears when the features are represented in different scales, making comparison between them is impossible. The value given by using this method is restricted by a higher and lower limitation to prevent the impact of the large scale features on the lower scale features [29, 30].

3.12 Preprocessing With IDS

Preprocessing is one of the necessary main elements of building a completed Intrusion detection system. Implementing multistage preprocessing has advantages in intrusion detection system:

1. Reducing the dimensionality of dataset features, and this increases the performance efficiency of the algorithms process and reducing the computation time of classifier, with the consideration of not to influence the effectiveness of the data [14].
2. Redundant, irrelevant or noisy attributes makes detection more difficult and deleting these undesired elements from the dataset increasing the performance of the machine learning algorithms [32].
3. It increases the accuracy rate in machine learning in other words it increases accuracy detection rate in detection system [34].

3.13 Information Retrieval

Information retrieval is defined as “Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies information need from within large collections (usually stored on computers)” [35]. For our work the model will be built by using term frequency-item document frequency (TF-IDF) weighting equation. TF-IDF is a method of information retrieval, word weighting according to the appearing times of a string of text in a document or corpus.

3.14 Machine learning

It is a branch of artificial intelligence [36], uses specific algorithms through programming to allow the computer system the ability to predict decisions in the direction of anomalous behaviors[14]. These programmable algorithms are a sequence of instructions that can convert input data into a new form of ordered list, these lists help computers learn and have experience based on the past learning [33]. It is a methodology of learning the computer system through performing classification, regression or clustering of data to discover a new knowledge. It is like a computer system has the ability to answer new and an unknown question in the

future from past data learning. There are various types of ML algorithms that can be used with returning a practical and effective performance [37]. Basically ML algorithms are measured by calculating the accuracy and efficiency of the realization results [38]. There are two types of ML methods supervised learning and unsupervised learning. Supervised ML uses classification and regression technique in prediction, unsupervised ML uses clustering in prediction. There are special algorithms for each type of species used according to the domain type or the purpose of using ML.

3.14.1 Supervised Learning

In supervised learning the training is supervised by labeled dataset which contains one or more class attributes dedicated only to label each record along the dataset, for example the last two features are the number 48 and 49 represent the labeled features of the UNSW-NB15 dataset and the attribute number 42 in KDD99 Dataset. Using of a supervised machine learning algorithm results high precision and regulation in classification [28]. There is some supervised machine learning algorithms are widespread implemented such as artificial neural network (ANN), decision trees (DT), Naive Bayes (BN) which will be experienced to build our model in the next chapter, in this chapter they will be explained briefly.

3.14.2 Unsupervised Learning

In recognition between supervised learning and unsupervised learning, the training in unsupervised learning has been done by using an unlabeled dataset as data packets from network environment. The output of the training should be grouped into types of normal or abnormal behaviors, according to the properties of the objects as an individual or as a group appears in the dataset attributes to build the predictive model. There are widespread implemented unsupervised machine learning algorithms such as K-means algorithm.

3.15 Machine Learning Algorithms for IDS

ML with IDS can be defined as: it is a software application for monitoring and analyzing malicious activities. In last twenty years, researchers were very interested

in investing modern ML algorithms in IDS to find anomaly malicious activity within the network [14].

Below we show the classifier algorithms that will be implemented in evaluating and building an IDS model. With varying efficiency of the classifier performance depending on the type of application used so it is important to experiment with them to decide which one is the best in accuracy detection for our model. All these algorithms are supervised classifiers and they had been used by many researchers in modeling IDS.

3.15. Artificial Neural Network

Neural network (ANN) is a classifier model for various information processing tasks. The term of neural network has been derived from the biological description of the brain cells of human beings and animals. There are a massive number of interconnected neurons as a neuronal structure. All neurons have a neuron body and at least one input and an output, Every input is connected with the output of another neuron body and every output is connected with the input of another neuron body, as is shown in Figure 3.8 [36].

Neurons in the brain are organized in a shape that allows performing special rules for specialized tasks. These rules have been studied and designed in applicable module with the same basic principles in computation field. ANN algorithm simulates these rules in transforming input data to the output data according to the match targets [28]. Generally artificial neural network consists of three elements called layers:

1. Input Layer: data fed into the ANN algorithm.
2. Output Layer: outputs prediction results.
3. Hidden Layer: represents the processing between input layer and output layer.

The main idea of ANN can be summarized by modifying the current input nodes (neurons) weights continuously if the results of prediction are incorrect and the design of arranging the connections among nodes [30]. MLP is a structured model of ANN. Before go deep in MLP some principles belong to it must be mentioned.

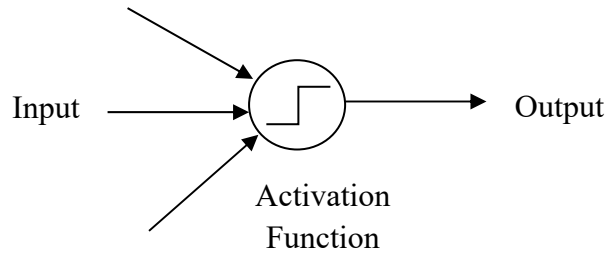


Figure 3.8: Perceptron a simple network structure of ANN.

Perceptron is the simplest network structure of ANN it is known as the single layer perceptron. It consists of only two layers, multi input and a single output and it can classify data with two classes, as shown in Figure 3.10. MLP contains hidden layer(s) which connects between input and output layers with a number of nodes in each layer usually in output layer more than two output nodes because it accepts multi classes and it deals with only numerical attributes data. Three layer nodes are full connection among them give ability to network classify multi classes.

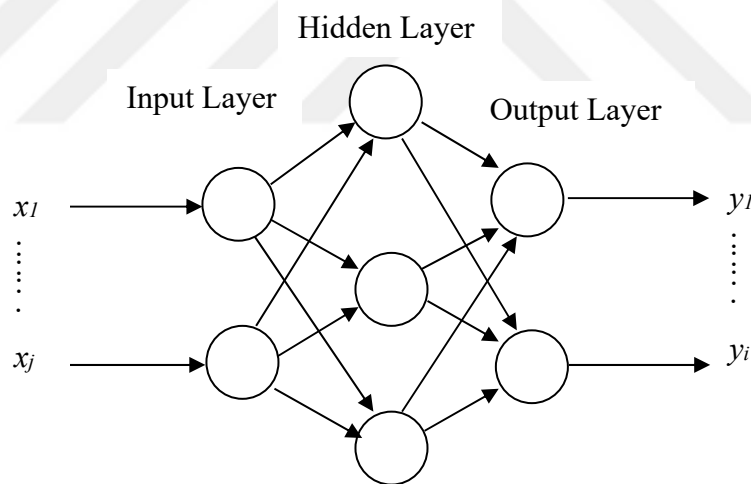


Figure 3.9: Structure of MLP.

3.15.1 Advantages and Disadvantages

There some advantages and disadvantages in applying ANN with IDS [39], as shown in Table 3.1.

Table 3.1: Advantages and disadvantages in applying ANN with IDS.

Advantages	Disadvantages
<ul style="list-style-type: none">• The Multilayer as hidden layers increases the efficiency of classification.• It can detect nonlinear relationships between dependent and independent variables.• It has toleration to noisy data.	<ul style="list-style-type: none">• It requires a larger number of instances of the training set.• It requires longer computation time.• It is lesser flexibility• It possesses greater computational complexity.

3.15.2 Decision Tree Algorithm

Decision tree algorithm is a classification algorithm; it can predict the value of output variables for the set of input variables [29]. It deals with classed training dataset because this algorithm is a supervised classification algorithm. Generally, decision tree algorithm builds a hierarchical structure model as top down tree form. Each node in the tree is called a split criterion, it is a condition such as if statement for one or more features of training dataset. Tree starts from top (root) and branches to nodes until it reaches to leaves [40]. Each condition node processes as a sub decision tree for sub training data set. The benefit of splitting decision tree process to nodes is reducing the size of the class variable in each branch by dividing the training data set. It looks like clustering operation, but it depends on class label in the dataset. Condition state in the nodes or split criteria is the main process operation to classify feature variables as a part of prediction process. Decision tree algorithm arranges nodes in levels according to the effective of feature variables, the most effective in the top and the less effective in the bottom that are branches to the leaves which is the predicted decision [30]. This common behavior is shared by all kinds of decision tree algorithms.

C4.5 algorithm is a decision tree algorithm; it is an enhanced of the older version ID3 algorithm. It makes its decisions by building a top down decision trees from training set attributes variables by using information entropy.

3.15.3 Advantages and Disadvantages

There some advantages and disadvantages in applying C4.5 with IDS [39], as shows in Table 3.2.

Table 3.2: Advantages and disadvantages in applying C4.5 with IDS.

Advantages	Disadvantages
<ul style="list-style-type: none"> • It is able to handle with high dimensional data. • It is easy to understand. • Has an ability of processing both numerical and categorical data. 	<ul style="list-style-type: none"> • The problem is in finding the smallest path of decision. • Limited to one output attribute. • Decision tree algorithms are unstable and trees created from numeric datasets can be complex.

3.15.4 Naive Bayes

Naive Bayes predicts the result of classification by using conditional probabilities of Bayes theorem. There are several applications use Bayes theorem in probability and statistics. The naive Bayes classifier model is a simplified Bayes classifier model [29].

The Bayes theorem was presented by Thomas Bayes in 1783 can be summarized as equation 3.1:

$$y = \underset{c_i}{\text{arg max}} \{P(c_i | x)\} \quad (3.1)$$

The Bayes theorem has ability to invert the $P(c_i|x)$ posterior probability in a form of likelihood with class and predictor prior probability, as follows:

$$P(c_i | x) = \frac{P(x | c_i) \cdot P(c_i)}{P(x)} \quad (3.2)$$

With training dataset D which contains n instances x_i in a d dimensional attributes, and y_i is the real class of each instance, with $y_i \in (c_1, c_2, \dots, c_k)$. $P(x, c_i)$ is known as the likelihood. It is the probability of a single instance indicates how much the instance related with specific class. c_i is the number of classes in the dataset $c_i, i = (c_1, c_2, \dots, c_i)$. $P(c_i)$ is the probability of one class occurrences form whole classes in the dataset. $P(x)$ is the probability of instance x occurring in the all dataset. $P(c_i|x)$ is known as the posterior probability it is the predicted class of the dataset's instance. This rule is the main principle of the Naive Bayes classifier.

3.15.5 Advantages and Disadvantages

There some advantages and disadvantages in applying NB with IDS [39], as shown in Table 3.3.

Table 3.3: Advantages and disadvantages in applying NB with IDS.

Advantages	Disadvantages
<ul style="list-style-type: none">• The prior knowledge about the system is simply that some variations might influence others.• High accuracy and faster with large training set.	<ul style="list-style-type: none">• The prediction is built on independent condition class.• Lack of available probability data.

3.16 Evaluation

Evaluation is the final stage of building the IDS model. It evaluates the performance efficiency of the model. The gained results of evaluation can be analyzed and compared with the results of other models. Training and testing, evaluation approach or cross validation is the common approaches have been used.

CHAPTER FOUR

DATASET FOR IMPLEMENTATION AND EVALUATION THE MODEL

4.1 Introduction

In this chapter we will provide the general description of testing and evaluation dataset. The dataset is one of the important substantive parts in IDS. From various types of data set, our choice was the most comprehensive and modern UNSW-NB15 dataset for our IDS system. The UNSW-NB15 data set was created and established by the synthetic environment at the University of New South Wales in Australia (UNSW) cyber security lab [4, 6].

4.2 UNSW-NB15 Description

UNSW-NB15 dataset represents a modern fashion of the real normal network traffic and the synthetically abnormal network traffic for simulating network community behavior [4, 6]. It contains nine main types of attacks each type is a family of various attacks, it has a labeled attribute so it is known as the labeled data set. There are 49 features in this dataset, these features define the behavior of attack and normal traffic packets. Features can be exploited by using machine learning in designing intrusion detection system. Data appears in dataset in a form of alphabets and numeric data [41].

UNSW-NB15 dataset is available for researchers from the web site of the university [42]. It is divided into Four downloadable files of data records each file has CSV format; these CSV files have the names UNSWNB15_1.csv, UNSW-NB15_2. Csv, UNSW-NB15_3.csv and UNSW-NB15_4.csv. In the first three CSV files contains 700000 records for each one and the fourth one contains 440044 records, totally number of records in four files above is 2,540,044 [6]. UNSW-NB15

dataset contains nine major types each type is a family of attacks, which classified in the dataset as its name in the labeled traffic type field such as a one feature [4, 41].

4.3 The Nine Categories of Attacks in UNSW-NB15 Dataset

According to what we mentioned previously there are nine main types of attacks each type branches to a group of attacks, overall they had been labeled in the dataset. Describing the attack is distinguishing between features. We will give a general description of each group depict the importance and novelty of the attack types adopted in the data set. The nine categories of attacks in UNSW-NB15 Dataset as shown below:

1. Fuzzers: It is recognized as a program, in this attack type, robotically semi-random data generated and feed into a suspend program or network and discover bugs. This kind is extraordinary than the traditional pure term of brute force attack in cryptanalysis field. Fuzzers attack is working to find exploitable holes in programming or in networking protocols after observing sets of known to be exploitable values as a fuzz vectors. The data type of exploitable sets contains integer, characters, metadata, binary or combination of them. fuzzing is a strong code tool for injection vulnerabilities in software or protocols such as HTTP protocol by dealing with extensible markup language XML. The malicious code can be written by RUBY or PYTHON programming language. This type of attack enable the attacker access to the information system after gaining Authority or by passing through firewalls such as in public-facing interface [4, 41].

2. Reconnaissance: The objective of attacker in this type is network protocols or web application to stale system information stimulates the attack. Clearly intrusion detection/prevention system can detect and prevent this type of attack by analyzing traffic packets. The reconnaissance is focused on a particular type of internet protocol or web application. This type divided into passive and active attack and passive, divided into direct and indirect each type depends on the behavior of attack. The passive reconnaissance attack is collecting information about network and web application by directly or indirectly methods. The indirect passive reconnaissance method is looking for reachable general information and Internet domain registration services. The direct passive reconnaissance attack can use specific tools that enable collecting available information from remote networks. The

active reconnaissance attack is described as an action of collecting network information from the target using directly methods [6, 43].

3. Shellcode: Code or fragments of code was written through low level programming machine language for various environments of operating systems to compromise either the computer network or local computer. Shellcode used as the payload within the exploitation of a software program bug which permits an unauthorized user to communicate with the computer by using command line of operating system. The results of exploiting vulnerability are a malicious code is running in the system[6, 44, 45].

4. Analysis: This attack includes a bunch of intrusion types under this heading such as port scans for web applications, spam for emails and HTML files for web scripts.

5. Backdoors: is one of the threatening techniques for accessing the computer system or its data without surveillance, usually the attacker is impersonated as authorized member to remotely access the system. The intruder Starts the attack by Locate the system entry and withstand in a hidden way [6]. Backdoor attack is a program written by intruder or written by the program developers. A backdoor is also known as a trapdoor [46]. This type of attack increases as community of network grows and networks become more multiple. Hackers use backdoors to inject malicious codes to gain access by embeds malicious codes with other commonplace programs.

6. Denial of Service: In this attack an authorized user is prevented of using accessing the computer system or its data partly or entirely. In denial of service (DoS) attacker group of attackers tries to fold or disconnect the service to the user by overload or disconnect network or shut down a computer. Generally DoS attack target internet and web application servers and obscured or slow down their services. There are many types of DoS like distributed Denial of Service (DDoS) attack but the famous one is that sending a wave of requests or enormous blocks of data to a computer network more than their handling capacity. There are several vestibule can be exploited such as potential physical, software, and network vulnerabilities to access and block services [6, 15].

7. Exploits: The attacker takes advantage of vulnerabilities in the system with previous experience such as weak programming or known loopholes in internet protocols.

8. Generic: The attacker targets all block ciphers with knowing the key length hash function with regardless about the type of hash function until finds collision in a block cipher.

9. Worms: It is a malicious code got ability to generate copies of it by itself and send copies from the host computer to other computers through networks [15]. Once it reaches to the other computer it can propagate in the same way because of the characteristics of the embedded programmed propagation. The attacker can gain some purposes from the victim's computer by separating worms. Worms target to operate undesirable purposes into a victim's computer, such as find another host to allocate a new copy or originate a remotely connected with the host. Worms can attack whole common operating systems and take advantage of email, browsers and sharing files.

As mentioned previously UNSW-NB15 dataset it contains nine main types of attacks each type is a family of various attacks. Table 4.1 shows each attack type and the number of times repeated with normal packets.

Table 4.1: Normal and attack categories and their frequencies in dataset.

Attack category	Number of Events
Normal	2218761
Fuzzers	24246
Reconnaissance	13987
Shellcode	1511
Analysis	2677
Backdoors	2329
DoS	16353
Worms	174
Exploits	44525
Generic	215481
Total	2540044

From the table (4.1) we can observe the total number of whole records is 2540044 and the difference between 2218761 records of normal and 321283 of abnormal records the conclusion from this ratio is that the percentage of the appearance of an attack in UNSW-NB15 is 12.64% and the ratio of normal records is 87.36%.

4.4 UNSW-NB15 Features Description

There are 49 features in UNSW-NB15 dataset, these features define the behavior of attack records and normal records. As shown in Table 4.2 the description of each feature and the type of data in data set files. These features divided into six groups Appear sequentially from 1 to 5 Flow features is for identifying the attributes, from 6 to 18 Basic features represents protocols connections, from 19 to 26 Content features encapsulates the attributes of network, from 27 to 36 Time features, from 37 to 47 Additional generated features divides features in data set into two groups General purpose and Connection features according to the type of record and the last group from 48 to 49 Labeled Features[6].

Table 4.2: Features description of UNSW-NB15 [42].

No.	Name	Type	Description
1	srcip	Nominal	Source IP address
2	sport	integer	Source port number
3	dstip	nominal	Destination IP address
4	dsport	integer	Destination port number
5	proto	nominal	Transaction protocol
6	state	nominal	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
7	dur	Float	Record total duration
8	sbytes	Integer	Source to destination transaction bytes
9	dbytes	Integer	Destination to source transaction bytes
10	sttl	Integer	Source to destination time to live value
11	dttl	Integer	Destination to source time to live value
12	sloss	Integer	Source packets retransmitted or dropped
13	dloss	Integer	Destination packets retransmitted or dropped
14	service	nominal	http, ftp, smtp, ssh, dns, ftp-data,irc and (-) if not much used service
15	Sload	Float	Source bits per second
16	Dload	Float	Destination bits per second
17	Spkts	integer	Source to destination packet count
18	Dpkts	integer	Destination to source packet count
19	swin	integer	Source TCP window advertisement value
20	dwin	integer	Destination TCP window advertisement value
21	stcpb	integer	Source TCP base sequence number

Table 4.2 (Continued): Features description of UNSW-NB15 [42].

No.	Name	Type	Description
22	dtcpb	integer	Destination TCP base sequence number
23	smeansz	integer	Mean of the ?ow packet size transmitted by the src
24	dmeansz	integer	Mean of the ?ow packet size transmitted by the dst
25	trans_depth	integer	Represents the pipelined depth into the connection of http request/response transaction
26	res_bdy_len	integer	Actual uncompressed content size of the data transferred from the server's http service.
27	Sjit	Float	Source jitter (mSec)
28	Djit	Float	Destination jitter (mSec)
29	Stime	Timestamp	record start time
30	Ltime	Timestamp	record last time
31	Sintpkt	Float	Source interpacket arrival time (mSec)
32	Dintpkt	Float	Destination interpacket arrival time (mSec)
33	tcprtt	Float	TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'.
34	synack	Float	TCP connection setup time, the time between the SYN and the SYN_ACK packets.
35	ackdat	Float	TCP connection setup time, the time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	Binary	If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0
37	ct_state_ttl	Integer	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
38	ct_flw_http_mthd	Integer	No. of flows that has methods such as Get and Post in http service.
39	is_ftp_login	Binary	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	integer	No of flows that has a command in ftp session.
41	ct_srv_src	integer	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).
42	ct_srv_dst	integer	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
43	ct_dst_ltm	integer	No. of connections of the same destination address (3) in 100 connections according to the last time (26).
44	ct_src_ltm	integer	No. of connections of the same source address (1) in 100 connections according to the last time (26).
45	ct_src_dport_ltm	integer	No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26).
46	ct_dst_sport_ltm	integer	No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).
47	ct_dst_src_ltm	integer	No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).
48	attack_cat	nominal	The name of each attack category. In this data set, nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms
49	Label	binary	0 for normal and 1 for attack records

The overall structure of the Table 4.2 shows the UNSW-NB15 dataset consists of different data types such as integer, binary, float, timestamp and nominal. There are some attributes whose values are related to other attributes that appear at the beginning of the table.

CHAPTER FIVE

MODEL METHODOLOGY

5.1 Introduction

This chapter reviews the model methodology implemented in this research and presents a solution to the intrusion detection problem by using data mining and information retrieval techniques. We will outline the theoretical description of these techniques also the dataset that simulated the real mutual network packets. Recognizing normal and malicious packet problem of simulated dataset is the main objective of this research. We proposed a model as a recognizing solution. The main function of our model is converting the data type of simulated dataset to acceptable forms of classification algorithms which results efficient performance of high correctly detection rate. We will experiment the efficiency of the same converted new dataset with different ML classifiers in two approaches, the first one with multi class dataset and second approach with binary class dataset. The converted new datasets will be evaluated by using cross-validation approach in training and testing stage.

5.2 Model Architecture

The architecture of our model is depicted in Figure 5.1. The diagram below represents the overview of how the simulated dataset will be converted and classified. The simulated dataset is a structured data, its records in the form of real network packets whose data type and attributes was explained in Chapter 4. After a proper amount of dataset records will be selected randomly, the model will preprocess selected dataset in two steps. First step the selected dataset will be processed by using TF.IDF techniques to weight each item in the dataset also replace the item value by its weight. Second step in the dataset attributes will be selected.

The new dataset has two categorical attributes one for binary class and another for multi class. Different techniques of ML algorithms will be applied for anomaly detection. These techniques are supervised classifiers. This thesis will use Multilayer Perceptron, Naïve Bayes and C4.5 decision tree algorithms to evaluate the approach.

Below is a sequence of steps used in this research work:

1. Selecting randomly subset of the UNSW-NB15 dataset.
2. Creating two copies of selected subset, with deleting different only one class attribute from two class attributes to construct multi and binary class data sets.
3. Converting file type from.csv extension to.txt extension for compatibility with our model which developed with C# programming language by using Microsoft Excel.
4. Dataset preprocessing by transforming and normalizing both multi and binary class datasets files with TF.IDF algorithm.
5. Datasets preprocessing by reducing attributes from both multi and binary class dataset files.
6. Training and testing the MLP, NB and C4.5 ML algorithms with both multi and binary class datasets by using cross validation approach.
7. Evaluating results of both datasets.

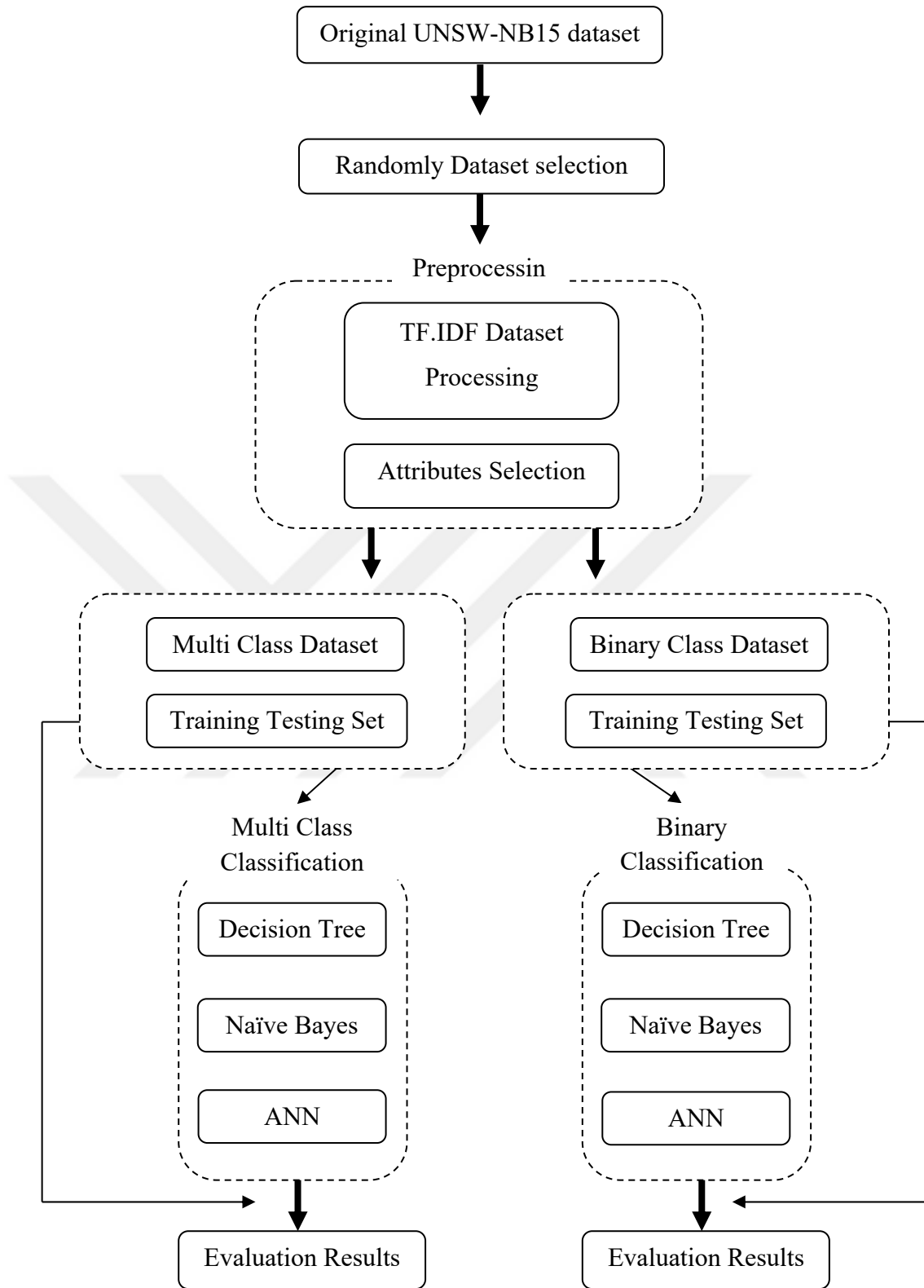


Figure 5.1: Model architecture.

5.3 Implementing Tools

5.3.1 Visual Studio C#

It is an object-oriented programming language from Microsoft. Visual C# is one of many programming languages presents a variety of secure and robust applications that run on the .NET Framework. First appearing of this programming language was in 2000. In our work we developed our preprocessing model with visual studio C# 2013 Ultimate [47].

5.3.2 Weka

Our model classification stage will be implemented by using Weka 3.8, it is an open source software tool contains several of data mining and machine learning algorithms such as data preprocessing, classification, clustering, regression, association rule mining along with data visualization functions. It is freely downloadable from official web site of Waikato University, New Zealand [48]. First version of this software was developed in 1997, then a number of updates were made by researchers [35, 49] [50]. We will use Weka software for selecting the proper attributes, implementing ANN, C4.5 and NB classifiers and evaluating the model results.

5.3.3 Microsoft Excel

It is a part of Microsoft suite. MS Excel as is known, it contains several data analysis tools. This application is dedicated to PC computers for home use and small business. In our work we will this application to prepare dataset files for C# and Weka software also to determine the class type of dataset by erasing one from two attribute classes.

5.4 Dataset Selection

There are 49 features in UNSW-NB15 dataset; these features define the behavior of attack and normal records. UNSW-NB15 dataset consists of different data types such as integer, binary, float, timestamp and nominal, in addition to

missing values. In this work we will experiment 250000 records of overall UNSW-NB15 dataset. The new selected subset will be used for training and testing stage. Instances are chosen randomly from the whole UNSW-NB15 dataset. The portion of the dataset that will be experimented represents approximately 10% of 2540044 the total number of instances is the UNSW-NB15 dataset. The proportion of normal records against the attacker records by choosing dataset is 194438 normal records equals to 77.77% and 55562 records equals to 22.22%, as shown in Figure 5.2.

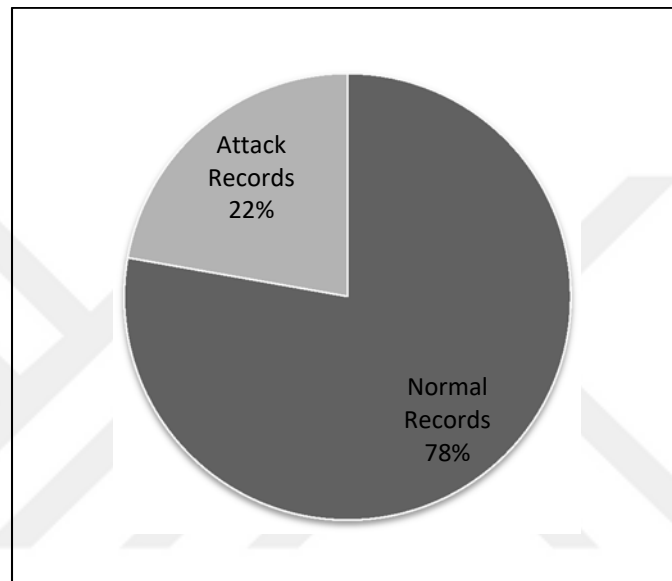


Figure 5.2: Rate of normal records to attack records.

5.5 Customize Dataset

There are two attribute classes in UNSW-NB15 dataset since the main categories of the dataset records are normal and attack [4]. Attribute number 48 is an Attack_Cat represents attack categories which consist 10 classes, 9 classes of attack which are Fuzzers, Reconnaissance, Shellcode, Analysis, Backdoors, DoS, Worms, Exploits and Generic and 1 class for normal behavior. Attribute number 49 is a Label represents either attack or normal. Binary class dataset contains 1 for attack and 0 for normal records. To prepare dataset for experimental classification, the class column must be determined by canceling one of the two labeled attributes. For multi class dataset the attribute number 49 should be deleted and for binary class dataset the attribute number 48 must be deleted.

5.6 Dataset Preprocessing

Data Mining is the process to produce useful information for future prediction and classification. The dataset contains impossible values and missing values to apply it with classifiers. Data preprocessing improves the quality of data to make it more meaningful for classifiers by using data transformation and attributes reduction [49, 50]. The implementation of data preprocessing improves the performance of the classifiers and thus improves the prediction results and execution time [9].

5.6.1 TF.IDF

The TF-IDF is abbreviated of term frequency – inverse document frequency is a statistical measure used in information retrieval (IR) and text mining realm to weight a word to a document in a corpus by finding a statistical measure of the weight to evaluate the importance of a word. The importance of the word increases with the increasing frequency of the word in the text, and the corresponding increase in frequency of the same word in the documents in corpus. Different weighting schemes of the TF.IDF have been used in modern search engines to score and rank documents terms [51]. There are two terms TF and IDF in the mother term of TF-IDF as explained below:

Term Frequency (TF) measures the occurring of how many a term is repeated in a document with consideration that the document is different in text length (number of words) [51]. In text document it is possible that a term is repeated much more times in long text documents than shorter text document, as defined in equation.

The weight of that term denoted as term frequency $tf_{i,j}$ is the number of times term i appears in a document j , and the denominator is the \max_k number of a term occurrences from every terms in the document j . Dividing operation produce normalized weight value of each term in document [52].

$$tf_{ij} = \frac{f_{i,j}}{\max_k f_{k,j}} \quad (5.1)$$

Inverse Document Frequency (IDF) measures the general importance of a term in an individual or a number of documents. idf is obtained by the logarithm of dividing the total number of documents in a collection N by n (document frequency) number of documents which contains the term [52].

$$idf_i = \log_2 \frac{N}{n_i} \quad (5.2)$$

The logarithm is used to reduce the differences between the frequencies of the uneven appearance of the terms in the documents, because some of the terms appear significantly in a single document and a set of documents more than other terms [51].

5.6.1.1 TF-IDF weighting

By merging the principles of term frequency and inverse document frequency, to produce a composite weight for each term t in each document d [52].

$$tf_{ij} \cdot idf_i = tf_{ij} \times idf_i \quad (5.3)$$

There are some properties in TF.IDF equation:

1. The weight is higher when a term t appears many times within a small collection of documents.
2. The weight is lower when the term appears fewer times in a document, or appears in many documents.
3. The weight is the lowest when the term appears in virtually all documents.

5.6.1.2 TF.IDF dataset processing

The general use of TF.IDF approach is with text mining to give a weight for every word in a document to find the special words that characterize documents like in search engines or document recognition [13]. In this research we will employ this property with structured data form instead of unstructured text documents. The selected UNSW-NB15 dataset is a structured dataset. It has continuous, symbolic, categorical and missing value items with attributes. TF.IDF Process will be used for converting the different data types of homogeneous numeric value and eliminating missing values by converting it to normalized numeric form according to the Equations 5.1, 5.2 and 5.3. Normalizing values is a converting method to restrict

dataset values by a higher and lower limitation to prevent the impact of the large scale features over the lower scale features [29, 30]. TF.IDF presents normalization by the TF Equation 5.1, for example a segment of 100 record contains 470 items if we suppose one item occurrences one time and another item occurrences 100 times, then the TF of first item equals to $1/100=0.01$ and the TF for the second item equals to $100/100=1$, so that:

1. The highest TF value $tf_{ij}=1$
2. The lowest TF value $tf_{ij} > 0$

The value of each item in the dataset will be replaced by the item's weight in overall dataset. We will experiment this model with different datasets of various segment sizes such as 50, 500, 1000, 4000 and 5000 records for multi and binary class datasets. This process can be illustrated in Figure 5.3. The sequences of TF.IDF processing steps as below:

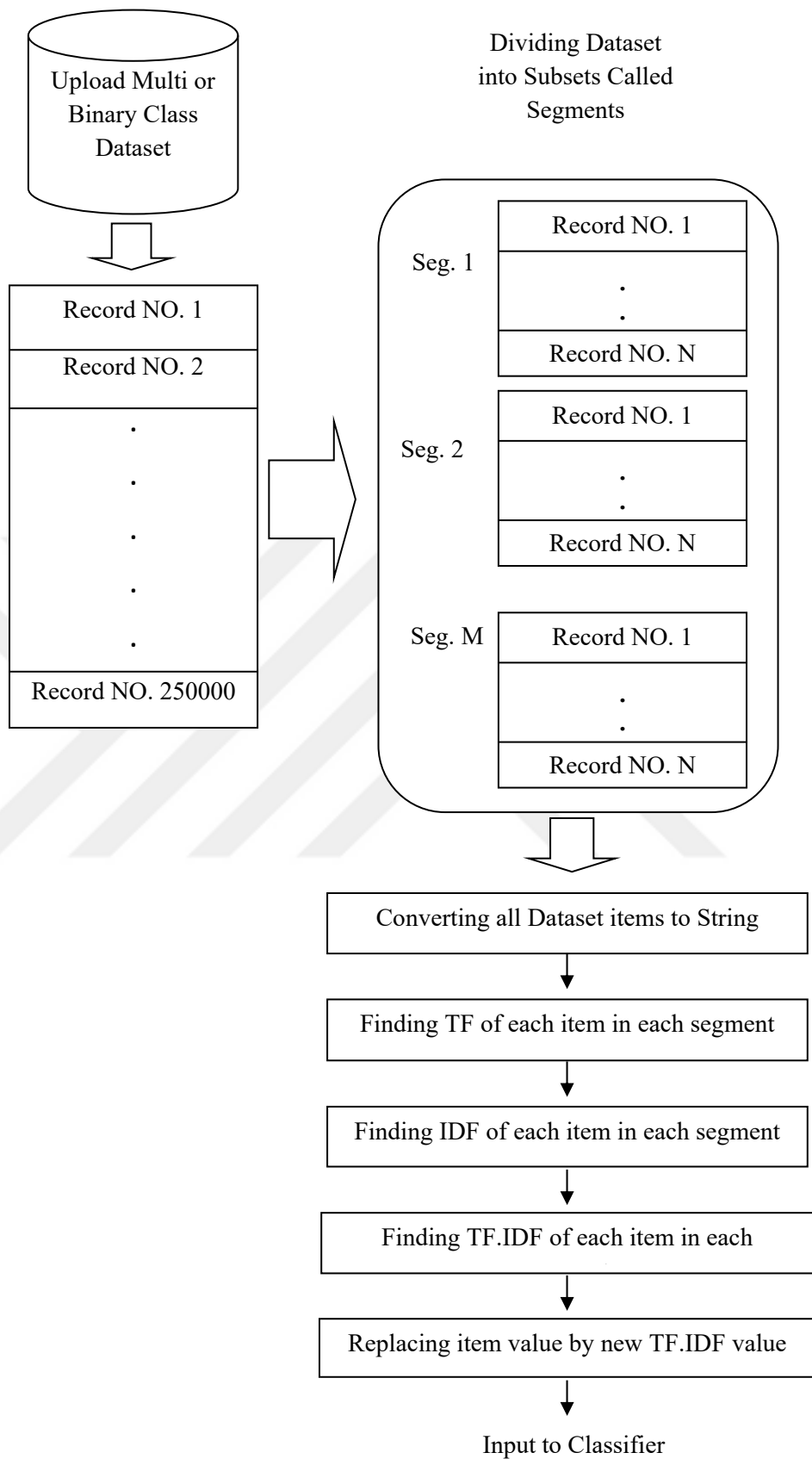


Figure 5.3: TF.IDF processing steps.

1. Dividing the overall dataset D to equal segments size M , each segment will be considered as a document and overall new dataset D' will be considered as a corpus or a collection of documents N , as shown below.

$$N = \frac{D}{M}$$

$$D' = M \times N$$

2. Converting different data types, in addition to missing values in every record in every segments item in string type as a first step in computing the weight of items. In this step the structured form of the segments will be converted into unstructured form as a conventional document.

3. Find the item with the largest number of occurrences in a single segment (document) to perform value normalization.

4. Finding TF of each item in a segment (document) by using equation 5.1.

5. Finding IDF of each item in a segment with all segments in the overall dataset by using equation 5.2.

6. Computing TF.IDF of each item by using equation 5.3. The results of this step represent the numeric weight of each item i , the final equation will be:

$$tf_{iM} \cdot idf_i = \frac{f_{iM}}{\max_M f_{kM}} \times \log \frac{N}{n_i} \quad (5.4)$$

7. Replacing the generated value from the TF.IDF computation of each item with the original value of the same item.

5.6.2 Attributes Reduction

After TF.IDF stage as mentioned previously the dataset is converted to numeric form, so that there are no different data types such as timestamp, nominal or missing values. In this stage the dataset size need reducing by eliminating the attributes that irrelevant to reduce dataset space and transforming dataset to another new one. The purpose of feature selection is increasing efficiency and reducing the computation time of classifier [30].

This data set has significant features according to what the authors of dataset Nour Moustafa and Jill Slay [6] had presented in the field of selecting attributes of the UNSW-NB15 dataset. They used central points of attribute values and A priori algorithm to mitigate the FAR impact. This algorithm is designed to process the dataset in a short time, due to its dependency on the central points of feature values with partitioning data records into equal parts. They selected high ranked features and removed irrelevant features for each class which cause high FAR. The number of features were recommended by authors are shown below in Table 5.1.

Table 5.1: Class name and relevant features[6].

NO.	Class Name	Feature numbers
1	Normal	11,34,19,20,21,37,6,10,11,36,47
2	DoS	6,11,15 16,36,37,39,40,42,44,45
3	Fuzzers	6,11,14,15,16,36,37,39,40,41,42
4	Backdoors	6,10,11,14,15,16,37,41,42,44,45
5	Exploits	10,41,42,6,37,46,11,19,36,5,45
6	Analysis	6,10,11,12,13,14,15,16,34,35,37
7	Generic	6,9,10,11,12,13,15,16,17,18,20
8	Reconnaissance	10,14,37,41,42,43,44,9,16,17,28
9	Shellcode	6,9,10,12,13,14,15,16,17,18,23
10	Worms	41,37,9,11,10,46,23,17,14,5,13

In this work we will rely on what the Nour Moustafa and Jill Slay [6] presented in selecting the high ranked features of UNSW-NB15 dataset. Since our work deals with over all dataset, the features will be union to create a new set of appropriate attributes from the original dataset, with adding the multi or binary class attribute. The new set will be as shown below in Table 5.2.

Table 5.2: The new set of UNSW-NB15 dataset.

NO.	Feature Number	Feature Name	NO.	Feature Number	Feature Name	NO.	Feature Number	Feature Name
1	5	proto	11	17	Spkts	21	37	ct_state_ttl
2	6	state	12	18	Dpkts	22	39	is_ftp_login
3	9	dbytes	13	19	swin	23	40	ct_ftp_cmd
4	10	sttl	14	20	dwin	24	41	ct_srv_src
5	11	dttl	15	21	stcpb	25	42	ct_srv_dst
6	12	sloss	16	23	smeansz	26	43	ct_dst_ltm
7	13	dloss	17	28	Djit	27	44	ct_src_ltm
8	14	service	18	34	synack	28	45	ct_src_dport_ltm
9	15	Sload	19	35	ackdat	29	46	ct_dst_sport_ltm
10	16	Dload	20	36	is_sm_ips_ports	30	47	ct_dst_src_ltm
							48/49	attack_cat/label

Weka software is used for reducing dataset attributes by selecting the chosen subset in Table 5.2. We didn't apply reducing attributes before TF.IDF processing to give flexibility in attribute selection to our model. The main benefit of multi class attack classification, it gives us a thorough and deep understanding the relationship between the dataset features and attack categories [53]. Likewise for binary class it is important to maintain the results of the classification within the circle of attack.

5.7 Machine Learning Classifiers

The IDS model should be experimented by applying ML classifier. We chose different classification techniques for checking our model [54]. Our choice was Multilayer Perceptron, Naïve Bayes and C4.5 decision tree algorithms which they represent supervised classifiers and are widely used in network security. They are selected in despite of the contrasted results in the detection accuracy and processing time for the purpose of study and analyze the results of the model. The same training testing set will be picked for them to evaluate our model by using Weka software.

5.7.1 C4.5 Decision Tree

Another name of decision tree C4.5 algorithm is J48. It is an open source developed by java is offered as a classification approach in the Weka software. C4.5 algorithm is an enhanced version of ID3 decision tree algorithm by Quinlan in 1993 [55]. It is a supervised classification algorithm [35]. It builds a top down decision trees from selected dataset attributes variables by using information entropy. C4.5 decision tree algorithm is superior over ID3 algorithm in [56]:

1. Dealing with continuous attributes.
2. Dealing with missed and noisy values.
3. Algorithm mitigate over fitting of the data.

The algorithm is not restricted by the type of attribute values in constructing the decision tree branching. With binary data type there is a couple of branching from each node they are 0 or 1 in condition statement it can represented by yes or no. The algorithm takes the same behavior with Boolean data type. In the numerical attributes the outcome is two branches such as binary but instead of numbers the algorithm uses a range of numbers in each branch by creating a threshold for that attribute [57]. Attribute values are divided into two parts first one greater than threshold and the second one equal or less than threshold. This threshold gives the algorithm the ability of handling with continuous values. The number of outcomes may be increased more than two branches if the type of data value is categorical, the data will be grouped as smaller sets of data, each set represents an outcome in prediction process.

C4.5 uses information gain and entropy to choose and splitting the set of attributes from the multiple attributes dataset as a subset of one class or more than one [30]. Entropy measures the amount uncertainty in the dataset, if a partition has lower entropy it is relatively pure or most of instances has the same class type and it is considered desirable because no need to split pure node into another nodes [30]. If partition has higher entropy then labels are mixed and there is no majority to class the attributes [29].

Entropy is a theory of Shannon it can be represented as $E(S)$ for S is a set of instances computed as shown in equation 5.5. $p_1 \dots p_k$ is the class distribution of the training data set with the number of k classes [30].

$$E(S) = -\sum_{j=1}^k p_j \log_2(p_j) \quad (5.5)$$

The information gain is another measure in decision tree algorithm it can be found by calculates the reduction between the Entropy $E(S)$ before split and after split $Entropy-Split(S \Rightarrow S_1... S_r)$. Entropy-Split equation () as [30]:

$$Entropy-split(S \Rightarrow S_1... S_r) = \sum_{i=1}^r \frac{|S_i|}{|S|} E(S_i) \quad (5.6)$$

S is a set which is split into $S_1... S_r$. for each S_i , where the weight of S_i is $|S_i|$. Information Gain is a measure of effectiveness of all possible value in attribute A in training set as the equation 5.7 below [58] [30].

$$InformationGain(S, A) = Entropy(S) - \sum_{i=1}^r \frac{|S_i|}{|S|} E(S_i) \quad (5.7)$$

Large values of information gain (reduction) are desirable. The algorithm splits the effective attributes that have the highest information gain value from training set as a sub set [58]. Information gain is much related to entropy and both of them are used to compare between two splits with same larger weight [30]. For sample set S , on the assumption each attribute has i different discrete values then the $S_1... S_r$. for each S_i divided by attribute can generate the following equation:

$$SplitInfo(S, A) = -\sum_{i=1}^r \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} \quad (5.8)$$

The equations 5.5, 5.6 and 5.7 are the basic concepts inherited from ID3 decision tree algorithm. When C4.5 uses information gain is not anew implementation but it needs it to calculate the gain ratio as below [58] [30]:

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInfo(S, A)} \quad (5.9)$$

In our model we have considered the number of attributes is 30 to construct a decision tree.

5.7.1.1 Pruning

Pruning is a way of mitigating the over fitting of data in decision tree by cutting portion of the tree that is not influence to the classifying accuracy. Using pruning increases the classifier accuracy exactly with noisy data or in classifying instances which are not belonging to the class predicted by that leaf [55, 57, 58]. The pruning mechanism starts after the decision tree created. It checks the tree nodes and attempts to mitigate branches by replacing undesired nodes by leaf nodes [40].

5.7.1.2 C4.5 (C4.5) algorithm

Firstly the C4.5 algorithm as shown in a pseudo code as below in Figure 5.4 accepts all attribute's data type from training dataset. It checks the termination criteria if it is satisfied or no need to split a pure node because all instances have the same class value, then stopping is applied. In another choice when termination criteria is not satisfied then the algorithm computes information-theoretic criteria for all attributes such as in step 6. The algorithm chooses the best attribute as a root of decision tree according to information-theoretic criteria. After determine the root the algorithm splits the attributes as nodes from lowest level to the top of the tree with updating until termination criteria is satisfied.

Algorithm to generate C4.5 decision tree

Input: an attribute –valued dataset D

1. Tree = {}
2. if D is “pure” or stopping criteria met then
3. terminate
4. else if
5. for all attribute $a \in D$ do
6. Compute information-theoretic criteria if we split on a
7. end for
8. a_{best} = Best attribute according to above computed criteria
9. Tree = Create a decision node that tests a_{best} in the root
10. D_v = induced sub –datasets from D based on a_{best}
11. For all D_v do
12. $Tree_v = C4.5(D_v)$
13. Attach $Tree_v$, to the corresponding branch of tree
14. end for
15. return Tree

Figure 5.4: Pseudo code of C4.5 algorithm [40, 57].

C4.5 algorithm will be implemented as an open source name J48 algorithm by using Weka software, with default algorithm setting as shown in Table 5.3.

Table 5.3: J48 algorithm setting by using Weka software.

Options	Value
seed	1
Unpruned	False
confidenceFactor	0.25
numFolds	3
numDecimalPlaces	2
batchSize	100
reducedErrorPruning	False
useLaplace	False
doNotMakeSplitPointActualValue	False
debug	False
subtreeRaising	True
saveInstanceData	False

5.7.2 Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron (MLP) is a multi-layer feedforward neural network. It is one of successful classification algorithm in IDS realm. Because it is a feedforward the data is transmits in one way direction from input to output by hidden layer(s) [30].

Number of input nodes represents the dimensionality of dataset as equation 5.10, d is the number of nodes in input layer, the parameters w_i are the weights w_i , $i=1, 2, \dots, k$. k is the number of hidden layer nodes, w_{ij} the number of connections, y_i is the output from the input node and x_j is the attribute data value $x_j, j=1, 2, \dots, d$. This is for one instance of dataset is inputted to the MLP at a time. In our work there are nine classes for each class there is an output node. The number of nodes in input layer must cover the number of attribute in the dataset. This stage of processing it is the same in single layer perceptron. w_{i0} is the bias unit as the weight; it is used to make the model in general form. Always it is equal to one.

$$y_i = \sum_{j=1}^d w_{ij} x_j + w_{i0} \quad (5.10)$$

After that, data is transferred to the next layer. Activation function is applied in each node commonly sigmoid non-linear function has been used. The output value of this function is between 0 and 1. The value between 0 and 1 called threshold and if the node operation results in the threshold boundaries the node will be activated to transfer its value otherwise not. Activation function shown as equation below [33]:

$$z_h = \text{Sigmoid}(\mathcal{W}_h^T \mathcal{X}) = \frac{1}{1 + \exp\left[-\sum_{j=1}^d w_{hj} x_j + w_{h0}\right]} \quad h=1, \dots, H. \quad (5.11)$$

The $x_j, j = 0, \dots, d$ are the inputs and $z_h, h = 1, \dots, H$ is the hidden layer nodes where H is the dimensionality of this hidden layer. w_{hj} are weights in the hidden layer. w_{h0} is bias values in the hidden layer. Weights, w_{hj}, w_{h0} is the initial random values that are generated when the network starts processing.

5.7.1 Backpropagation and Error Redaction

The main propose of backpropagation stage is to estimate the weights in the primary network layers by error estimation from the errors in later layers. Error estimation is used to update the weight of the node after compute the error with consideration of the weight of that node [30]. The weight updating in MLP can be calculated as the equation 5.12. MLP attempts to reduce the error by exploiting backpropagation training algorithm. This operation called error minimization. Backpropagation starts its procedure with random values for weights and bias as initial values in the beginning, it continues in its procedure until lowest error rate is obtained [59]. Error reduction operation stops when the error deference equal to the result of equation 5.12 below. The operation of backpropagation by using error minimization is done by comparing the result as class y' from the output layer of a data instance with the actual class y of this data in the training dataset. The MLP achieves prediction with high accuracy when it is used with the Mean Square Error (MSE) as error redaction estimator as equation 5.12.

$$\text{Error} = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2 \quad (5.12)$$

In the MSE equation n represents the number of instances in training dataset, y_i is the actual class of each instance and y_i' is the result of MLP classification for the same instance.

MLP algorithm will be implemented by using Weka software, with default algorithm setting as shown in Table 5.4.

Table 5.4: MLP algorithm setting by using Weka software.

Options	Value
seed	0
momentum	0.2
nominalToBinaryFilter	True
hiddenLayers	a
validationThreshold	20
normalizeAttributes	True
numDecimalPlaces	2
batchSize	100
decay	False
trainingTime	500
debug	False
autoBuild	True
normalizeNumericClass	True
learningRate	0.3
doNotCheckCapabilities	False
reset	True

5.7.3 Naive Bayes

The Naive Bayes (NB) is a fast and a high accuracy supervised machine learning classifier from probabilistic classifier family [60]. With training dataset D which contains n instances x_i in a d dimensional attributes, and y_i is the real class of each instance, with $y_i \in (c_1, c_2, \dots, c_k)$. $P(x, c_i)$ is known as the likelihood. It is the probability of a single instance indicates how much the instance related with specific class. c_i is the number of classes in the dataset $c_i, i = (c_1, c_2, \dots, c_i)$. $P(c_i|x)$ is known as the posterior probability it is the predicted class of the dataset's instance.

Naive Bayes estimate the probability of classifying instances in dataset by estimating each feature in that instance independently from other features, for this reason there is no need to compute class and predictor prior probability. Naive Bayes computes likelihood probability to classify each attribute. Likelihood equation 5.13 is as observed below, computes the probability of the class for the instance. That means the probability of an attribute will not affect at the probability of another attributes [61].

$$P(\mathbf{x} | c_i) = P(x_1 | c_i) \times P(x_2 | c_i) \times \dots \times P(x_n | c_i) \quad (5.13)$$

Likelihood equation 5.14 can be decomposed into a product of probabilities for each attribute independently:

$$P(\mathbf{x} | c_i) = P(x_1, x_2, \dots, x_d | c_i) = \prod_{j=1}^d P(x_j | c_i) \quad (5.14)$$

This translates into the probability of generating instance d given class cj equals the probability of class cj generating the observed value for the first feature, multiplied by the probability of class cj generating the observed value for the second feature time the probability of class cj generating the observed value for the third feature and so on until the probability of the last feature has been input [60].

When the Naive Bayes algorithm working with numeric attributes assumes normal distribution to find the probability of the classification by using the mean μ_{ij} and variance σ^2_{ij} for each attribute x_i and for each class c_i as shown in equation 5.15 [29].

$$P(x_j | c_i) \propto P(x_j | \mu_{ij}, \sigma^2_{ij}) = \frac{1}{\sqrt{2\pi\sigma^2_{ij}}} \exp\left\{-\frac{(x_j - \mu_{ij})^2}{2\sigma^2_{ij}}\right\} \quad (5.15)$$

For our work the UNSW-NB15 dataset was converted to numeric attributes for testing classification with the Naive Bayes algorithm. In the testing phase, after computing each attribute's value of an instance the class which has the highest value will be the predicted class.

NB algorithm will be implemented by using Weka software, with default algorithm setting as shown in Table 5.5.

Table 5.5: NB algorithm setting by using Weka software.

Options	Value
useKernelEstimator	False
numDecimalPlaces	2
batchSize	100
debug	False
displayModelInOldFormat	False
doNotCheckCapabilities	False
useSupervisedDiscretization	False

5.8 Training and Testing Stages

In this thesis all machine learning classifiers will be trained and tested by using cross validation with randomly selected multi and binary datasets to evaluate the model performance with estimate the complexity of the UNSW-NB15 dataset.

5.9 Cross Validation

The cross validation has been made for machine learning performance testing and evaluating the effectiveness [62]. It divides the labeled dataset D randomly into K equal size parts which is called folds. It tests the whole data set in K times by testing one part in each time. This approach tests all parts of data set with all dataset then the average accuracy of each part will be reported. Finally, the average of k testing folds is calculated to gain the overall performance accuracy.

Usually the K folds are equal to 5 or 10 [29]. Generally the common approach to evaluate machine learning is done by 10 folds which mean the dataset is divided into 10 parts, each part is selected randomly. This method is based on holding one part for testing and nine parts for training. Finally overall dataset will be tested and trained for 10 times, each time [63]. Extensive researches with different datasets and various machine learning algorithms were used 10 folds to obtain the estimate of error [35].

This operation is applied through using a cross validation feature in Weka software in determining the number of folds. In this thesis, we will use 10 fold cross

validation. Both of multi and binary class datasets are picked for Multilayer Perceptron, Naïve Bayes, and C4.5 decision tree algorithms. The results of the classifier are the part that really lets us know whether our IDS model is it working perfectly or not.



CHAPTER SIX

RESULTS AND DISCUSSION

6.1 Introduction

This chapter depicts the results of our IDS model after the completion of training and testing stages. We will review the results of two experiments. The first experiment was implemented with multi class dataset and the second one with the binary class dataset. Both of these datasets with their prepossessed data are tested with Multilayer Perceptron, C4.5 and NB classifiers. The model efficiency of the IDS model was evaluated with the common indicators of machine learning performance. All experiments and results were done by using the Weka 3.8 software. The system was implemented two times in each time five different classifiers were tested.

6.2 Environment

The experiments were completed on a portable computer with i7-6700HQ 2.6 GHz CPU and 8 GB RAM. The Multilayer Perceptron, C4.5 has known J48 and NB algorithms are implemented in Weka 3.8 software.

6.3 IDS Model Performance Evaluation

The consideration of IDS is a sensitive part in network security world, the performance of intrusion detection model (IDS) should be evaluated as an evaluation system procedure to find the detection and failing detection rate in proposed model [7] [6, 49, 64]. Confusion matrix is two dimensions table displays the performance results of ML classification with one row and one column for each class. The matrix

shows the number of the actual class in rows and predicted class is columns as Table 6.1 shows the representation of confusion matrix. There are four measurement principles can be concluded from confusion matrix [29, 35, 62].

Table 6.1: Representation of confusion matrix.

		Predictive	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

1. TP: It means true positive. It is the number of instances from whole dataset that classified correctly as positive.
2. FP: It means false positive. It is the number of instances from whole dataset that classified as positive while it is negative.
3. TN: It means true negative. It is the number of instances from whole dataset that classified correctly as negative.
4. FN: It means false negative. It is the number of instances from whole dataset that classified as negative while it is positive.

For our work we developed confusion matrix in different dimensions according to the experiment. First experiment with multi class dataset, it contains ten classes the confusion matrix is mapped as 10×10 rows and columns. Likewise the confusion matrix is mapped as 2×2 with binary class dataset.

6.4 Evaluation Indicators

In this work we will mention seven indicators to evaluate our IDS model:

1. True positive rate (TPR): It is known as sensitive [29], it is the number of instances in dataset that classified correctly for all classes.

$$TPR = \frac{TP}{TP + FN} \times 100 \quad (6.1)$$

2. False positives rate (FPR): It represents the number of instances for all classes in the dataset that classified incorrectly [65].

$$FPR = \frac{FP}{FP + TN} \times 100 \quad (6.2)$$

3. Precision: It represents the measure of probability that positive instances are predicted correctly[7].

$$precision = \frac{TP}{TP + FP} \times 100 \quad (6.3)$$

4. Accuracy: In weka it is known as correctly classified instances. It is the common measure of ML classification performance. Accuracy is the percentage of instances which is detected correctly over all the instances in the dataset.

$$accuracy = \frac{TP + TN}{n} \times 100 \quad (6.4)$$

$$n = TP + TN + FP + FN$$

5. Error Rate: In weka it is known as incorrectly classified instances. It is the percentage of the incorrectly detected instances over all the instances of dataset [65].

$$errorrate = \frac{FP + FN}{n} \times 100 \quad (6.5)$$

$$n = TP + TN + FP + FN$$

6. F-measure: It measures the accuracy of the test. It scores the balance between precision and TPR. The F-measure is considered as the harmonic mean of TPR and precision [66], as equation 6.6:

$$f - measure = \frac{2 \times (precision \times TPR)}{(precision + TPR)} \quad (6.6)$$

The equations above are the global measures of ML classifier performance. The efficient and reliable classifier is when the accuracy rate is close to 100% and the error rate is close to 0%. The Weka software presents all indication rates that mentioned above.

6.5 Results

We will review the results of implementing Multilayer Perceptron, C4.5 and NB classifiers with multi and binary class preprocessed datasets. Five different datasets were tested with each classifier. Each dataset has different segment size of 50, 500, 1000, 4000 and 5000 records. The results testing stages appear in the confusion matrix and Detailed Accuracy by Class of Weka software. Confusion matrix was developed based on the ML algorithms with different segmented datasets. The average of TPR, FPR, Precision and F-Measure ratios with correctly and incorrectly classified instances will be shown below. In the most right of tables in this section we mentioned the building model time the number of classified classes for each case.

6.5.1 Results of the First Experiment

In this part we will present the results of implementing Multilayer Perceptron, C4.5 and NB classifiers with multi class preprocessed dataset. The results of the implementation of each algorithm appear as follows:

6.5.1.1 Decision tree C4.5 (J48)

The highest accuracy of correctly classified instances is when the segment size equals to 4000 records and it detected 9 from 10 classes as shown in Figure 6.1 the Confusion matrix and detailed accuracies by classes. A dataset with a segment size equals to 500 is the least training time they consumed 32 second. The least accuracy is 96.76% when the segment size is 50. Table 6.2 shows overall averages of performance evaluation of J48 with multi class datasets.

Table 6.2: Performance evaluation with C4.5 (J48) with multi class datasets.

Seg. Size	Correct Acc.	Incorrect Acc.	TPR	FPR	Pre.	F-M	Time Sec.	No. of classes
50	96.79%	3.20%	0.968	0.021	0.966	0.966	91	9
500	97.27%	2.73%	0.973	0.013	0.971	0.971	32	8
1000	97.36%	2.63%	0.974	0.013	0.972	0.972	36	8
4000	97.40%	2.60%	0.974	0.013	0.973	0.972	44	9
5000	97.36%	2.61%	0.974	0.013	0.972	0.972	92	9

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.997	0.016	0.995	0.997	0.996	0.982	0.997	0.998	Normal
	0.854	0.015	0.588	0.854	0.697	0.700	0.947	0.697	Exploits
	0.665	0.001	0.834	0.665	0.740	0.743	0.965	0.705	Reconnaissance
	0.686	0.003	0.753	0.686	0.718	0.715	0.960	0.692	Attack
	0.985	0.001	0.994	0.985	0.989	0.987	0.997	0.989	Generic
	0.141	0.003	0.337	0.141	0.199	0.213	0.931	0.294	DoS
	0.265	0.000	0.377	0.265	0.311	0.316	0.728	0.134	Shellcode
	0.018	0.000	0.200	0.018	0.034	0.060	0.937	0.046	Backdoor
	0.136	0.000	0.595	0.136	0.221	0.284	0.975	0.177	Analysis
	0.000	0.000	0.000	0.000	0.000	-0.000	0.762	0.013	worms
Weighted Avg.	0.974	0.013	0.973	0.974	0.972	0.961	0.994	0.973	

=== Confusion Matrix ===

	a	b	c	d	e	f	g	h	i	j	<-- classified as
193788	71	31	502	8	11	5	0	22	0	0	a = Normal
110	5168	113	80	150	374	33	9	8	4	1	b = Exploits
71	425	1195	41	10	47	6	2	0	1	1	c = Reconnaissance
561	285	35	2095	18	49	7	4	0	0	0	d = Attack
29	459	8	22	40833	73	14	5	2	0	0	e = Generic
35	1869	13	20	55	330	14	3	0	0	0	f = DoS
30	33	31	19	12	11	49	0	0	0	0	g = Shellcode
3	257	3	4	9	43	2	6	0	0	0	h = Backdoor
47	211	0	0	2	39	0	0	47	0	0	i = Analysis
0	11	3	0	2	2	0	1	0	0	0	j = worms

Figure 6.1: C4.5 (J48) with multi class segmented dataset of 4000 records.

6.5.1.2 Multilayer perceptron

The MLP consumed much time in training stage to build the model, the highest accuracy was with 4000 and 5000 segmented datasets whilst the 5000 dataset was taken 3990 second and detected 7 classes than the 5000 which was taken 4685 second and detected 6 classes as shown in Table 6.3. Figure 6.2 shows the Confusion matrix and detailed accuracies by classes of MLP with segmented dataset of 5000 records. The least accuracy was 96.46% with segment size 50. The longest time was taken with segment size 1000, it was 5018 Sc.

Table 6.3: Performance evaluation with MLP with multi class datasets.

Seg. Size	Correct Acc.	Incorrect Acc.	TPR	FPR	Pre.	F-M	Time Sec.	No. of classes
50	96.46%	3.54%	0.965	0.027	0.963	0.964	4563	6
500	96.63%	3.37%	0.966	0.025	0.965	0.965	4958	7
1000	96.73%	3.27%	0.967	0.022	0.966	0.966	5018	7
4000	96.78%	3.22%	0.968	0.022	0.966	0.967	4685	6
5000	96.78%	3.22%	0.968	0.022	0.966	0.967	3990	7

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.995	0.027	0.992	0.995	0.993	0.970	0.999	1.000	normal
	0.743	0.013	0.583	0.743	0.653	0.649	0.991	0.716	Exploits
	0.544	0.003	0.577	0.544	0.560	0.557	0.987	0.531	Reconnaissance
	0.545	0.005	0.599	0.545	0.571	0.566	0.987	0.575	Fuzzers
	0.981	0.000	0.999	0.981	0.990	0.988	0.999	0.997	Generic
	0.323	0.006	0.341	0.323	0.332	0.326	0.987	0.316	DoS
	0.016	0.000	0.273	0.016	0.031	0.066	0.988	0.072	Shellcode
	0.000	0.000	0.000	0.000	0.000	0.000	0.980	0.043	Backdoor
	0.000	0.000	0.000	0.000	0.000	-0.000	0.989	0.077	Analysis
	0.000	0.000	0.000	0.000	0.000	0.000	0.986	0.005	worms
Weighted Avg.	0.968	0.022	0.966	0.968	0.967	0.948	0.999	0.974	

=== Confusion Matrix ===

	a	b	c	d	e	f	g	h	i	j	<-- classified as
193385	134	193	715	5	2	1	0	3	0	0	a = normal
307	4494	131	159	10	943	2	0	3	0	0	b = Exploits
145	463	978	101	4	105	2	0	0	0	0	c = Reconnaissance
826	289	176	1663	8	92	0	0	0	0	0	d = Fuzzers
52	491	74	60	40663	101	3	0	1	0	0	e = Generic
46	1449	49	40	0	755	0	0	0	0	0	f = DoS
47	21	87	27	0	0	3	0	0	0	0	g = Shellcode
14	197	4	9	0	103	0	0	0	0	0	h = Backdoor
77	159	0	0	0	110	0	0	0	0	0	i = Analysis
0	15	3	1	0	0	0	0	0	0	0	j = worms

Figure 6.2: MLP with multi class segmented dataset of 5000 records.

6.5.1.3 Naïve bayes

The results of implementing NB classifier with all segmented dataset were relatively close as shown below in Table 6.4. The highest accuracy was 71.10% with 50 segment size and it was less building model time than others, it was 1.64 second. The least accuracy was 70.33% with 1000 segment size. Figure 6.3 shows the Confusion matrix and detailed accuracies by classes of NB with segmented dataset of 50 records.

```

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.691   0.002   0.999     0.691   0.817     0.575   0.928    0.978    normal
      0.050   0.013   0.090     0.050   0.064     0.050   0.724    0.114    Exploits
      0.039   0.009   0.030     0.039   0.034     0.026   0.860    0.049    Reconnaissance
      0.409   0.021   0.196     0.409   0.265     0.271   0.900    0.164    Fuzzers
      0.972   0.228   0.459     0.972   0.623     0.579   0.962    0.900    Generic
      0.612   0.017   0.258     0.612   0.363     0.389   0.894    0.197    DoS
      0.276   0.008   0.024     0.276   0.044     0.079   0.958    0.021    Shellcode
      0.138   0.005   0.035     0.138   0.055     0.067   0.904    0.034    Backdoor
      0.000   0.001   0.000     0.000   0.000     -0.001  0.878    0.020    Analysis
      0.789   0.026   0.002     0.789   0.005     0.042   0.886    0.003    worms
Weighted Avg.  0.711   0.040   0.861     0.711   0.747     0.552   0.928    0.917

=== Confusion Matrix ===

      a      b      c      d      e      f      g      h      i      j  <-- classified as
134319  3031  2091  3718  47226  714   980   15   207  2137 | a = normal
      2    302    51    571    37   1756  168  575    8  2579 | b = Exploits
      15     0    71    246    37    463  428  63   24   451 | c = Reconnaissance
      51     2    74   1250   209   467  423  72   25   481 | d = Fuzzers
      14     4    29   440  40269  267   32   58    6   326 | e = Generic
      3    12    32    75    9   1432  33  421    9   313 | f = DoS
      1     0    10    45    3    41   51    0    7    27 | g = Shellcode
      2     3     4    18    1   222    6   45    0   26 | h = Backdoor
      0     0     0     6    2   194    0   51    0   93 | i = Analysis
      0     0     0     2    0     2    0    0    0   15 | j = worms

```

Figure 6.3: NB with multi class segmented dataset of 50 records.

Table 6.4: Performance evaluation with NB with multi class datasets.

Seg. Size	Correct Acc.	Incorrect Acc.	TPR	FPR	Pre.	F-M	Time Sec.	No. of classes
50	71.10%	28.90%	0.711	0.040	0.861	0.747	1.64	9
500	70.80%	29.20%	0.708	0.040	0.860	0.746	2.17	8
1000	70.33%	29.67%	0.703	0.039	0.859	0.743	1.71	9
4000	70.38%	29.62%	0.704	0.041	0.859	0.744	2.8	9
5000	70.47%	29.63%	0.705	0.039	0.859	0.744	2.44	9

6.5.2 Results of the Second Experiment

In the second experiment we will present the results of implementing MLP, C4.5 and NB classifiers with binary class preprocessed dataset. The results of the implementation of each algorithm appear as follows:

6.5.2.1 Decision tree C4.5 (J48)

The J48 results with different segmented datasets are shown below in Table 6.5. The highest accuracy of correctly classified instances is 99.43% when the

segment size equals to 1000 records; it was needed 153 second for training time as shown in Figure 6.4 the confusion matrix and detailed accuracies. The least accuracy is 98.98% when the segment size equals to 50 records, it was needed 115 Sec for training time.

Table 6.5: Performance evaluation with C4.5 (J48) with binary class datasets.

Seg. Size	Correct Acc.	Incorrect Acc.	TPR	FPR	Pre.	F-M	Time Sec.	No. of classes
50	98.98%	1.02 %	0.990	0.019	0.990	0.990	115	2
500	99.36%	0.64%	0.994	0.012	0.994	0.994	85	2
1000	99.43%	0.57 %	0.994	0.011	0.994	0.994	153	2
4000	99.40%	0.6 %	0.994	0.012	0.994	0.994	159	2
5000	99.35%	0.65%	0.994	0.013	0.994	0.994	186	2

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.996	0.014	0.996	0.996	0.996	0.983	0.998	0.999	Normal
	0.986	0.004	0.988	0.986	0.987	0.983	0.998	0.990	Attack
Weighted Avg.	0.994	0.011	0.994	0.994	0.994	0.983	0.998	0.997	

=== Confusion Matrix ===

a	b	<-- classified as
193753	685	a = Normal
751	54811	b = Attack

Figure 6.4: C4.5 (J48) with binary class segmented dataset of 1000 records.

6.5.2.2 Multilayer perceptron

After MLP were implemented with different segmented datasets, results are shown below in Table 6.6. The MLP consumed much time in training stage to build the model, the less time required was 3582 Sc., the highest accuracy was with 4000 segmented datasets whilst the taken with 4000 dataset was 3866 second. Figure 6.5 shows the Confusion matrix and detailed accuracies by classes of MLP with segmented dataset of 4000 records. The least accuracy is 98.64% with segment size 50. Longer time is taken with segment size 5000, it was 4237 second.

Table 6.6: Performance evaluation with MLP with binary class datasets.

Seg. Size	Correct Acc.	Incorrect Acc.	TPR	FPR	Pre.	F-M	Time Sec.	No. of classes
50	98.64%	1.36%	0.986	0.022	0.987	0.986	4221	2
500	98.92%	1.08%	0.989	0.016	0.989	0.989	3582	2
1000	98.96%	1.04%	0.990	0.018	0.990	0.990	4229	2
4000	99.03%	0.97%	0.990	0.016	0.990	0.990	3866	2
5000	99.00%	1.00%	0.990	0.015	0.990	0.990	4237	2

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.993	0.018	0.995	0.993	0.994	0.972	0.999	1.000	Normal
	0.982	0.007	0.975	0.982	0.978	0.972	0.999	0.998	Fuzzers
Weighted Avg.	0.990	0.016	0.990	0.990	0.990	0.972	0.999	0.999	

=== Confusion Matrix ===

a	b	<-- classified as
193016	1422	a = Normal
1002	54560	b = Fuzzers

Figure 6.5: MLP with binary class segmented dataset of 4000 records.

6.5.2.3 Naïve bayes

The results of implementing NB classifier with all segmented dataset were relatively close in TPR, FPR, Precision and F-Measure ratios as shown below in Table 6.7. The highest accuracy of correctly classified instances is 76.89% when the segment size equals to 1000 records; it was needed 1.5 second for training time as shown in Figure 6.6 the confusion matrix and class detailed accuracies.

Table 6.7: Performance evaluation with NB with binary class datasets.

Seg. Size	Correct Acc.	Incorrect Acc.	TPR	FPR	Pre.	F-M	Time Sec.	No. of classes
50	76.65%	23.35%	0.766	0.165	0.846	0.785	1.5	2
500	76.88%	23.12%	0.769	0.166	0.847	0.787	1.6	2
1000	76.89%	23.11%	0.769	0.166	0.847	0.787	1.5	2
4000	76.88%	23.12%	0.769	0.165	0.847	0.787	2.7	2
5000	76.88%	23.12%	0.769	0.165	0.847	0.787	3.2	2

```

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.743   0.140   0.949     0.743   0.833     0.514   0.931    0.976    Normal
          0.860   0.257   0.489     0.860   0.623     0.514   0.930    0.872    Attack
Weighted Avg.  0.769   0.166   0.847     0.769   0.787     0.514   0.931    0.953

=== Confusion Matrix ===

      a      b  <-- classified as
144448 49990 |      a = Normal
  7787 47775 |      b = Attack

```

Figure 6.6: NB with binary class segmented dataset of 1000 records.

6.6 Results Discussion

The performance results of implementing MLP, C4.5 and NB classifiers with two experiments of both multi class datasets and the binary class dataset were achieved along 50, 500, 1000, 4000, 5000 records of segment size will be discussed in this part.

1. The results of this model proved it is considered an efficient method to use TF.IDF in transforming datasets of Intrusion detection system.
2. Different segment sizes in both binary and multi class IDS datasets returned different accuracies; Table 6.8 shows the increase in accuracy of every classifier for each dataset. The increase in accuracy reflects the efficiency of transforming dataset with TF.IDF with various segment sizes.

Table 6.8: Highest and lowest detection accuracy and the increase in accuracy.

ML	Dataset Class Type	Highest	Seg. Size	Lowest	Seg. Size	Inc.
C4.5	Multi	97.40%	4000	96.8%	50	0.6%
	Binary	99.43%	1000	98.98%	50	0.45%
MLP	Multi	96.78%	4000+5000	96.46%	50	0.32%
	Binary	99.03%	4000	98.64%	50	0.39%
NB	Multi	71.10%	50	70.33%	4000	0.77%
	Binary	76.89%	1000	76.65%	50	0.24%

1. Building IDS model with different ML techniques returns different detection indicators.

2. C4.5 classifier achieved the highest accuracy, highest TPR and the lowest FPR with UNSW-NB15 multi class dataset. The accuracy was 97.40%, the TPR was 97.40% and FPR was 1.3%. Figure 6.7 illustrates the comparison between C4.5 and other classifiers.
3. NB was distinguished than the other classifiers in building model time. It consumed the lowest time between 3.2 second and 1.5 second but the model performance dropped down. It achieved the lowest accuracies with an effective detection ratio of classes.

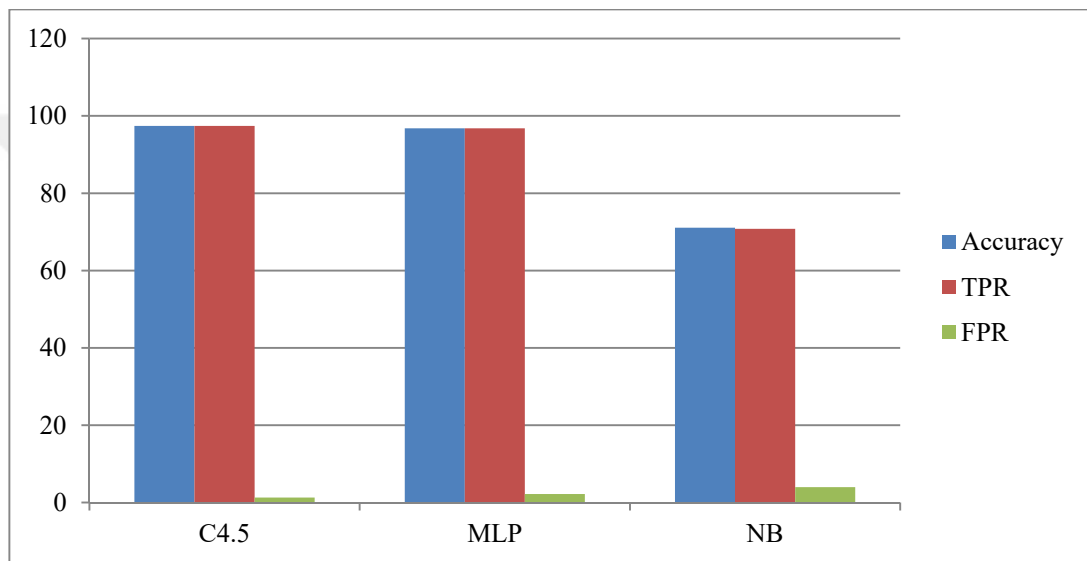


Figure 6.7: Comparison between C4.5 and other classifiers with multi class datasets.

4. C4.5 classifier achieved the highest accuracy, highest TPR and the lowest FPR with UNSW-NB15 binary class dataset. The accuracy was 99.43%, the TPR was 99.4% and FPR was 1.1%. Figure 6.8 illustrates the comparison between C4.5 and other classifiers.

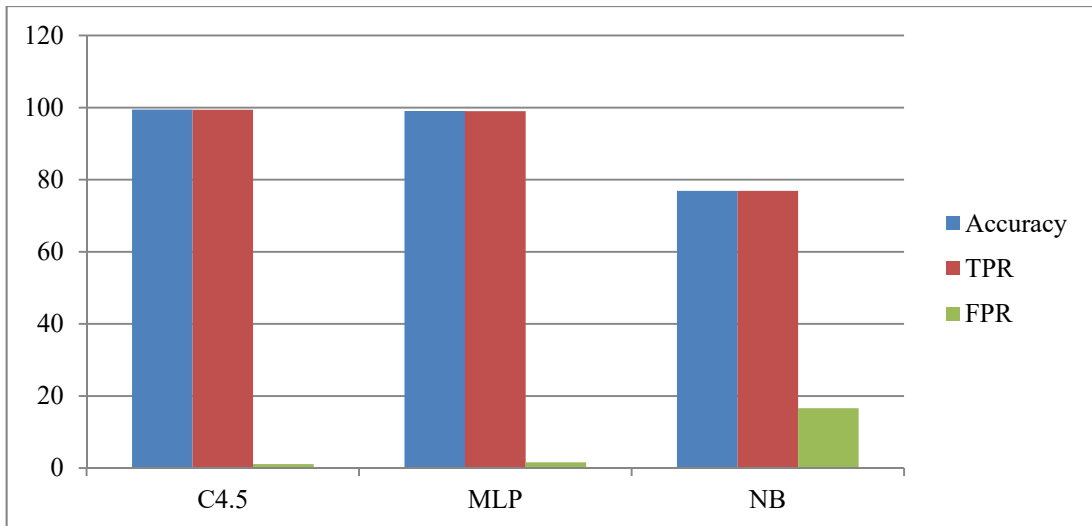


Figure 6.8: Comparison between C4.5 and other classifiers with binary class datasets.

5. Comparatively MLP has a good accuracy exactly with the dataset that its segment size equals to 4000 records, but it didn't reach to C4.5 in accuracy and number of detecting classes. MLP was able to detect between 6 and 7 classes from 10.

6. The ML algorithms did not object about the type of data used, which are originally containing missing values, characters and time range values, but on the contrary, the results were given indicates that the data has been transformed into an effective form in dealing with the ML algorithms.

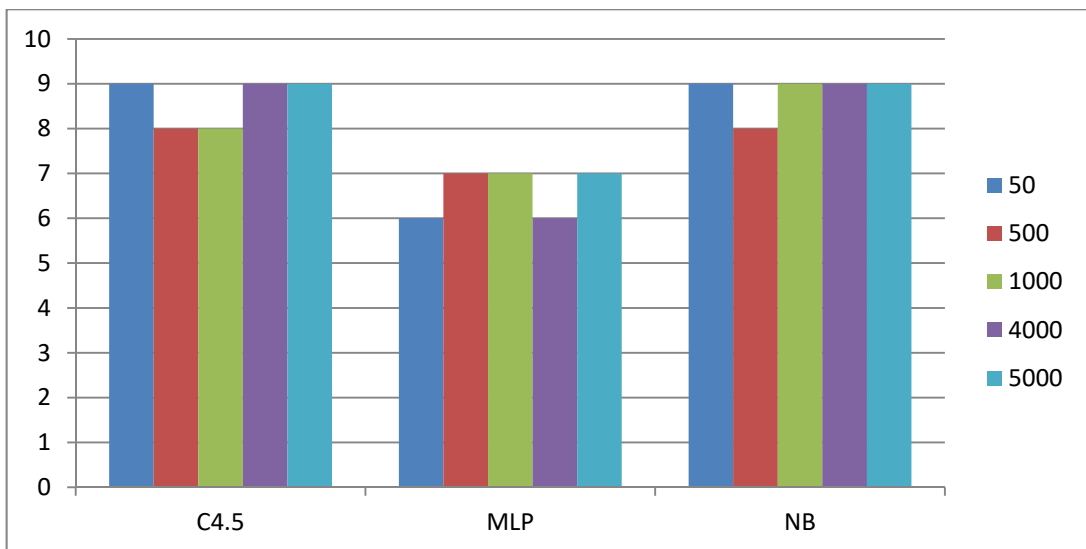


Figure 6.9: Each classifier with different segmented dataset and the number of detecting classes.

7. No one of all experiment was able to detect 10 classes of multi datasets, as illustrate in Figure 6.9 which shows each classifier with different segmented dataset and the number of detecting classes, because there are some attacks appear rarely in the UNSW-NB15 dataset, such as Worms attack appears 174 times than the overall dataset is 250044 records. In our randomly selected dataset which contains 250000 records there are only 19 records of Worms attacking class. We can describe this dataset as imbalanced dataset.

8. Because the property of the similarities between the values of UNSW-NB15 dataset records showed that, the ML cannot detect several records categories [6]. This property has been reduced by using TF.IDF for transforming dataset records; evidence of this is the high accuracy detection results.

6.7 Comparisons With Previous Works

In the work presented by Amar Agrawal et al. [4] proposed a hybrid an IDS including two stages first one, misuse detection model by using a Binary Tree Classifier for detecting only known attacks and the second stage for anomaly detection model based on SVM Classifier. The hybrids IDS has been experimented and evaluated using benchmark UNSW-NB15. The model was compared with ZeroR, Naive Bayes and C4.5 (J48). The highest accuracy was obtained with decision tree J48 87.52% with UNSW-NB15 dataset while the proposed mode results 88.55%.

Hossein Gharaee and Hamid Hosseinvand [5] presented an IDS used a new feature selection model to reduce the dimension of UNSW-NB15 through using the discriminating properties of Genetic algorithm. They used support vector machine as detection algorithm. the results of UNSW-NBI5 dataset were 97.45 % with Normal class, 96.39 % Fuzzers attack, 91.55 % Reconnaissance attack, 99.45 % Shellcode attack, 91.24 % DoS attack, 79.19 % Exploits attack and 97.51 % Generic attack. The average accuracy of these results equal to 93.24%, In spite of they avoid using all dataset classes. They tested only 6 attack types and one normal class from 10 overall classes in the UNSW-NBI5 dataset. Because there are some attacks appear rarely in the UNSW-NB15 dataset, such as Worms, Analyses and Backdoors attacks. In their research they extracted 30 attributes from 47.

In our work we applied classifiers with all classes belongs to dataset, the results of our model has higher accuracy, higher TPR and lower FPR with C4.5 (J48) algorithm as shown in Table 6.9 below.

Table 6.9: Comparison between our work and Amar Agrawal et al. [4] with Hossein Gharaee and Hamid Hosseinvand [5].

	Acc.%	TPR%	FPR%
Our model	97.40	97.4%	1.3%
Hossein Gharaee and Hamid Hosseinvand	93.24	90.74	4.42%
Amar Agrawal et al.	88.55%.	N/A	N/A



CHAPTER SEVEN

CONCLUSIONS AND FUTURE WORK

7.1 Introduction

This chapter will display the conclusions of our model and proposed future work.

7.2 Conclusion

In this thesis, we developed an IDS model based on decision tree C4.5 algorithm, to solve the intrusion detection problem. The model transforms the dataset as a step of the important preprocessing steps to convert data types to an efficient form can enhance machine learning performance. Transformation data uses term frequency – inverse document frequency (TF.IDF) to achieve a statistical measure by weighting the dataset items to evaluate the importance of a word as a transformation approach. The value of each item in the dataset will be replaced by the weight of the item. The model is tested with randomly selected 250000 records of the UNSW-NB15 dataset, then creating various datasets, each of which has a specific segment size as 50, 500, 1000, 4000 and 5000 records. The single dataset is considered a corpus and each segment is considered a document. In addition the same datasets was characterized as two subsets of multi and binary of normal and attack class datasets. Only 30 attributes from the UNSW-NB15 dataset which has 47 attributes was exploited. We have compared the performance of Multilayer Perceptron and Naive Bayes as various classification techniques with decision tree algorithm C4.5 that is used in this model by using Weka software. Training and testing stage have been achieved with 10 fold cross validation. From the results of detection indicators we concluded that the C4.5 classifier achieved the highest accuracy 97.40%, highest

TPR 97.40% and the lowest FPR 1.3% with 4000 segment size of multi class dataset, with the same classifier, but with 1000 segment size of the binary class dataset the highest accuracy 99.43%, highest TPR 99.4% and the lowest FPR 1.1%. Different segment sizes increase the accuracy of C4.5 in both binary to 0.6% and multi class to 0.45% which leads to return different accuracies. The increase in accuracy reflects the efficiency of transforming dataset with TF.IDF with various segment sizes which leads to support effective solution in distinguishing malicious from normal network packets. TF.IDF as transformer results high efficiency in this model because the effectiveness in decision tree algorithm C4.5 in dealing with continuous value attributes and mitigate over fitting of the data by pruning. For future work, it would be interesting to test another ML which they are well known with IDSs such as Random Forest or Random Tree with different segment sizes with UNSW-NB15 dataset or with another one.

REFERENCES

- [1] Armin, J., et al. *2020 cybercrime economic costs: No measure no solution*. in *availability, reliability and security (ares), 2015 10th international conference on*. 2015. IEEE.
- [2] Moustafa, N. and J. Slay. *UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)*. in *Military Communications and Information Systems Conference (MilCIS), 2015*. 2015. IEEE.
- [3] Mehmood, T. and H.B.M. Rais. *Machine learning algorithms in context of intrusion detection*. in *Computer and Information Sciences (ICCOINS), 2016 3rd International Conference on*. 2016. IEEE.
- [4] Agrawal, A., S. Mohammed, and J. Fiaidhi, *Developing Data Mining Techniques for Intruder Detection in Network Traffic*. *International Journal of Security and Its Applications*, 2016. **10**(8): p. 335-342.
- [5] Gharaee, H. and H. Hosseinvand. *A new feature selection IDS based on genetic algorithm and SVM*. in *Telecommunications (IST), 2016 8th International Symposium on*. 2016. IEEE.
- [6] Moustafa, N. and J. Slay. *The significant features of the UNSW-NB15 and the KDD99 data sets for Network Intrusion Detection Systems*. in *Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2015 4th International Workshop on*. 2015. IEEE.
- [7] Garg, T. and S.S. Khurana. *Comparison of classification techniques for intrusion detection dataset using WEKA*. in *Recent Advances and Innovations in Engineering (ICRAIE), 2014*. 2014. IEEE.
- [8] Mane, V.D. and S. Pawar, *Anomaly based IDS using Backpropagation Neural Network*. *International Journal of Computer Applications*, 2016. **136**(10): p. 29-34.
- [9] Deshmukh, D.H., T. Ghorpade, and P. Padiya. *Intrusion detection system by improved preprocessing methods and Naïve Bayes classifier using NSL-KDD*

99 Dataset. in *Electronics and Communication Systems (ICECS), 2014 International Conference on.* 2014. IEEE.

- [10] Singh, V. and S. Puthran. *Intrusion detection system using data mining a review.* in *Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), 2016 International Conference on.* 2016. IEEE.
- [11] Zhao, Y. *Network intrusion detection system model based on data mining.* in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2016 17th IEEE/ACIS International Conference on.* 2016. IEEE.
- [12] Mogal, D.G., S.R. Ghungrad, and B.B. Bhusare, *NIDS using Machine Learning Classifiers on UNSW-NB15 and KDDCUP99 Datasets.* *Ijarcce*, 2017. **6(4)**: p. 533-537.
- [13] Dadgar, S.M.H., M.S. Araghi, and M.M. Farahani. *A novel text mining approach based on TF-IDF and Support Vector Machine for news classification.* in *Engineering and Technology (ICETECH), 2016 IEEE International Conference on.* 2016. IEEE.
- [14] Bhattacharyya, D.K. and J.K. Kalita, *Network anomaly detection: A machine learning perspective.* 2013: CRC Press.
- [15] Stallings, W. and M.P. Tahiliani, *Cryptography and network security: principles and practice.* Vol. 6. 2014: Pearson London.
- [16] Roshani Gaidhane, S., Prof. C. Vaidya, Dr. M. Raghuwanshi, *Survey: Learning Techniques for Intrusion Detection System (IDS).* *International Journal of Advance Foundation and Research in Computer (IJAFRC)* 2014. **Volume 1**(Issue 2): p. 8.
- [17] Soniya, S.S. and S.M.C. Vigila. *Intrusion detection system: Classification and techniques.* in *Circuit, Power and Computing Technologies (ICCPCT), 2016 International Conference on.* 2016. IEEE.
- [18] Canavan, J.E., *Fundamentals of network security.* 2001: Artech House.
- [19] Yu, Z. and J.J. Tsai, *Intrusion detection: a machine learning approach.* Vol. 3. 2011: World Scientific.
- [20] Garzia, F., *Handbook of Communications Security.* 2013: WIT Press.

- [21] Taluja, M.S. and R.L. Dua, *Survey on Network Security, Threats & Firewalls*. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2012. 1(7): p. pp: 53-58.
- [22] Bijone, M., *A Survey on Secure Network: Intrusion Detection & Prevention Approaches*. American Journal of Information Systems, 2016. 4(3): p. 69-88.
- [23] Kizza, J.M., *Guide to computer network security*. 2009: Springer.
- [24] <https://www.symantec.com/connect/articles/evolution-intrusion-detection-systems>.
- [25] Subaira, A. and P. Anitha. *Efficient classification mechanism for network intrusion detection system based on data mining techniques: a survey*. in *Intelligent Systems and Control (ISCO), 2014 IEEE 8th International Conference on*. 2014. IEEE.
- [26] Rhodes-Ousley, M., *The Complete Reference™ Information Security*, ed. S. Edition. 2013: The McGraw-Hill Companies.
- [27] *Ádám, N., et al. Artificial neural network based IDS*. in *Applied Machine Intelligence and Informatics (SAMi), 2017 IEEE 15th International Symposium on*. 2017. IEEE.
- [28] Dua, S. and X. Du, *Data mining and machine learning in cybersecurity*. 2016: CRC press.
- [29] Zaki, M.J., W. Meira Jr, and W. Meira, *Data mining and analysis: fundamental concepts and algorithms*. 2014: Cambridge University Press.
- [30] Aggarwal, C.C., *Data mining: the textbook*. 2015: Springer.
- [31] Han, J., J. Pei, and M. Kamber, *Data mining: concepts and techniques*. 2011: Elsevier.
- [32] FABIO, M.P., C.E. DE LA HOZ, and M. ALEXIS, *APPLICATION OF FEAST (FEATURE SELECTION TOOLBOX) IN IDS (INTRUSION DETECTION SYSTEMS)*. Journal of Theoretical and Applied Information Technology, 2014. 70(3).
- [33] Alpaydın, E., *Introduction to Machine Learning*. Second Edition ed. 2010: The MIT Press Cambridge, Massachusetts London, England.

- [34] Desale, K.S. and R. Ade. *Genetic algorithm based feature selection approach for effective intrusion detection system*. in *Computer Communication and Informatics (ICCCI), 2015 International Conference on*. 2015. IEEE.
- [35] Witten, I.H., et al., *Data Mining: Practical machine learning tools and techniques*. 2016: Morgan Kaufmann.
- [36] Bell, J., *Machine learning: hands-on for developers and technical professionals*. 2014: John Wiley & Sons.
- [37] Sharma, R.K., H.K. Kalita, and P. Borah. *Analysis of machine learning techniques based intrusion detection systems*. in *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*. 2016. Springer.
- [38] Jain2, U.M.a.A., *AN IMPROVED METHOD TO DETECT INTRUSION USING MACHINE LEARNING ALGORITHMS*. Informatics Engineering, an International Journal (IEIJ), 2016. **Vol.4, No.2, June 2016**.
- [39] Rafsanjani, M.K. and Z.A. Varzaneha, *Intrusion Detection By Data Mining Algorithms: A Review*. Journal of New Results in Science, 2013. **2(2)**.
- [40] R. Revathy, R.L., *Comparative Analysis of C4.5 and C5.0 Algorithms on Crop Pest Data*. International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization), 2017. **Vol. 5, Special Issue 1, March 2017**.
- [41] Dipali G. Mogall, S.R.G., Bapusaheb B. Bhusare, *A Review on High Ranked Features based NIDS*. International Journal of Advanced Research in Computer and Communication Engineering, 2017. **Vol. 6, Issue 3, March 2017**.
- [42] <https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets/>.
- [43] Allen, W.H., G.A. Marin, and L.A. Rivera. *Automated detection of malicious reconnaissance to enhance network security*. in *SoutheastCon, 2005. Proceedings. IEEE*. 2005. IEEE.
- [44] Knudsen, J., *Practical Considerations Of Fuzzing: Generating Insight into Areas of Risk*. Biomedical instrumentation & technology, 2014. **48(s1)**: p. 48-53.

- [45] Arce, I., *The shellcode generation*. IEEE Security & Privacy, 2004. **2**(5): p. 72-76.
- [46] Alminshid, K. and M.N. Omar. *Detecting backdoor using stepping stone detection approach*. in *Informatics and Applications (ICIA), 2013 Second International Conference on*. 2013. IEEE.
- [47] <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>.
- [48] <http://www.cs.waikato.ac.nz/ml/weka/index.html>.
- [49] Deshmukh, D.H., T. Ghorpade, and P. Padiya. *Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset*. in *Communication, Information & Computing Technology (ICCICT), 2015 International Conference on*. 2015. IEEE.
- [50] Katkar, V.D. and D.S. Bhatia. *Lightweight approach for detection of denial of service attacks using numeric to binary preprocessing*. in *Circuits, Systems, Communication and Information Technology Applications (CSCITA), 2014 International Conference on*. 2014. IEEE.
- [51] Schütze, C.D.M.P.R.H., *An Introduction to Information Retrieval*. 2009: Cambridge University Press Cambridge, England.
- [52] Leskovec, J., A. Rajaraman, and J.D. Ullman, *Mining of massive datasets*. 2014: Cambridge university press.
- [53] Hagos, D.H., et al. *Enhancing Security Attacks Analysis Using Regularized Machine Learning Techniques*. in *Advanced Information Networking and Applications (AINA), 2017 IEEE 31st International Conference on*. 2017. IEEE.
- [54] Belouch, M., S. El Hadaj, and M. Idhammad, *A Two-Stage Classifier Approach using RepTree Algorithm for Network Intrusion Detection*. INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, 2017. **8**(6): p. 389-394.
- [55] Hssina, B., et al., *A comparative study of decision tree ID3 and C4. 5*. International Journal of Advanced Computer Science and Applications, 2014. **4**(2): p. 13-19.

- [56] Bashir, U. and M. Chachoo, *PERFORMANCE EVALUATION OF J48 AND BAYES ALGORITHMS FOR INTRUSION DETECTION SYSTEM*.
- [57] Kumar, X.W.a.V., *The Top Ten Algorithms in Data Mining*, ed. V. Kumar. 2009, United States of America: Chapman & Hall/CRC Taylor & Francis Group.
- [58] Relan, N.G. and D.R. Patil. *Implementation of network intrusion detection system using variant of decision tree algorithm*. in *Nascent Technologies in the Engineering Field (ICNTE), 2015 International Conference on*. 2015. IEEE.
- [59] 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).
- [60] Goeschel, K. *Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis*. in *SoutheastCon, 2016*. 2016. IEEE.
- [61] Mukherjee, S. and N. Sharma, *Intrusion detection using naive Bayes classifier with feature reduction*. *Procedia Technology*, 2012. 4: p. 119-128.
- [62] Choudhury, S. and A. Bhowal. *Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection*. in *Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on*. 2015. IEEE.
- [63] Panda, M. and M.R. Patra. *A comparative study of data mining algorithms for network intrusion detection*. in *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on*. 2008. IEEE.
- [64] Kumar, S. and A. Yadav. *Increasing performance of intrusion detection system using neural network*. in *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*. 2014. IEEE.
- [65] Gadai, S.M.A.M. and R.A. Mokhtar. *Anomaly detection approach using hybrid algorithm of data mining technique*. in *Communication, Control, Computing and Electronics Engineering (ICCCCEE), 2017 International Conference on*. 2017. IEEE.
- [66] Elhamahmy, M., H.N. Elmahdy, and I.A. Saroit, *A New Approach for Evaluating Intrusion Detection System*. *CiiT International Journal of Artificial Intelligent Systems and Machine Learning*, 2010. 2(11).

CURRICULUM VITAE

PERSONAL DETAILS AND CONTACT DETAILS



Name : Khaldoon Ali Hammood Awadh.
Date of Birth : June. 23, 1978.
Title : Deputy Prime Programmers.
Address : 18, Al karama Q., Hillah City, Babylon Province, Iraq.
Tel : 009647809398888
Email : khaldoonawadh@gmail.com
Occupation : Employee in the General Directorate of Electricity Distribution for Middle Euphrates – Ministry of Electricity – 60 St. Hillah city, Babylon Province, Iraq.

EDUCATION AND QUALIFICATIONS

From (Oct. 1, 2001) to (Dec. 21, 2002): Higher Diploma in Computer Science / Data Security, Informatics Institute for Postgraduate Studies – Iraqi Commission for Computers & Informatics.

From (Oct. 1, 1997) to (July. 10, 2001): BSc in Computer Science, College of Science – Babylon University.

RELEVANT WORK

From (Sept. 19, 2012) to (Dec. 31, 2014): Head of Informatics Department – General Directorate Mentioned Above.

From (Jan. 24, 2012) to (Sept. 19, 2012): Deputy Prime Programmers in Training Department – General Directorate Mentioned Above.

From (Sept. 16, 2009) to (Jan. 24, 2012): Head of the Maintenance Computers and Internet Section – Informatics Department – General Directorate Mentioned Above.

From (Aug. 24, 2009) to (Sept. 16, 2009): Head of Systems and Software Section – Informatics Department – General Directorate Mentioned Above.

From (Aug. 2, 2004) to (Aug. 24, 2009): Head of Computer and Internet Section – Communication and Computer Department – General Directorate Mentioned Above.

PUBLICATION

Khaldoon Ali Hammood Awadh, "Availability Cache Level 1 and Level 2", Higher Diploma in Computer Science / Data Security thesis Submitted to Informatics Institute for Postgraduate Studies – Iraqi Commission for Computers & Informatics, 2002.

