Islamic University – Gaza

Deanship of Graduate Studies

Faculty of Information Technology

Adaptive Worms Detection Model Based on Multi Classifiers

Prepared by

Hanaa A. Qeshta

Supervised by

Dr. Tawfiq S. Barhoom

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science In Information Technology

2012-1433 H


بسم الله الرحمن الرحيم

Dedication

To the soul of my father

To my beloved mother

To sisters and brothers

To all friends

Acknowledgements

Praise is to Allah, the Almighty for having guided me at every stage of my life.

This thesis is the result of years of work whereby I have been accompanied and supported by many people. It is wonderful that I now have the opportunity to express my gratitude to all of them.

This work would not have been possible without the constant encouragement and support I received from Dr. Tawfiq S. Barhoom, my advisor and mentor. I would like to express my deep and sincere gratitude to him. His understanding and personal guidance have provided a good basis for the present thesis.

Special thanks to Dr. Ayesha Binte Ashfaq from NUST School of Electrical Engineering and Computer Science, Islamabad, for his provide assistance and guidance.

I would like to thank Mr. Motaz Saad, for his valuable scientific and technical notes, also I extend my thanks to Eng. Ahmed Al-Astal for reviewing my thesis.

Also, I would like to take this opportunity to express my profound gratitude to my beloved family - my mother, brothers and sisters - without whom I would ever have been able to achieve so much.

I offer my thanks and appreciation to all of those who supported me in any respect during the completion of the research.

Lastly, but certainly not least, I want to thank my friends, for their moral support during this study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

IDS	Intrusion Detection System
HIDS	Host-Based Intrusion Detection System
NIDS	Network-Based Intrusion Detection System
WD	Worms Detection
MCS	Multi-classifier system
NB	Naïve Bayes
DT	Decision Tree
ANN	Artificial Neural Networks
BPNN	Back propagation NN algorithm
SVMs	Support vector machines
RI	Rule Induction
K-NN	K-Nearest Neighbor
RF	Random Forest
BN	Bayesian Network
WDMAC	**A**daptive **W**orms **D**etection Based **M**ulti **C**lassifiers (our proposed model)

نموذج متكيف لكشف ديدان الحاسوب
بالاعتماد على التصنيفات المتعددة

في الآونة الأخيرة، مع وجود وزيادة استخدام الشبكات والإنترنت، أصبح لأمن الشبكات مكاناً لاهتمام الباحثين، حيث التهديدات بكافة أشكالها تمثل قضية مهمة وصعبة لأنظمة الشبكات، ومصدراً للقلق. تشكل هجمات الديدان تهديدات أمنية كبيرة على أمن الشبكات والإنترنت، حيث تعتبر الديدان سبباً في الكثير من المشاكل. باستخدام الأساليب التقليدية "الكشف عن سوء استخدام" للكشف عن الديدان من خلال توقيعاتهم، حيث كانت هذه الطريقة غير قادرة على كشف الديدان الغير معروفة التوقيعات قبل ظهور توقيعاتهم. لذلك الكشف عن الديدان، والديدان الجديدة غير معروفة التوقيعات ما زالت قضية مهمة وصعبة. التركيز في الابحاث على اكتشاف الديدان يتحول من استخدام نهج كشف إساءة استخدام "أنماط التوقيع" إلى استخدام نهج كشف السلوك الشاذ "تحديد سلوك ضار". وبالإضافة إلى ذلك استخدام مصنفات "كشف السلوك الشاذ" المستخدمة بشكل فردي ومستقل في أنظمة كشف الديدان غير قادرة للوصول إلى درجة دقة مقبولة في العالم الحقيقي للانتشار. لذلك، فإن طريقة الجمع المركب لعدد من المصنفات مفيدة بشكل خاص للمشاكل الصعبة، ولتحقيق مستوى عالي من الدقة ومعدلات الكشف وتقليل معدل التصنيف الخطأ. في هذا البحث، اقترحنا استخدام عدد من المصنفات مثل (NB, DT, and ANN) بشكل المصنفات المتعددة لكي تكون قادرة على التكيف للكشف عن الديدان المعروفة والغير معروف معتمدة على نهج الكشف عن السلوك الشاذ، لكي نحقق أعلى معدل للدقة والكشف عن الديدان، وانخفاض معدل التصنيف الخطأ. وقد اوضحت نتائجنا إلى أن النموذج المقترح قد حققت أعلى معدلات الكشف ودقة التصنيف، حيث كشف عن الديدان المعروفة بمعدل لا يقل عن 98.30 ٪، و معدل التصنيف الخطأ كان 1.70 ٪، في حين أن معدل الكشف عن دودة الغير معروفة حوالي 98.05٪، ومعدل التصنيف الخطأ كان 1.95 %.

كلمات مفتاحية: الديدان، تنقيب البيانات، التصنيفات المتعددة، نظام كشف التسلل، كشف سوء استخدام، كشف السلوك الشاذ، BN، DT، ANN.

Adaptive Worms Detection Model Based on Multi Classifiers

Abstract

In recent times, the networks security has become a place of interest of researchers, threats has become an issue for each networks system, and a source of concern. Worms attacks have been major security threats to networks and the internet, which are cause many of problems. Using traditional approaches "misuse detection" to detect worms through their signatures unable to detect unknown worms before the appearance of their signatures. So Detecting worms, especially new and unknown worms is still a challenging task. The focus of worm detection research is shifting from using misuse detection " signature patterns " to anomaly detection " identifying the malicious behavior ". In addition standalone anomaly classifiers used by anomaly worms detection systems are unable to access acceptable accuracies in real-world deployments. Therefore, the combination method is particularly useful for difficult problems, and to achieve higher accuracies and detection rates, and rising classification error. In this research, we proposed using data mining techniques by combination of classifiers (Naïve Bayes, Decision Tree, and Artificial Neural Network) as in multi classifiers to be able adaptive for detecting known/ unknown worms depend on behavior-anomaly detection approach, to achieve higher accuracies and detection rate, and lower classification error rate. The results show that the proposed model has achieved higher accuracies and detection rates of classification, where detection known worms are at least 98.30%, with classification error rate 1.30%, while the unknown worm detection rate is about 98.05%, with classification error rate 1.95%.

Keywords: Worms, Data Mining, Multi Classification, Misuse Detection, Behavior-Anomaly Detection, Naïve Bayes, Decision Tree, Artificial Neural Network.

CHAPTER 1: Introduction

1.1 Introduction

There is no doubt that while the internet is widely growing and more and more network services are being created, the number of the internet users is extremely increasing. The internet, as no other communication medium, has given an international dimension to the world. It has become the universal source of information for millions of people, at home, at school, and at work.

As a result of this growth and expansion, networks' threats and internet security attacks have become a very important area of research. More specifically, worms have been stated as the major source of threats related to internet and networks security [25]. In recent years, many studies has been proposed to detect worms as a misuse detection by using signatures of worms. This approach is unable to reach zero-day state and hence is not effective in protecting networks against the current deployment of worms and the speed at which they spread. On the other hand, another approach has been proposed to detect worms based on anomaly behavior detection where it is possible to detect abnormal behaviors and generate alarms, this approach is useful to detect unknown attacks. Also, current worm defense begins with manual worm detection followed by repairing the damage.

Definition 1.1: *Worms* are a form of malware, or malicious piece of code where a self-duplicating and self-propagating spreads itself from computer to computer across network without any human interference , and starts destructive attacks against computer networks, in order to cause huge damages to the network [8][24].

Definition 1.2: *Unknown worms,* defined by researchers as new worms where it's signature is unknown before it is seen and discovery .

Definition 1.3: *Known worm,* defined by researchers as worms become known where it's signature is known, after it is seen and discovery.

Definition 1.4: *Worms Detection systems (WDS),* defined by researchers as WDS has become vital to the field of Internet and network security, where considered the approaches used to early detect known/unknown worms. Generally, worm detection can be classified into network based and host-based detection. Where network-based detect worm attacks by monitoring, collecting, and analyzing worm generated traffic. Host-based detect worm attacks by monitoring, collecting, and analyzing the worm behavior on end-hosts [15].

Definition 1.5: *Zero-day attack,* exploit is one that takes advantage of a security vulnerability on the same day that the vulnerability becomes generally known. There are zero days between the time the vulnerability is discovered and the first attack [40].

Definition 1.6: *Adaptive worm detection system (AWDS),* defined by researcher as systems able to detect known and unknown worms, even if those worms change their behavior or change their effect on the infected network traffic.

Definition 1.7: *Multi-classifier system(MCS)* combines the outputs from several individual classifiers to generate a final prediction score [17].

Few systems have been developed to detect unknown worms' attacks. Those systems are usually crippled by many factors such as the amount of traffics of the network. Such a way that aims to automatically detect unknown worms is particularly challenging, because it is difficult to predict what form the next worm to be detected. Hence, automatic detection of unknown worms and fast response to changes have become a critical issue, in order to be able to cope with newly released worms which can infect millions of hosts in a matter of seconds.

Several types of machine learning techniques were used in the field of intrusion detection in general and in detecting worms in particular. Data Mining has an important role and is essential in worms detection systems, where several of worms detection is mainly based on data mining techniques [13].

We proposed “WDMAC” model that an adaptive worm detection model based on multi classifiers is capable of being adaptive to detect known/unknown worms with highest detection rate and lowest classification error.

1.2 Goals of Worms Detection systems:

- Protecting networks from worms and decreasing the effect of damage caused by different and unknown types of worms.
- To overcome the shortage of traditional approach of worm detection, known as "misuse detection", which cannot detect unknown/new worms without extracting the signature by experts .
- To overcome the challenge of anomaly-based detection systems which is defining what normal network behavior is, deciding the threshold to generates the alarms, and prevention of false alarms.

- To enhancement of standalone classifiers that cannot provide acceptable accuracies in real-world deployments
- To introduce an adaptive detection model that has the ability of detecting known and unknown worms in early stages.
- Improving the detection rate, and reducing the misclassification rate.
- Introducing a new way of integration of classification methods by combining them in order to achieve an acceptable accuracy in real world.

1.3 Network Security and Worms Detection

Worms are simply designed to cause damages to a computer system as much as possible. Network access authorization is the procedure of granting/ preventing privileges to access network, whether this access takes the form of destruction, normal use, disclosure, modification, or disruption. Network security and worms' detection are interconnected and share common services in order to save and keep confidentiality, integrity and availability of the network.

In worm's detection, different approaches of worm's detection are used to protect from worm's risks and penetration dangers caused after the network penetration from theft or destruction, damage.

1.4 Research Motivation

Using a computer connected to a large network making us at risk of a worm infection. Worms can be dangerous, so it's important to know how to protect our computers from different kinds of worms.

The continuous increase of dangers, threats, and attacks methods with the increase of growth of networks systems, led to an increasing in challenges of the security issues related to networks systems. Since worms are one of the top malicious threats where there are hundreds of worms appear on a daily basis across networks in a matter of seconds, there is an urgent need to develop and propose new and effective approaches to detect worms.

There is traditional detection approach known as *"misuse detection"* which uses worm's signatures to detect the worms. But this method failed to reach the state of zero-day attacks. Another approach known as *"anomaly-behavior detection"*, depends on the detection of unusual behaviors and generate alerts. In our research, we adopted on anomaly detection

because of the need to reach the maximum protection from the risk of worms, and access to zero-day case.

1.5 Problem Statement

Effectiveness of worms' detection systems depends on the ability of early detecting of worms. Most of recent researches on worms' detection can be classified into three categories: researches that tend to use one type of classifiers independently , researches that use a number of classifiers and compare them to choose the best results, and researches that use some of classifiers method, and extract the average of classifiers accuracies to determine the percentage of detection. These methods are unable to achieve an acceptable and sufficient accuracy and are ineffective in a risky real world. So There is an urgent need to have a worm detection model that is capable of being adaptive to detect known/unknown worms early before the damages occur, and to provide protection from its dangers and access to zero-day worms attacks, with highest detection rate and lowest classification error.

1.6 Research Objectives

Recent researches have shown that standalone anomaly classifiers used by anomaly detection systems are unable to give acceptable accuracies in real-world deployments. Therefore, the combination method is particularly useful for difficult problems, and is more likely to achieve higher accuracies. Even though new anomaly detection systems have been developed based on using multiple classifiers, where the outputs are combined to formulate an aggregated anomaly score, choosing the number and the types of classifiers to be used and combined is still challenging [17]. Moreover, using the concept of multi classifiers can be time and effort consuming.

Driven by these challenges, we proposed the concept of using multi classifiers. By using this type of classifiers, it should be possible to detect known/ unknown worms depend on behavior-anomaly detection approach, the accuracy and detection rate should be improved, and error rate is expected to be decreased.

1.6.1 Main Objective

The main objective of this research is to propose (WDMAC) which is an adaptive worms detection model based on anomaly-behavior detection that can detect known and unknown worms by using multi classifiers in order to achieve an acceptable accuracy .

1.6.2 Specific Objectives

There are many specific objectives extracted from the main objective:

- Identifying the different types of worms, their way of spreading, the phases of the worms' life, and the extent of its effect on the behavior of network traffic. These factors help us to extract important characteristics and are important for building "WDMAC" model.
- Applying worms' detection model based on anomaly-behavior detection using supervised learning machine technique and by combination of multi classifiers in data mining, so that to be able to detect both known and unknown worms.
- Training "WDMAC" model on normal network behavior, to be able to predict the behavior of non-normal, and to be effective in protecting the network and detect known/ unknown worms' attacks.
- Testing "WDMAC" model on new untested network behavior, to observe the system's ability to detect this behavior, so that we can prove that this model is able to adapt and to detect known/ unknown worms.
- Trying to test various behaviors of "WDMAC" model and evaluating the results.
- Reduce the false positive and negative rate, and improve the detection rate through the measurement and evaluation by using programs and tools.
- Improve network security and protecting them from threats of worms.
- Introducing a new way of detecting zero-day worms' attacks.
- Compare "WDMAC" model with other existing models.

1.7 Research Scope and Limitation

This research aims to propose an adaptive model for worms detection by combination of multi classifiers which is able to detect known / unknown worms with high detection rate, high accuracy, and low false rate. This work is applied with some limitations and assumption such as:

- The "WDMAC" model based on host intrusion detection system (HIDS).
- The "WDMAC" model is built using behavior anomaly detection technique.
- The "WDMAC" model is limited for supervised learning with single class label.
- Using combination of multi classification techniques in data mining to detect worms.

- "WDMAC" model work depends on direct worms.
- "WDMAC" model work depended on internet worms.
- The data sets used in this research are collected from "Wireless and Secure Networks (WiSNet) Research Lab at the NUST School of Electrical Engineering and Computer Science (SEECS)"in real time.
- The Use of specific worms such as Blaster, CodeRed2, Forbot-FU, Rbot.CCC, and Zotob.G,.

1.8 Significance of the research

- Add a significant contribution to scientific research in the field of finding effective solutions in worms' detection.
- Helping concerned people working in various domains that have worm's detection to get a better prediction for classification.
- Using more classification techniques as combination to achieve misclassification rate close to zero.

1.9 Research Methodology

In our research, we devote our study on detecting known/ unknown worms. In "WDMAC" model we will use adaptive supervised learning machine technique based on combination of multi classifiers in data mining domain. Our research methodology consists of 6 main phases as follows:

1.9.1 Research and survey

Include reviewing the recent researches of worms detection that is closely related to the thesis problem statement. Then analyzing the existing methods, and identifying the drawbacks and disadvantages of each method in order to be overcome in our research.

1.9.2 Data set collection and preprocessing

In this step we will collect the data set from [37], where these datasets have been collected from the network end points such as homes, offices and universities. These data sets were collected from 13 different network endpoints. The datasets have been collected over a period of 12-months. Each network end-point has different behavior from each other. We will explain these datasets in details in chapter 4.

1.9.3 Build WDMAC model (proposed model)

The aim of this research to build a new model to solve worm's detection problem using multi classifiers as integration and combination strategy, this model named "WDMAC" model. Chapter 4 depicts in details the "WDMAC" proposed model.

1.9.4 Apply WDMAC model

Using Rapid Miner program, we will apply "WDMAC" model. The structure of our model, training, testing, and extracting the results will be explained in chapter (5).

1.9.5 Design experimental scenarios

To verify the developed model we will use various suitable real problems and artificial worms' datasets with corresponding datasets that are commonly used in worm's detection research.

1.9.6 Evaluate the obtained results

In this stage we will analyze the obtained results and justify the feasibility of our model by comparing it with other approaches.

1.10 Outline of the Thesis

This dissertation has been divided into six major chapters, which are structured around the objectives of the research. The dissertation is organized as follows:

Chapter 2, Presents Literature Review of worms and worms' detection approaches. Also, this chapter presents details about machine learning and data mining techniques, classification methods, and classification algorithms used on "WDMAC" model.

Chapter 3, Presents some related work of worms detection, and highlights its main shortages which are to be avoided and solved in our work.

Chapter 4, Includes the methodology steps and the architecture of the "WDMAC" model. An explanation about the data sets used in the experiments, preprocessing of these data set, and the experiment cases is included as well. Also, this chapter presents the baseline experiments to choose the optimal classifiers algorithms.

Chapter 5, Give the details about the sets of experiments, and analyze the experimental results. Also discussion for each set experiments. Produced some experiments to comparison goals.

Chapter 6, Will draw the conclusion and summarize the research achievement of experiments and suggests future work.

CHAPTER 2: Literature Review

In this chapter, we will identify worms, types of worms, characteristics of worms, worms' life phases, and how do worms spread across networks. Then we will describe and compare various approaches of worms' detection. Finally, we will explain the use of machine learning and data mining, especially classification techniques, and clarify their effectiveness in the detection of worms.

2.1 Brief Details about Computer Worms

The first worm attack on the internet was on 1988 with a worm named Morris worm. This worm attack changed people's way of thinking towards internet [30]. Thus, computer worms have become real and serious threat to computer networks, and can cause billions of dollars of damages within a few hours, and cause many of problems such as loss and theft of information, discontinuity access, unauthorized change of data, damages of systems, and networks or services, and denial-of-service attacks. In the next sub sections we will recognize worms closely.

2.1.1 Worms Definition

There exist many different definitions of a computer worm. A computer worm is defined as a malicious code that is self-replicating, self-contained, does not need to be part of another program to propagate itself, and does not need a host, it can spread on its own [8][19][32].

2.1.2 Types of Worms

- **Direct worms,** don't need a medium to spread, because they use computer networks, exploiting operating systems bugs or vulnerable [9].
- **Indirect worms,** spread in an "indirect" way, using deceitful means like peer to peer file sharing or, e-mails, and instant messaging [9].

2.1.3 The way of worm spread

Each worm has a different way of spreading, for instance:

- **Internet worms:** spread by copying themselves to network resources, exploiting operating system vulnerabilities, and penetrating public networks [29].

- **Email worms:** spread by infected e-mail messages carried by an attachment, or the e-mail may contain links to an infected web site. When the user opens the attachment, or clicks the link, the host gets infected immediately, and exploits the vulnerable e-mail software in the host machine to send infected e-mails to addresses stored in address book. Thus, new machines get infected [39] [20].
- **Instant Messaging worms:** spread using instant messaging applications by sending links to infected websites to everyone on the local contact list [26].
- **Peer–to–peer worms:** where worms copy themselves into a shared folder by placing a copy of itself under an acceptable name.

2.1.4 Characteristics of worms

There are two main common characteristics of most worms:

- **First,** most of the worms generate large amounts of similar or identical traffic, in this case we can detect worms using their signatures (misuse detection), but this method is not effective in the case of not knowing the signatures of worms [18].
- **Second,** most worms use random scanning in order to detect new target (victim). The worm picks a random IP address, scans it to see if it is vulnerable host, and reach to inactive IP addresses, then attempts an infection [23]. In this case, by observing rapid increases abnormal inactive IP addresses, can detect the appearance of worms [18]. Random scanning has some very good properties: it results in the worm scattering itself quickly through the network and the scans themselves seem to come from everywhere [23].

2.1.5 The worms' life phases

- **Target finding:** is the first step of a worms' life cycle where the worm discovers victims (vulnerable hosts).
- **Transferring:** refers to sending a copy of the worm to the target after the victim (target) is discovered.
- **Activation:** occurs when a worm starts performing its malicious activities. Activation might be triggered on a specific date or under certain conditions.
- **Infection:** is the result of the worm performing its malicious activities on the host.

During the phase of target finding and worm transferring, the worm is active over the Internet, making it possible for network-based intrusion detection systems (NIDSs) to catch the worm. The activities in the two latter phases are limited to local machines and are harder to be detected by NIDSs. The first two phases cause network activities, worm behaviors in these two phases are critical for developing detection algorithms [25][18]. Categorizes characteristics of worms in the target finding and worm transfer phases into four categories based on the worm's target finding scheme, propagation scheme, transmission scheme, and payload format, shows in Figure 2.1.

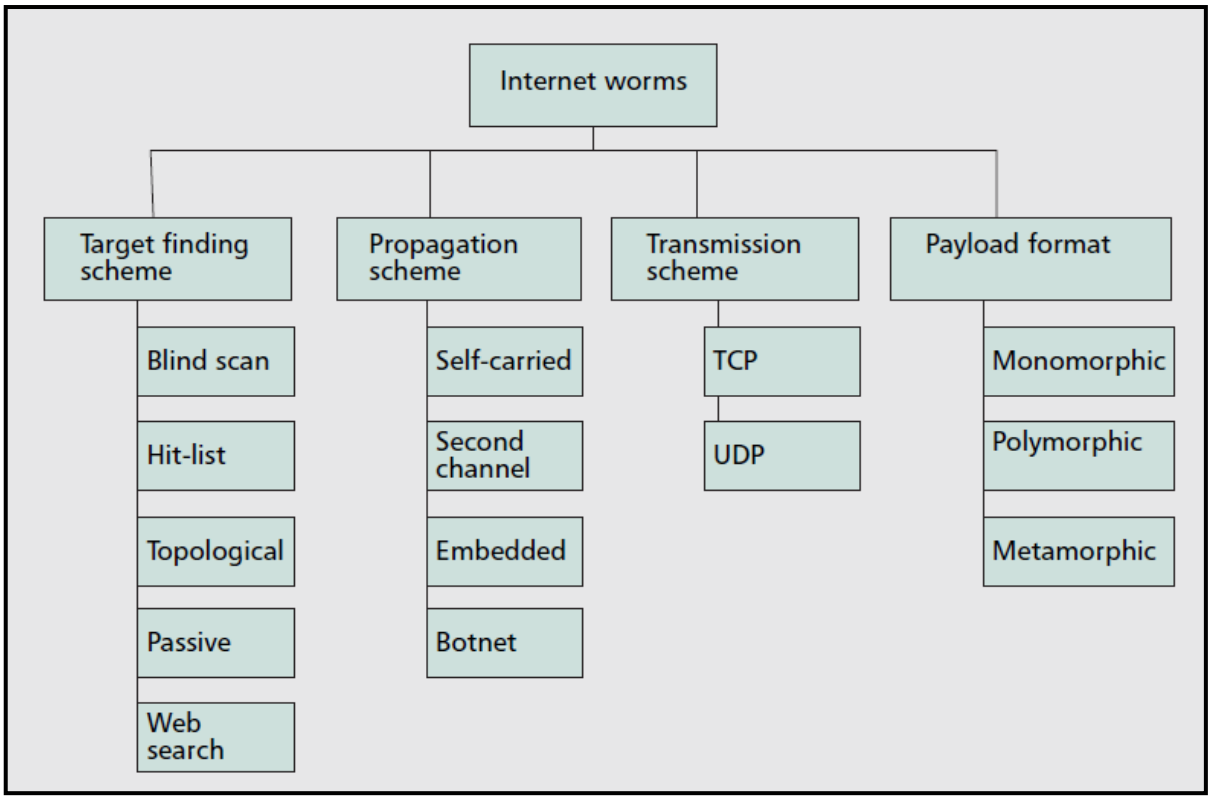

Figure 2.1: Categorization of worm characteristics [18]

2.2 Intrusion Detection System (IDS), and Worm Detection

The Purpose of Intrusion Detection Systems (IDS) is to monitor network assets in order to detect misuse or abnormal behavior, that is statistically analyzing input data (e.g., network traffic) for the purpose of detecting whether an intrusion has occurred or is occurring [21]. The types of IDS can be divided into two categories: network based (NIDS) and host based (HIDS). Network based (NIDS) tries to detect any attempt to subvert the

normal behavior of the system by analyzing the network traffic. Host based (HIDS) to act as the last line of defense, which seeks to detect intrusions by analyzing the events on the local system while the IDS is running. Generally host based IDSs classified into two categories: anomaly detection and misuse detection. Misuse detection try to identify behavior patterns that are characteristic of intrusions, but this can be difficult if an attack exhibits novel behavior, as it may when attackers develop new strategies. Anomaly detectors try to characterize the normal behavior of a system so that any deviation from that behavior can be labeled as a possible intrusion. Anomaly detection assumes that misuse or intrusions are strongly correlated to abnormal behavior exhibited by either the user or by the system itself. Anomaly detection approaches must first determine the normal behavior of the object being monitored, and then use deviations from this baseline to detect possible intrusions [22]. In worms detection the details HIDS as the following:

2.2.1 Misuse Detection

The traditional way of worm detection based on signature, known as signature based detection. A signature is a unique pattern in the worm body that can identify it as a particular type of worm [20]. So, a worm can be detected from its signature, this method is adopted in most commercial software and is based on the using of signatures of worms that are available after the attacking the network (victim) [29]. But the problem with this approach is that it involves significant amount of human intervention and may take a long time (from days to weeks) to discover the signature [20]. Extraction of signatures must be done by experts [25] [29]. So, this approach fails and not useful against "zero-day" attacks of computer worm. Besides, signature matching is not effective against polymorphism [20].

2.2.2 Behavioral, Anomaly Detection

This approach has become a great challenge, and requires benign and infected behavior [29]. It is based on the detection of abnormal behaviors and generates alarms. This technique requires the definition of normal network behaviors, to be able to predict the behavior of non-normal network behaviors; it depends on the period of training before the system can be effective in protecting the network. Even though this approach is the best to detect unknown worms, it is challenged by the problem of defining what normal network behavior is, deciding the threshold to generates the alarms, and prevention of false alarms [18].

2.3 Machine Learning

Is a branch of artificial intelligence domain, which is concerned about the development of methods for machine learning from experience or extract knowledge from given examples in a database. It is used to improve and develop IDS. There are a variety of machine learning ways, but, most of them fall under the following two classes [31]:

2.3.1 Supervised Learning

Produces a function that maps inputs to a desired outputs. The function is trained by training examples which consist of pairs of input objects, and desired outputs. One standard formulation of the supervised learning task is the classification problem [31].

2.3.2 Unsupervised Learning

It is a set of inputs where labeled examples are not given. It can be thought of as finding patterns in unlabeled input data. Clustering problems is a simple example of this technique. The application of unsupervised learning technique in the intrusion detection area can be used to detect new kinds of attacks, provided, the attacks exhibit some unusual character in some feature space [31].

2.4 Data Mining

It is considered as one of the applications of supervised machine learning, and it plays an important role in the process of retrieving the lost information. Data mining refer to the analysis of large quantities of data that are stored in computers [4], and is defined as knowledge discovery, which is the process of extracting useful patterns from large volumes of data using special algorithms [3][35]. Many terms carry a similar or slightly different meaning to data mining, such as knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging [7]. Data Mining is essentially a process of data drive extraction of not so obvious but useful information from large databases that is interactive and iterative. Knowledge discovery as a process consists of an iterative sequence of the following steps:

1) **Data Cleaning:** is removing the noise and inconsistent data.

2) **Data Integration:** where multiple data sources may be combined. These sources may include multiple databases, data cubes, or flat files.

3) **Data Selection:** where data relevant to the analysis task are retrieved from the database. So, irrelevant, weakly relevant or redundant attributes may be detected and removed.

4) **Data Transformation:** where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance.

5) **Data Mining:** an essential process where intelligent methods are applied on data to extract data patterns for decision making.

6) **Pattern Evaluation:** to identify the truly interesting patterns based on some interestingness measures. A pattern consider interesting if it is: Valid, Novel, Actionable, Understandable

7) **Knowledge Presentation:** is the framework that converts a large amount of data into a particular data or procedure that human being can figure out based on an intention. In Knowledge representation visualization tools and knowledge representation techniques are used to present the mined knowledge to the user.

Figure 2.2, illustrates data mining as a step in the process of knowledge discovery.

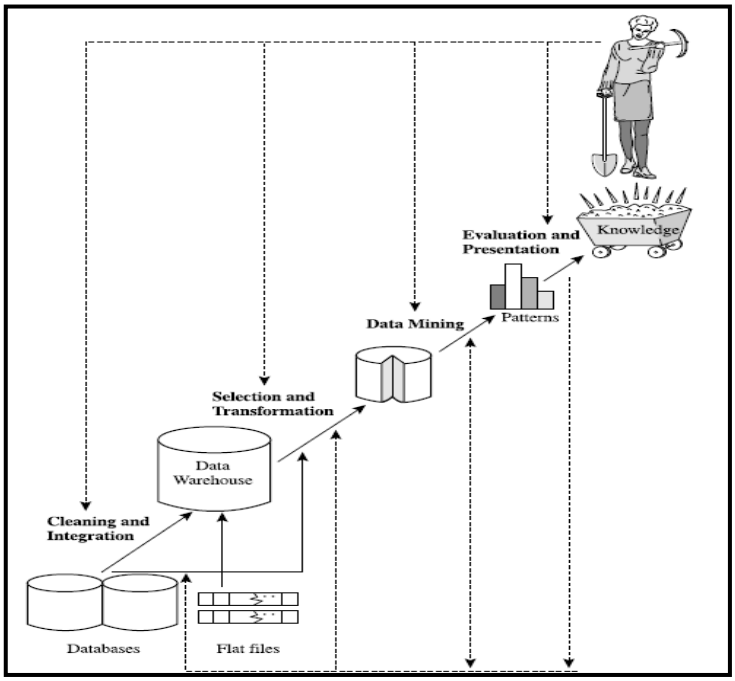

Figure 2.2: Data mining as a step in the process of knowledge discovery [7].

Data Mining functionalities are used to specify the type of patterns to be found in the data mining tasks. In general data mining tasks can be classified into two main categories: descriptive and predictive. Descriptive mining tasks characterize the general properties of the data. Predictive mining tasks perform inferences on the current data in order to make predictions [7]. Most of data mining tasks can be one or combination of the following:

1) Classification: used for predictive mining tasks. This methods is intended for learning different functions that map each item of the selected data into one of a predefined set of classes. Given the set of predefined classes, a number of attributes, and a "learning (or training) set," the classification methods can automatically predict the class of other unclassified data of the learning set[7].

2) Prediction: used for predictive mining tasks**.** Analysis is related to regression techniques. The key idea of prediction analysis is to discover the relationship between the dependent and independent variables. For example, by using historical data from both sales and profit, either linear or nonlinear regression techniques can produce a fitted regression curve that can be used for profit prediction in the future [4].

3) Association Rules: used for descriptive mining tasks. It aims to find out the relationship among valuables in database, and produce a set of rules describing the set of features that are strongly related to each other's, so that the relationship of a particular item in a data transaction on other items in the same transaction is used to predict patterns [7].

4) **Clustering:** used for descriptive mining tasks. It is unsupervised, and does not require a learning set. It shares a common methodological ground with Classification. It ungrouped data and uses automatic techniques to put this data into groups [4]. In other words, finds groups of data pointes (clusters) so that data points that belong to one cluster are more similar to each other than to data points belonging to different cluster.

5) **Outlier Analysis:** used for predictive mining tasks**.** Discovers data points that are significantly different than the rest of the data. Such points are known as exceptions or surprises. While outliers can be considered noise and discarded in some applications, they can reveal important knowledge in other domains, and thus can be very significant and their analysis valuable. So that very important identify the outliers [7].

2.4.1 Classification

It is one of the data mining techniques that falls under supervised machine learning techniques classification. The classifier needs to be trained with labeled input examples, so that it could understand the characteristics of different classes, and then, it could be able to map new data items to different classes [31][3]. There are many classification algorithms in data mining. We will describes some of those algorithms in order to be used in our research such as Naïve Bayes (NB), Decision Tree (DT), and Artificial Neural Network (ANN). Following is a brief overview about the classification algorithms mentioned above. Two key research problems related to classification results are the evaluation of misclassification and prediction power [4].

2.4.1.1 Naïve Bayes (NB)

Naïve Bayes is a technique for estimating probabilities of individual variable values, given a class, from training data and to then allow the use of these probabilities to classify new entities, which is a term in Bayesian statistics dealing with a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. In simple terms, a naïve Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. The naïve Bayesian classifier, works as following derivation [7]:

1) Let ***D*** be a training set of tuples and their associated class labels. Each tuple is represented by an ***n***-dimensional attribute vector, $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$, ***n*** measurements made on the tuple from ***n*** attributes, respectively, $A_1, A_2, \ldots, A_n$.
2) Suppose that there are ***m*** classes $C_1, C_2, \ldots, C_m$. Given a tuple, **X**, the classifier will predict that **X** belongs to the class having the highest posterior probability, conditioned on **X**. That is, the naïve Bayesian classifier predicts that tuple **X** belongs to the class $\mathbf{C_i}$ if and only if

$$P(C_i|X) > P(C_j|X) \quad for\ 1 \leq j \leq m, j \neq i$$

The maximize $\mathbf{P(C_i|X)}$. The class $\mathbf{C_i}$ for which $\mathbf{P(C_j|X)}$ is maximized is called the maximum posteriori hypothesis. By Bayes' theorem (Equation (2.1)),

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{p(X)} \qquad (2.1)$$

3) Since P(X) is constant for all classes, only (Equation (2.2)) need to be maximized.

$$P(C_i|X) = P(X|C_i)P(C_i) \quad (2.2)$$

4) Based on the assumption is that attributes are conditionally independent (i.e., no dependence relation between attributes),the computing of P(X|Ci) using (Equation (2.3)).

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \quad (2.3)$$

Reduces the computation cost by Equation 2.2, only counts the class distribution. If A_k is categorical, $P(x_k|C_i)$ is the no. of tuples in C_i having value x_k for A_k divided by $|C_i, D|$ no. of tuples of Ci in D). And if A_k is continuous-valued, $P(xk|Ci)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ and $P(xk|Ci)$ is:

$$P(X|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i} \quad (2.4)$$

$$g(x_k, \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.5)$$

Where μ is the mean and σ is the variance. If an attribute value doesn't occur with every class value, the probability will be zero, and a posteriori probability will also be zero.

2.4.1.2 Decision Tree (DT)

Decision Tree is a common method used in statistics, data mining and machine learning, where It is an efficient method for producing classifiers from data. It is considered as a tree-structured plan of a set of attributes to be tested in order to predict the output. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Moreover, it is a type of tree-diagram used in determining the optimum course of action, in situations having several possible alternatives with uncertain outcomes. A decision tree classifier is modeled in two phases: tree building and tree pruning. In tree building, the decision tree model is built by recursively splitting the training data set and assigning a class label to leaf by the most frequent class. Pruning a sub tree with a leaf or a branch if lower training error obtained. Table (2.1) presents decision tree algorithm [7].

Table 2.1: Basic structure of Decision Tree algorithm [7].

Input:

- Data partition, **D**, which is a set of training tuples and their associated class labels;
- *attribute list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of *a splitting attribute* and, possibly, either *a split point* or *splitting subset*.

Output: A decision tree.

Method:

(1) create a node ***N***;
(2) **if** tuples in ***D*** are all of the same class, ***C*** **then**
(3) return ***N*** as a leaf node labeled with the class ***C***;
(4) **if** *attribute_list* is empty **then**
(5) return ***N*** as a leaf node labeled with the majority class in ***D***; // majority voting
(6) apply **Attribute selection method**(**D**, *attribute-list*) to **find** the "best" *splitting criterion*;
(7) label node ***N*** with *splitting_criterion*;
(8) **if** *splitting_attribute* is discrete-valued **and** multiway splits allowed **then** // not restricted to binary trees
(9) *attribute_list* ← *attribute_ list – splitting_attribute*; // remove splitting attribute
(10) **for each** outcome **j** of splitting criterion
// partition the tuples and grow subtrees for each partition
(11) let $\mathbf{D}_j$ be the set of data tuples in **D** satisfying outcome **j**; // a partition
(12) **if** $\mathbf{D}_j$ is empty **then**
(13) attach a leaf labeled with the majority class in **D** to node **N**;
(14) **else** attach the node returned by **Generate decision tree**($\mathbf{D}_j$, attribute list) to node **N**;
end for
(15) return **N**;

2.4.1.3 Artificial Neural Networks (ANN)

Artificial Neural Networks are one of the classification methods defined by non-linear predictive models that learn through training and resemble biological neural networks in structure which can be used to model complex relationships between inputs and outputs, in order to find patterns in the data. It consists of an interconnected group of artificial neurons (which are actually mathematical functions) used to process information. An important characteristic of artificial neural network is that, during the learning phase, it can change its structure based on external or internal information that flows through the network [31] [33].

There are many different kinds of neural networks and neural network algorithms. The most popular neural network algorithm is back propagation (BP), which performs learning on a multilayer feed-forward neural network. It iteratively learns a set of weights for prediction of the class label of tuples. A multilayer feed-forward neural network consists of an input layer, one or more hidden layer, and an output layer. The multilayer neural network shown in Figure 2.3.

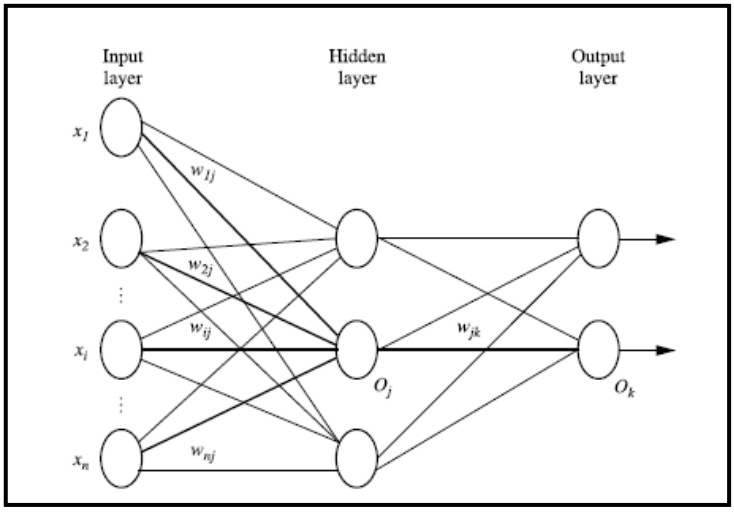

Figure 2.3: A Multilayer Feed-Forward Neural Network [7].

Back propagation learns by iteratively processing a data set of training tuples, comparing the network's prediction for each tuple with the actual known target value. The target value may be the known class label of the training tuple (for classification problems) or a continuous value (for prediction). For each training tuple, the weights are modified so as to minimize the mean squared error between the network's prediction and the actual target value. These modifications are made in the "backwards" direction, that is, from the output layer, through each hidden layer down to the first hidden layer. Although it is not guaranteed, in general the weights will eventually converge, and the learning process stops. The steps involved are expressed in terms of inputs, outputs, and errors, and may seem critical if this is your first look at neural network learning. However, once you become familiar with the process, you will see that each step is inherently simple. Table (2.2) presents the description steps of Neural Network algorithm [7].

Table 2.2: Back propagation NN algorithm [7].

Input:

- ***D***, a data set consisting of the training tuples and their associated target values;
- ***l***, the learning rate;
- ***network***, a multilayer feed-forward network.

Output: A trained neural network.

Method:

(1) Initialize all weights and biases in *network*;
(2) **while** terminating condition is not satisfied {
(3) **for** each training tuple ***X*** in ***D {***
(4) // Propagate the inputs forward:
(5) **for** each input layer unit ***j*** *{*
(6) $O_j = I_j$; // output of an input unit is its actual input value
(7) **for** each hidden or output layer unit ***j {***
(8) $I_j = \sum_i w_{ij} O_i + \Theta_j$; //compute the net input of unit *j* with respect to the previous layer, *i*
(9) $O_j = 1/\ 1+e^{-Ij}$; } // compute the output of each unit *j*
(10) // Backpropagate the errors:
(11) **for** each unit *j* in the output layer
(12) $Err_j = O_j(1-O_j)(T_j-O_j)$; // compute the error
(13) **for** each unit *j* in the hidden layers, from the last to the first hidden layer
(14) $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$; // compute the error with respect to the next higher layer, *k*
(15) **for** each weight w_{ij} in *network {*
(16) $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
(17) $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
(18) **for** each bias Θ_j in *network {*
(19) $\Delta\Theta_j = (l) Err_j$; // bias increment
(20) $\Theta_j = \Theta_j + \Delta\Theta_j$; } // bias update
(21) }}

2.5 Multi Classifier System (MCS)

In this time, recent researches showed that standalone classifiers used by worms detection is unable to provide acceptable accuracies in real-world deployments. To achieve higher accuracies, in worms detection systems use multi classifiers whose outputs are combined to formulate an aggregate detect score.

A Multiple Classifier System (MCS) is a pattern classification system consists of an group of individual classifiers whose outputs on an input sample are combined in some way to get a final decision on its classification.

The goal of classifier combination that can allow to overcome some known limitations of the traditional approach to classifier design means that using a monolithic classifier chosen as the best one for the application at hand, among a given set of available classification algorithms.

It is often very difficult to find the real best classifier for the task at hand, while different classifiers designed for the same task can exhibit complementary strengths and weaknesses, where a proper combination of an ensemble of different classifiers could therefore be more effective than using a single, monolithic classifier.

Relevant contributions to MCS have been provided by the machine learning, neural networks, and statistics fields. Both theoretical and empirical evidence acquired in the past years led MCS to become to date one of the main tools for the design of classification systems. Despite this, MCS still exhibit several open issues, and therefore are still one of the main research topics in the anomaly detection field.

In "WDMAC" model used three classifiers algorithms which are (Naïve Bayes, Decision Tree, and Artificial Neural Network), where combined the tree outputs to generate the final output for all models, as the final output relies on equality the output of two model (for instance, if any two classifier equal "worm", and the third was "normal", so that the general output for "WDMAC" was "worm", or if any two classifier equal "normal", and the third was "worm", so that the general output for "WDMAC" was "normal").

Summary

In this chapter, we presented the details of computer worm, and approaches used in worms detection system. Data mining techniques and its use in worms detection have been explained as well. Furthermore, a brief description has been proposed about classifiers algorithms (NB, DT, and ANN) to be used in applying "WDMAC" model. Finally we explained the importance of multi classifiers system.

CHAPTER 3: Related Work

Many recent researches in the last few years have been proposed and presented about **"Worms Detection"** domain based on data mining as an efficient way to improve the security of networks. Classification techniques was the most widely used for many resent researches, **ANN** used in [25][16],and [10] with other classifiers such as **DT**, **NB**, and **BN**. In [8][28], **BN**, **DT**, and **RF** classifiers algorithms were used. In [14][15], **SVM** classifiers were used. In [13] **NB**, **J48**, **SMO** and **Winnow** classifiers were used. In [19], **DT**, and **K-NN** classifiers were used. In [17], 9 prominent classifiers were used. In [11], **DT**, **RF**, and **Bagging** were used. In the following, we discuss these approaches as follows:

Farag et-al [25] produced a model for detecting unknown worms based on data collected from the local victim information by using a developed application called (Worm Detection Traffic Analyzer). Their model is used to identify worm traffic from normal traffic; also it can predict the infection percentage in the network. The proposed system uses Artificial Neural Network (ANN) for classifying worm/ non worm traffic with accuracy of %99.96, and predicting the percentage of infection in the infected network with absolute error average from 0% to 4%, which can be used by the administrator to take the right action.

The problem of their system is that when the system becomes highly infected, this leads to slightly increase the prediction error rate. So, they need to modify the learning of ANN module to include more training set in for the conditions of high infection, or use multi classifiers algorithms.

Sarnsuwan et-al [8, 40], presented approaches to detect internet worm by using Bayesian network, C4.5 Decision tree and Random Forest classification approaches of data mining techniques. In [8], **Sarnsuwan et-al** tries to detect and classify many types of worms at network end point. But in [28], **Sarnsuwan et-al** considered behaviors of internet worm that is different from the normal pattern of internet activities and all network packets before they reach to the end-user by extracting 13 features of internet worm from these packets. In general they achieved good results as shown as follows:

Sarnsuwan et-al [8], produced techniques to detect and classify many types of internet worm at network end-point by using data mining approaches which are Bayesian network,

C4.5 Decision tree and Random Forest. They use port and protocol profiles to train and test their detection models. Their results show that the detection rates of classification and detection known worms are at least 98.5% while the unknown worm detection rate is about 97% with Decision tree and Random forest, and 80% with Bayesian network.

The researcher explained the drawback of t their model appears in the detection rate unknown worm was 97 % not sufficient compared with the great danger inflicted by worms on the network, and some worms have port profiles similar to those in the normal data profiles that may cause difficulty for worm detection.

Sarnsuwan et-al [28], presented a new approach to detect internet worm. They considered behaviors of internet worm that is different from the normal pattern of internet activities, and all network packets before they reach to the end-user by extracting 13 features of internet worm from these packets. Where used three efficient data mining algorithms which are Bayesian Network, Decision tree and Random Forest are considered to classify behaviors of Normal network data, Blaster Worm, UDP flood, Http flood and Port Scan. Their approach has achieved good results with detection rate over 99.6 percent and false alarm rate is close to zero with Random forest algorithm. Also the model can detect internet worm and classify DoS and Port Scan attacks with detection rate over 99% and false-alarm rate close to zero.

The researcher explained their experiences have achieved good results, but the details about the data sets and their volume is unclear (training, and testing data sets), since it plays an important role to evaluate the detection rate and false alarm rate.

Aiello et-al [9], proposed a new technique to detect internet worm based on the fact that an indirect worm (a worm spreading by e-mail and not using system bugs) needs to spread quickly and sends a lot of e-mail in a short while, producing an anomalous behavior. **Aiello et-al** found stealthy worms through detecting traffic anomalies, and focus on one mail-server log of a real network and the results obtained drove them to detect indirect worm with different approaches based on various parameters which are global email flow, single host e-mail flow, reject, and sender field analysis. The six approaches do detect various kind of worm (stealthy worms, lazy worms, hasty worms). **Aiello et-al** see there isn't an approach which could detect all the worms, so they think that it might be a good idea to use all the approaches in a threshold system. Also expect another possibility could be to skill

the system with an expert trainer using artificial neural network to achieve the best results in detection rate, low occurrences of false positives, and identify more features connected to worms activities.

Moskovitch et-al [10], present presented the concept of detecting unknown computer worms based on a host behavior, using Data Mining algorithms for detecting the presence of an unknown worm not necessarily by recognizing specific instances of the worm, but rather based on the computer measurements. During the experiments 323 computer features were monitored. Four feature selection techniques were used to reduce the amount of features and four classification algorithms which are Decision Trees, Naïve Bayes, Bayesian Networks and Artificial Neural Networks that applied on the resulting feature subsets. Their results indicate that using this approach resulted in an above 90% average accuracy, and for specific unknown worms accuracy reached above 99%, using just 20 features while maintaining a low level of false positive rate.

The researcher explained the advantage of the proposed approach is the automatic acquisition and maintenance of knowledge, based on inductive learning. This avoids the need for a human expert who is not always available. This is possible these days, based on the existing amount of known worms, as well as the generalization capabilities of classification algorithms.

Masud et-al [20], presented work data mining techniques to detect e-mail worms, where e-mail message contained many of different features such as the total number of words in message body, presence of binary attachments, type of attachments, and so on, that played important role to obtain an efficient classification model based on these features. Masud et-al [20], divided their work in three phases: the first phase, was reduced the number of features by using two different approaches which are feature-selection and dimension-reduction, the goal of this step to reduced noise and redundancy from the data, and select the best set of features that can efficiently distinguish between normal and abnormal emails. The second phase, applied two classification techniques which are Support Vector Machine (SVM), Naïve Bayes, and their combination. The last phase, the trained classifiers are tested on a dataset containing of known and unknown types of worms. The proposed feature-selection along with SVM classification achieves the best accuracy in detecting both known and unknown types of worms.

Ellis et-al [12], presented a new approach for automatic detection of worms using behavioral signatures that describes aspects of the behavior of any particular worm that are common across the manifestations of certain worm, which extends the contract in the temporal order, also presented within the context of a general worm propagation model. They presented the concept of a network application architecture (NAA, which defines how an enterprise will distribute the functionality of its network applications across its network) as a way to distribute network applications that impacts the sensitivity of behavioral signatures, and satisfies certain constraints significantly improves worm detection sensitivity.

Ismail et-al [13], evaluated the detection of worms based on content classification by using all machine learning techniques available in WEKA data mining tools. Four most accurate and quite fast classifiers are identified for further analysis–Naive Bayes, J48, SMO and Winnow can detect worms with accuracy between 94% and 99%. J48 produces the best performance than the other classifiers. They analyzed the accuracy these four classifiers under the presence of class noise in learning corpora. By injecting class noise ranging between 0% and 50% into positive and negative corpora, results from the simulation show gradual decrease in accuracy and increase in false positive and false negative for all analyzed techniques. The presence of the classes' noise affects false positive more significantly compared to false negative. The results show that worm detection with classification algorithms could not tolerate the presence of classes' noise in learning corpora.

The researcher explained the drawback of t their model appears in the accuracy rate with 94–99 % is not sufficient compared with the great danger inflicted by worms on the network and its resources.

Sharma et-al [14], presented reports on the efficiency of using a machine learning technique to detect variants of known worms in real-time. Support vector machines (SVMs) are a machine learning technique known to perform well at various pattern recognition tasks, this work applies SVMs to the worm detection problem, and classify various types of synthetically generated worms by the optimal configuration of SVMs and associated kernel functions greatly affects the classifiers performance in real time detection of variants of known worms. In addition, demonstrated that SVMs are robust to mutations in signature

model and, to a certain degree, signature corruption. for the difficulty in obtaining large numbers of real worm variants. **Sharma et-al,** were investigating a way of automatically generating variants from a single worm. This variant generator consists of three machines connected in series: an attacker, a forwarding proxy, and a target. The basic idea was to launch a worm from the first machine, reassemble its flow and randomly mutate it in the second machine, and then forward it to the third. In a final step the verify whether the target has been infected and label the generated mutation as malicious or benign accordingly.

The researcher appear the drawback of this study that lack of a real worm in order to be testing them and proving the effectiveness of the proposed approach.

Wang et-al [15], proposed a new worm detection approach based on mining dynamic program executions that captures dynamic program behavior to provide accurate and efficient detection against both seen and unseen worms. The execution on a large number of real-world worms and benign programs, and trace their system calls. They applied two classifier-learning algorithms which are Naive Bayes (NB) and Support Vector Machine (SVM)) to obtain classifiers from a large number of features extracted from the system call traces. The learned classifiers are further used to carry out rapid worm detection with low overhead on the end-host. The experimental results clearly demonstrate the effectiveness of proposed approach to detect new worms in terms of high detection rate with 99.5% in SVM and 96.4% in NB, and a low false positive rate with 2.22% in SVM and 6.67% NB.

The researcher explained that false positive rate 2.22 and 6.67 considered high in worm detection, where we aim to reach false positive close to zero.

Stopel et-al [16, 19], proposed the model to detect malicious activity of worms by looking at the attributes derived from the computer operation parameters such as memory usage, CPU usage, traffic activity etc, using Artificial Neural Networks ANN, and two other known classifications techniques, Decision Tree and k-Nearest Neighbors.

Stopel et-al [16], proposed an approach for detecting the presence of computer worms based on Artificial Neural Networks (ANN) using the computer's behavioral measures. The identification of the most prominent features to capture efficiently the computer behavior in the context of worm activity and compared three different feature selection techniques for the dimensionality reduction In order to evaluate the different techniques, several computers were infected with five different worms and 323 different features of the

infected computers were measured. The evaluation each technique by preprocessing the dataset according to each one and training the ANN model with the preprocessed data. Then evaluated the ability of the model to detect the presence of a new computer worm, in particular, during heavy user activity on the infected computers. The best accuracy was achieved by using only five attributes selected by the Fisher's score method. The average accuracy of new worm detection using these attributes is 0.90. These five attributes were related to memory management, and number of system context switches.

Stopel et-al [19], present a new approach based on ANN for detecting the presence of computer worms based on the computer's behavioral measures, and used two other known classifications techniques, Decision Tree and k-Nearest Neighbors, to test their ability to classify correctly the presence, and the type of the computer worms even during heavy user activity on the infected computers. By comparing these three approaches, the ANN approach has computational advantages when real-time computation is needed, and has the potential to detect previously unknown worms. Addition, ANN may be used to identify the most relevant, measurable features, and thus reduce the feature dimensionality.

The researcher explained the drawback of t their model appears in the accuracy rate with 90% is not sufficient compared with the great danger inflicted by worms on the network and its resources, and misclassifications that it still faces difficulties related to the detection of the worms in the beginning of their activity. Measure the evaluation of the system using additional types of malwares such as Viruses, Trojans and so on.

Ashfaq et-al [17], presented analysis of a combination of N parallel connected anomaly classifiers. They adapted and evaluated existing combining methods for the traffic anomaly detection problem and showed that the accuracies of these detectors can be improved. They proposed a Standard Deviation normalized Entropy of Accuracy (SDnEA) combining method which provided consistent and considerable accuracy (detection rates and false alarm) improvements over existing combiners. They also showed that increasing the number of classifiers does not induce a proportional increase in system accuracy. Therefore, a few judiciously selected classifiers can provide better system-level accuracy than many diverse classifiers. By using 9 prominent classifiers operating on two publicly-available traffic datasets, they show that around 3%−10% increase in detection rate and a

40% decrease in false alarm rate over existing combining techniques can be provided by the proposed information-theoretic NADS combining technique.

The researcher noted that the number of classifiers used in their study cost time and effort.

Rasheed et- al [27], proposed intelligent early system detection mechanism for detecting internet worm, which using connection behavior to detect internet worm. They considered the difference of normal connections and worm connections. The worm connections were expected to have high number of failure connections. In addition, the failure connections can be occurred when a source IP sends a request connection packet to an unused IP address or some ports that no longer in service. After that, ICMP packet, SYN/ACK packet and TCP RESET will be returned. This leads to the number of these packets will be high.

Siddiqui et-al [11], presented a novel idea of extracting variable length instruction sequences that can identify worms from clean programs using data mining techniques which are decision tree, bagging and random forest. The feature used for the process was the frequency of occurrence of variable length instruction sequences. The effect of using such a feature set is twofold as the instruction sequences can be traced back to the original code for further analysis in addition to being used in the classifier. Siddiqui et-al [11], used the sequences common to both worms and clean programs to remove any biases caused by the features that have all their occurrences in one class only. This approach showed 95.6% detection rate on novel worms, 3.8% false positive rate, and 96.2% to overall accuracy.

The researcher appeared that 3.8% false positive rate considered problem in worms detection, also detection rate and accuracy still unacceptable results with the risks of worms.

Summary

In this chapter we presented an overview about some of researches conducted in worms detection problem, where presented the worms detection based on data mining as an efficient way to improve the security of networks as classification techniques was the most widely used for many resent researches. We explained the drawbacks of the existing methods used in previous researches which were many problems related to accuracy, detection rate, classification error.

CHAPTER 4: Research Proposal and Methodology

In this chapter, we present and explain the proposed model (WDMAC) and methodology which we followed in this research. This chapter organized into four sections. Section 4.1, presents methodology steps of "WDMAC" model, given description of the collecting data sets and description of their attributes, worms which are used, and spilt the data set to groups that case1, 2, and 3. Section 4.2, contains data preprocessing and feature selection. Section 4.3, contains the process of building the "WDMAC" model including the baseline experiments to select the optimal classifiers algorithms in order to be used to build the "WDMAC" model. An explanation about the parameters for each algorithm has been mentioned as well. Section 4.5, present the measures to evaluating the performance of classification with explained the equations used. Section 4.6, explained "WDMAC" model.

4.1 Methodology Steps

To apply and evaluate "WDMAC" model, we use the following methodology steps as presented in Figure 4.1:

1) **Collection data sets:** the collection of data sets from "Wireless and Secure Networks (WiSNet) Research Lab at (SEECS)".
2) **Preprocessing data sets and Feature Selection:** for the purpose of applying "WDMAC" model, Data sets cleaning, features selection, and replacement of data sets attributes should be done.
3) **Applying the model:** by using three classification algorithms: Naïve Bayes (NB), Decision Tree (DT), and Artificial Neural Network (ANN) as multi classification.
4) **Evaluate the model:** to evaluate the classification performance of our model, we used accuracy, misclassification rate, detection rate, and F-measure.
5) **Comparing phase:** we applied two comparison:
   a) Compare the performance by using for each classifier algorithm as independently, and by using the "WDMAC" model for each case of dataset.
   b) Compare performance between "WDMAC" model and other works which can be used for worm detection.

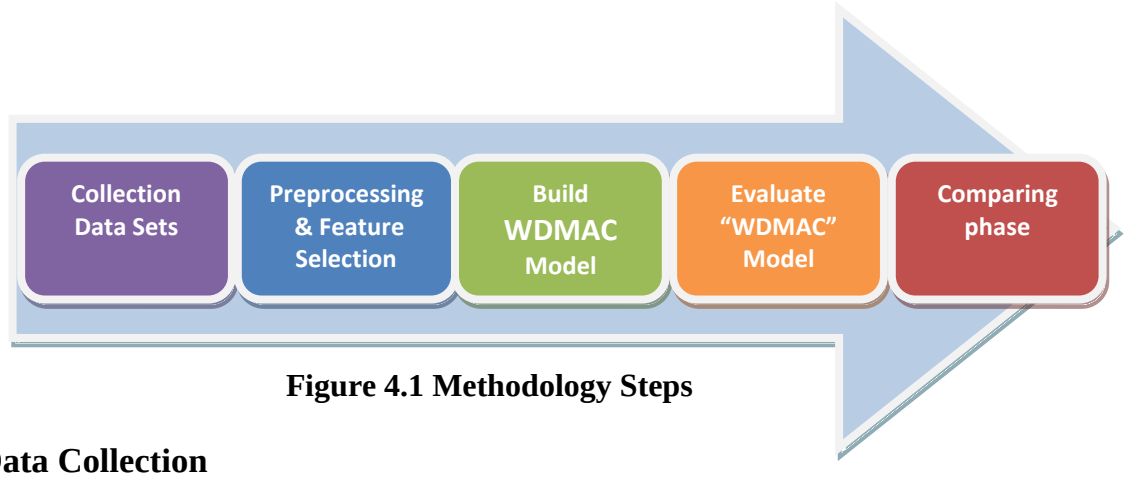

Figure 4.1 Methodology Steps

4.1.1 Data Collection

We use the datasets from [37], which have been collected from 13 different network endpoints. These datasets were collected over a period of 12-months, and from the network end points such as homes, offices and universities. Each network end-point has different behavior from each other, and some end points run peer to peer applications. Addition, There are many port numbers used in normal class data such as port numbers 22, 53, 80, 123, 135, 137, 138, 443, 445, 993 and 995 which are known ports (0:1023) and registered ports (1024:65535) for specific applications such as on-line Games and peer to peer applications. Each end host was installed with actual worm (i.e., Zotob.G, Forbot-FU, Blaster, Rbot.CCC) and simulated worm (i.e., CodeRedII). These datasets were collected by a multi-threaded windows application called "argus", which runs as a background process and stores network and keystroke activity in a log file. "argus" only logs session-level information where a session corresponds to bidirectional communication between two IP addresses. Communication between the same IP address on different ports is considered part of the same network session. This session-level granularity provides complete information about sessions originating from or terminating at an endpoint. Each session is logged using the information contained in the first packet of the session. A session expires if it does not send/receive a packet for more than τ seconds. In the collected data, τ is set to 10 minutes. For each logged session, "argus" also logs the last keystroke or mouse click that was pressed before the first packet of the session. The last keystroke is associated with a session only if the key was pressed no more than λ seconds before the session. If there was no key pressed in the last λ seconds before a session then a void keystroke value of zero is inserted. In the collected traces, λ is set to 10 seconds [36]. Assumed that the last pressed key has initiated the associated session. This session key logged as 'd9' for malicious sessions.

4.1.2 Log file Description

The log file is a text file, and each instance of dataset has 7 attributes as follows in Table 4.1:

Table 4.1: Description of Dataset attributes [36]

Attribute	Description
Session id	20-byte SHA-1 hash of the concatenated hostname and remote IP address.
Direction	One byte flag indicating (4) outgoing unicast, (3) incoming unicast, (2) outgoing broadcast, or (1) incoming broadcast packets.
Protocol	Transport-layer protocol (i.e., TCP or UDP) of the packet.
Source port	Source port of the packet.
Destination port	Destination port of the packet
Timestamp	Millisecond-resolution time of session initiation.
Virtual key code	One byte virtual key code that identifies the data if it is normal data or worm. Note that: the one byte virtual key code, as defined by Microsoft's MSDN library, of the last (keyboard or mouse) keystroke that was pressed before the session. In view of these stringent privacy considerations, we only logged the very last keystroke that was pressed right before the first packet of a new session.

4.1.3 Log file samples

Samples of datasets profiles is shown in Table 4.2. This table shoes that the log file consists of the following: the direction column which is a one byte flag represented by an integer where 1 represents "incoming broadcast packets", 2 represents "outgoing broadcast", 3 represents "outgoing unicast" and 4 represents "incoming unicast". The protocol column represents transport-layer protocols using an integer such as 6 represents "TCP" and 17 represents "UDP". The Key code column is one byte virtual key code that identifies the data types which are "d9" represents malicious (worm) behavior and others represent normal behavior.

Table 4.2: Samples of a data profile

Session ID	Direction	Protocol	Src Port	Des port	Time Stamp	Key code
Sha-1 code	3	6	2025	445	1130861747.125	d9
Sha-1 code	4	17	1026	53	1130863119.917	0
Sha-1 code	1	17	68	67	1130532152.766	1
Sha-1 code	4	6	1150	5061	1147212945.640	4c

4.1.4 Worms List

At the time during collected the dataset, these were the most prevalent malware threats. Even today, also we can find numerous studies being performed on these malware. We describe several information of various worms that are used in our experiments. Many characteristics of each worm including Port profiles, and rate of scan per second used by worm to infect new hosts [10][32], are shown in Table 4.3.

Table 4.3: Worms Characteristics

Worm	Port	Scan/second	Description
Blaster	TCP 135,444 UDP 69	10.5	Exploits a buffer overflow vulnerability of DCOM RPC on Windows XP and Windows 2000 by connecting to ports 135 and 4444 on TCP protocol and port 69 on UDP protocol. It can download and operate itself. After that, the sending DOS attacks to prevent patch update by sending SYN flood to the destination port 80.
Codered2	TCP 80	4.95	Uses a buffer overflow to exploit vulnerability on Microsoft IIS web servers. After the worm propagates itself to any host, it sends DOS attack and provides backdoors to attackers. Then, this worm will find new hosts to infect with port 80 on TCP protocol.

Forbot-FU	TCP 445	32.53	Propagates itself to other hosts with Trojan/Optix on Windows. This worm exploits buffer overflow vulnerability of Windows and provides backdoor to attackers with port 445 on TCP protocol.
Rbot.CCC	TCP 139,445	9.7	Provides backdoors and allows attackers to remotely access on the vulnerable computer via IRC channels on Windows platform. It propagates itself with ports 139 and 445 on TCP protocol.
Zotob.G	TCP 135,445, UDP 137	39.34	Exploits buffer over flow vulnerability on MS Windows Plug and Play and provides backdoors to attackers with ports 135, 445 on TCP protocol and port 137 on UDP protocol.

4.2 Data Preprocessing and Feature Selection

We used datasets from [37], and selected some attributes (Direction, Protocol, Src Port, Des Port, Time Stamp and Key Code) to be used as the details of our datasets. Then, replacement the worm key code from "d9" to "Worm" and replacement normal data key code to "Normal", in order to facilitate conducting experiments practical. The collected datasets for 3 cases of experiments by sampling of datasets from the 13 end-points, where the size of the data (no. of instances) estimated a huge number in the millions, so the sampling was manually so that achieved equality, and balancing in the number of samples for each worm, and diversity to include all forms of data instances, also for every worm, has been sampling where achieved balancing in the number of samples for each port connecting to same worm. We summarize that there are three main factors play an important role in chosen of the samples instances which are protocol, port, and direction.

For example, the worm (Blaster) that exploits by connecting to ports 135 and 4444 on TCP protocol and port 69 on UDP protocol, and has four types of direction packet. If assumed there 900 instances for Blaster worm, so the partition of this data was as follows: 600 instances for TCP protocol divided to 300 instances for each port (135, and 4444) included all types of direction packet, and 300 instances UDP protocol to port 69 included

all types of direction packet. Thus, for each type of worms to achieved equality and balancing for the data sets.

4.2.1 Cases of experiments

In the first case, the datasets used contain samples for all types of worms dataset, which have been used in our research, and normal dataset, but in individual process, means that every experiment contain one type of worms case. In second case, the datasets used contain samples for all types of worm's dataset, and normal dataset. In third case, the datasets used contain samples for all types of worms dataset except one, and normal dataset. For each case the data was partitioned into 70% training and 30% test data.

The details of each case as in section 5.2, and a summarization of these cases is shown in Table 4.4.

4.2.1.1 First case (known worm detection in individual)

Dataset used in this case is composed of 33876 profiles, where contained on **15000** normal profiles and **18876** worm profiles by sampling one worm type for every experiment.

4.2.1.2 Second case (known all worms detection)

The dataset used in this case is composed of **42275** profiles, where contained on **20000** normal profiles and **22275** worm profiles by sampling for all worm types which used in our research.

4.2.1.3 Third case (unknown worm detection)

The dataset used in this case is composed of **20000** normal profiles and **19156** worm profiles by sampling **5** worm types. In training dataset, all worm types used except one worm which is used as unknown worm in the testing dataset.

Table 4.4: Training and testing datasets for each case

Case #	Total No. of All Instances	Training phase		Testing phase		Output
		Worms	Normal	Worms	Normal	
1 (5 exp.)	33876	13213 (for each worm per exp.)	10500	5663 (for each worm per exp.)	4500	2 classes "Worm, or Normal"
2	42275	15593 (3119×5) (5 worms types)	14000	6682 (1336×5) (5 worms types)	6000	
3 (5 exp.)	39156	12474 (3119×4) (4 worms types, 5th worm in testing)	14000	6682 (1336×5) (5 worms types)	6000	

4.3 Building "WDMAC" model

To build "WDMAC" model which is an adaptive worm's detection model based on multi classifiers that able to detect known and unknown worms, we have conducted the following steps:

4.3.1 The Base Line Experiments

To select the classifiers to be used in building the "WDMAC" model, we decomposed datasets into 3 sample groups, and apply the six common algorithms of classification mentioned before: Support Vector Machines **(SVM)**, Rule Induction **(RI)**, K-Nearest Neighbor **(K-NN)**, Naïve Bayes **(NB)**, Decision Tree **(DT)**, and Artificial Neural Network **(ANN)**. The objective of this experiments is to determine which of those classifiers algorithms will be used to build the "WDMAC" model. The experiments based on the three cases that have been previously explained in section 4.2.1, the results is shown in Table 4.5(a, b, c, and d), consecutively. Also, concluding from all experiments, which shorten the these three tables of experiments in the following Table 4.5 (d), and Figure 4.2.

Table 4.5 (a): Experiments Results on Datasets of Case1.

Classifier	Overall Accuracy	Classification Error	Detection Rate	Recall	Precision	F-measure
NB	97.02	2.98	97.45	97.12	96.31	96.71
DT	98.72	1.28	98.99	98.85	98.56	98.70
ANN	99.72	0.28	99.75	99.72	99.73	99.72
K-NN	99.51	0.48	99.512	99.46	99.55	99.51
RI	88.86	11.14	88.86	90	84.43	86.14
SVM	89.74	10.26	89.74	89.97	84.85	86.13

Table 4.5 (b): Experiments Results on Datasets of Case2.

Classifier	Overall Accuracy	Classification Error	Detection Rate	Recall	Precision	F-measure
NB	95.73	4.27	95.91	95.82	95.71	95.76
DT	97.79	2.21	97.74	97.71	97.95	97.83
ANN	98.93	1.07	98.83	98.88	98.98	98.93
K-NN	99.98	0.02	99.98	99.98	99.96	99.98
RI	97.24	2.76	97.24	97.1	97.47	97.17
SVM	94.13	5.87	94.13	94.12	94.10	94.12

Table 4.5 (c): Experiments Results on Datasets of Case3.

Classifier	Overall Accuracy	Classification Error	Detection Rate	Recall	Precision	F-measure
NB	95.75	4.25	95.92	95.83	95.74	95.78
DT	96.29	3.71	96.59	96.78	96.49	96.63
ANN	96.65	3.35	96.82	96.74	96.97	96.86
K-NN	93.66	6.34	93.66	93.99	94.52	94.25
RI	90.85	9.15	90.85	91.32	92.23	91.77
SVM	81.45	18.55	81.45	81.25	81.66	81.45

Table 4.5 (d): Average of Experiments Results on Datasets of Case1, 2, and 3.

Classifier	Overall Accuracy	Classification Error	Detection Rate	Recall	Precision	F-measure
NB	96.17	3.83	96.43	96.26	95.92	96.08
DT	97.60	2.40	97.77	97.78	97.67	97.72
ANN	98.43	1.57	98.47	98.45	98.56	98.50
K-NN	97.72	2.28	97.72	97.81	98.01	97.91
RI	92.32	7.68	92.32	92.81	91.38	91.69
SVM	88.44	11.56	88.44	88.45	86.87	87.23

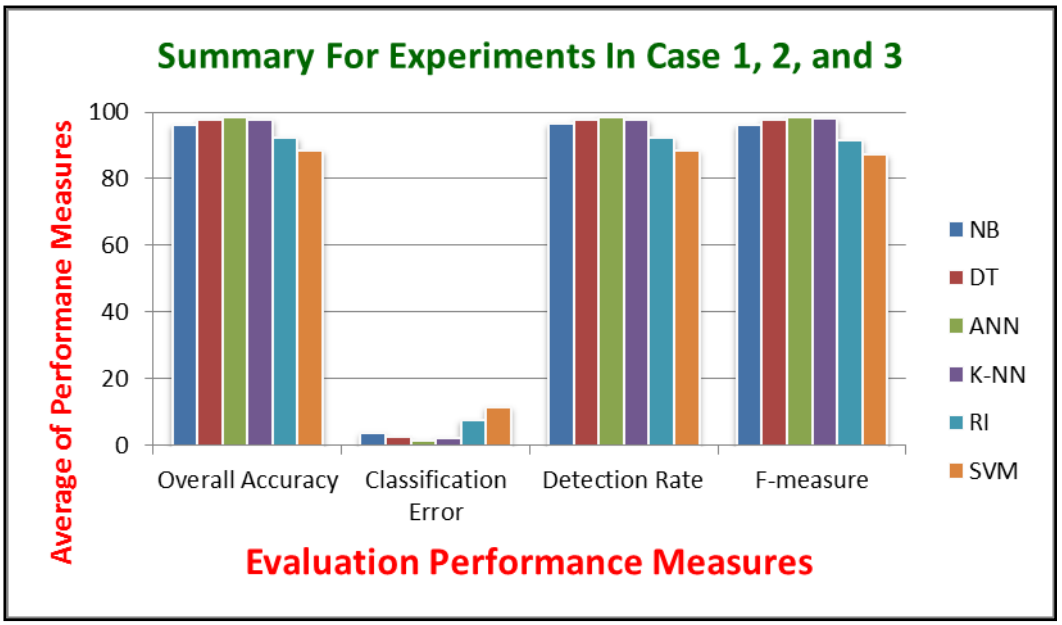

Figure 4.2: Average for each classifier on datasets of Case1, 2, and 3

4.3.2 Selection Optimal Classifiers algorithms

Based on the results which have been reached in Table 4.5(a, b, c, and d), and Figure 4.2. There are three main factors play important roles in the process of selecting the optimal classifiers which are accuracy, detection rate, and classification error. To selects one type of classifiers used in each layer we follow the following steps:

1) K-Nearest Neighbor (K-NN) algorithm achieved the highest results of Naive Bayes algorithm, but there is another factor to be taken in consideration, which is the results of NB in case 3 (unknown worm detection) was best of K-NN, where the case 3 of experiments consider the important case that was the natural case of worms detection.
2) Support Vector Machine (SVM) algorithm has been excluded because the results are not good compared with other algorithms used in this phase.
3) Rule Induction (RI) algorithm was excluded also, because the result is not good compared with other algorithms used in this phase.
4) So, the selection of suitable algorithm use in "WDMAC" model as follows: Naïve Bayes (NB) algorithm, Decision Tree (DT) algorithm, and Artificial Neural Network (ANN), where achieved the balancing between three factors.

4.4 Apply the "WDMAC" model

This section describes the types of classifiers algorithms used in our "WDMAC" model: Naïve Bayes (NB), Decision Tree (DT), and Artificial Neural Network (ANN), which are provided by RapidMiner [38] program. We present these classifiers algorithms and their settings which are used during experiments results by our model as the following:

4.4.1 Naïve Bayes (NB)

Naïve Bayes classifier used in our model is one of the most widely used classifiers, and based on the inferences of probabilistic graphic models which specify the probabilistic dependencies underlying a particular model using a graph structure. Table 4.6 explain the setting of Naïve Bayes classifier [38].

Table 4.6: Naïve Bayes Setting

Input	Output	Parameters
Training Set, expects example set.	**Model,** **Example Set,** Noted that, these set is replaced with processed testing set by researcher.	**Laplace correction:** Use it to prevent high influence of zero probabilities.

4.4.2 Decision Tree (DT)

Decision trees use simple knowledge representation to classify examples into a finite number of classes. In a typical setting, the tree nodes represent the attributes, the edges represent the possible values for a particular attribute, and the leaves are assigned with class labels. So, the tree structures are representing set of decisions. These decisions generate rules for the classification of a dataset. Table 4.7 explain the setting of Decision Tree [38].

Table 4.7: Decision Tree Parameters

Input	Output	Parameters
Training Set, expects example Set.	**Model,** **Example Set,** Noted that, these set is replaced with processed testing set by researcher.	**Criterion:** Specifies the used it for selecting attributes and numerical splits. We chose the gain ratio for the criterion term. **Minimal size for split:** The minimal size of a node in order to allow a split = 4. **Minimal leaf size:** The minimal size of all leaves = 2. **Minimal gain:** which must be achieved in order to produce a split = 0.1. **Maximal depth:** The maximum tree depth = 20. **Confidence:** used for the pessimistic error calculation of pruning = 0.25. **Number of pre pruning alternatives:** The number of alternative nodes tried when pre pruning would prevent a split = 3.

4.4.3 Artificial Neural Networks (ANN)

Artificial Neural networks are non-liner mapping structures based on the function the human brain acquires knowledge by learning. The multi-layer perceptron (MLP), trained by the back propagation (BP) algorithm, is one of the most widely used neural models for classification problems. Table 4.8 explains the settings of Artificial Neural Networks [38].

Table 4.8: Artificial Neural Network Parameters

Input	Output	Parameters
Training Set, expects example Set.	**Model**, **Example Set,** Noted that, these set is replaced with processed testing set by researcher.	**Hidden layers:** Describes the name and the size of all hidden layers. **Training cycles:** The number of training cycles used for the neural network training = 500. **Learning rate:** The learning rate determines by how much we change the weights at each step =0.3. **Momentum:** The momentum simply adds a fraction of the previous weight update to the current one = 0.2. **Decay:** Indicates if the learning rate should be decreased during learning. **Error epsilon:** The optimization is stopped if the training error gets below this epsilon value = 1.0E-5.

4.4.4 Final WDMAC Output

Final "WDMAC" model output by use three classifiers algorithms which are (Naïve Bayes, Decision Tree, and Artificial Neural Network), and combined the tree outputs to generate the final output for all models, as the final output relies on equality the output of two model as follows:

(a) If any two classifier equal "worm", and the third was "normal", so that the general output for "WDMAC" was "worm".

(b) If any two classifier equal "normal", and the third was "worm", so that the general output for "WDMAC" was "normal").

Finally, only confusion matrix will produce, where extract the results and computing the accuracy, detection rate, classification error, and f-measure.

4.5 Evaluate the "WDMAC" model

Performance evaluation of the "WDMAC" model is one of the most important tasks in our research. For the purpose of evaluating the results, we use confusion matrices that the commonly evaluation measures were a visualization tools used in supervised learning, and created for each classifier. Each column of the confusion matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The following four define the members of the matrix are: the True Positive rate (TP) (Eq. 4.1), False Positive rate (FP) (Eq. 4.2), True Negative rate (TN) (Eq. 4.3), False Negative rate (FN) (Eq. 4.4). Also, accuracy considered the most commonly to evaluate classification performance. In our research, other measures to evaluate classifiers performance, which are detection rate (Eq. 4.5), classification error (misclassification) rate (Eq. 4.6), recall (Eq. 4.7), precision (Eq. 4.8), overall accuracy (Eq. 4.9), and F-measure (Eq. 4.10), that can be defined as follows:

Confusion matrices: were created for each classifier using the actual and predicted responses [1][6]. The following four estimates define the members of the matrix (show Table 4.6).

Table 4.6: Simple Confusion Matrix

		True Class	
		Positive	**Negative**
Predicted Class	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

- **True Positive (TP):** refer to number of positive instances that correctly labeled the classifier [7].

$$True\ Pasitive\ rate = \frac{TP}{TP+FN} \qquad \text{(Eq. 4.1)}$$

- **False Positive (FP):** refer to number of negative instances that were incorrectly labeled the classifier [7].

$$False\ Positive\ rate = \frac{FP}{TN+FP} \qquad \text{(Eq. 4.2)}$$

- **True Negative (TN):** refer to number of negative instances that correctly labeled the classifier [7].

$$True\ Negative\ rate = \frac{TN}{TN+FP} \quad \text{(Eq. 4.3)}$$

- **False Negative (FN):** refer to number of positive instances that were incorrectly labeled the classifier [7].

$$False\ Negative\ rate = \frac{FN}{TP+FN} \quad \text{(Eq. 4.4)}$$

- **Detection Rate:** refer to percentage of positive instances that correctly labeled the classifier [8].

$$Detction\ Rate = \frac{N_{worm} \times |TP| + N_{normal} \times |TN|}{N_{worm} + N_{normal}} \quad \text{(Eq. 4.5)}$$

- **Classification error:** refer to relative number of misclassified, and the rate was recorded on the training and testing data sets [1].

$$Classification\ Rrror = \frac{FP+FN}{TP+TN} = 100 - overall\ accuracy \quad \text{(Eq. 4.6)}$$

- **Recall:** refer to number of positive instances that correctly labeled the classifier [4].

$$Recall = \frac{TP}{TP+FN} \quad \text{(Eq. 4.7)}$$

- **Precision:** refer to the percentage of retrieved instances that are relevant [4].

$$Precision = \frac{TP}{TP+FP} \quad \text{(Eq. 4.8)}$$

- **Accuracy:** refer the percentage of test set tuples that are correctly classified by the classifier [8].

$$Overall\ Accuracy = \frac{|TP|+|TN|}{|TP|+|FP|+|TN|+|FN|} \quad \text{(Eq. 4.9)}$$

- **F-measure:** refer to the harmonic mean of precision and recall [1][4].

$$F-measure = \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}} \quad \text{(Eq. 4.10)}$$

$$= \frac{2 \times Precision \times Recall}{Precision + Recall}$$

For Example:

We presented the example of confusion matrix on Table 4.6, and apply the previous equations which used in the evaluation measures on Blaster worm by NB classifier.

		True Class	
		Normal	**Worm**
Predicted Class	**Normal**	5849	381
	Worm	151	6301

TP = 5849 / 5849 + 151 = 0.9430 × 100 = 97.48 %

FP = 381/6301 + 381 = 0.0570 × 100 = 5.70 %

TN = 6301/6301 + 381 = 0.9430 × 100 = 94.30 %

FN = 151/5849 + 151 = 0.0252 × 100 = 2.52 %

Detection Rate = 6682 × 97.48 + 6000 × 94.30 / 6682 + 6000 = 95.98 %

Overall Accuracy = 97.48 + 94.30 / 97.48 + 5.70 + 94.30 + 2.52 = 0.9581 × 100= 95.81%

Classification Error = 5.70 + 2.52 / 97.48 + 94.30 = 0.0429 × 100 = 4.19 %

Recall = 97.48 + 94.30 / 2 = 95.89 %

Precision = ((97.48 / 97.48 + 5.70) + (94.30 / 94.30 +2.52)) / 2 = 0.9594 × 100 = 95.94 %

F-measure = 2 × 95.94 × 95.89 / 95.94 + 95.89 = 95.92 %

4.6 Adaptive Worm Detection Model Based on Multi Classifiers "WDMAC" (The Our Proposed Model)

The main objective of this research is to propose a new method of known/ unknown worms' detection. To achieve this, we used combination of classifiers as integration to be able to adapt with changes of worms nature, and to achieve higher accuracies and better detection and classification error rate,. Also, we try to overcome the drawbacks of the existing methods used in previous and related researches. For that, we propose "WDMAC" model for adaptive worms detection based on multi classifiers that is able to detect known and unknown worms.

To achieve the objective of this research, we propose the following steps shown in Figure 4.3:

Step I: Collecting datasets by sampling all datasets from the 13 end-points, and preprocessing them, then select some attributes. These dataset have been divided into 3 cases of experiments.

Step II: Dividing the data sets into two sets (training set and testing set) . The purpose of this division of data into two parts (training, and testing) is to unite the training and testing data of all classifiers model in "WDMAC" model.

Step III: For each case, we apply the "WDMAC" model as follows :

a) Apply Naïve Bayes classifier in the first step on training set to build NB model, and tested on testing set. This step will produce output (worm/normal)

b) Apply Decision Tree classifier in the second step on the same training set to build DT model, and tested on also the same testing set in previous classifier. Also, this step will produce output (worm/normal).

c) Apply Artificial Neural Networks in the last step on same training set to build DT model, and tested on the same testing set. Also, this step will produce output (worm/normal).

Step IV: We combined the tree outputs from previous steps to generate the final output for all models and final confusion matrix as the following:

(a) If any two classifier equal "worm", and the third was "normal", so that the general output for "WDMAC" was "worm".

(b) If any two classifier equal "normal", and the third was "worm", so that the general output for "WDMAC" was "normal".

Step V: Extraction results to evaluate classification performance by using the final confusion matrix from previous steps to computing overall classification accuracy, detection rate, misclassification rate, and F-measure.

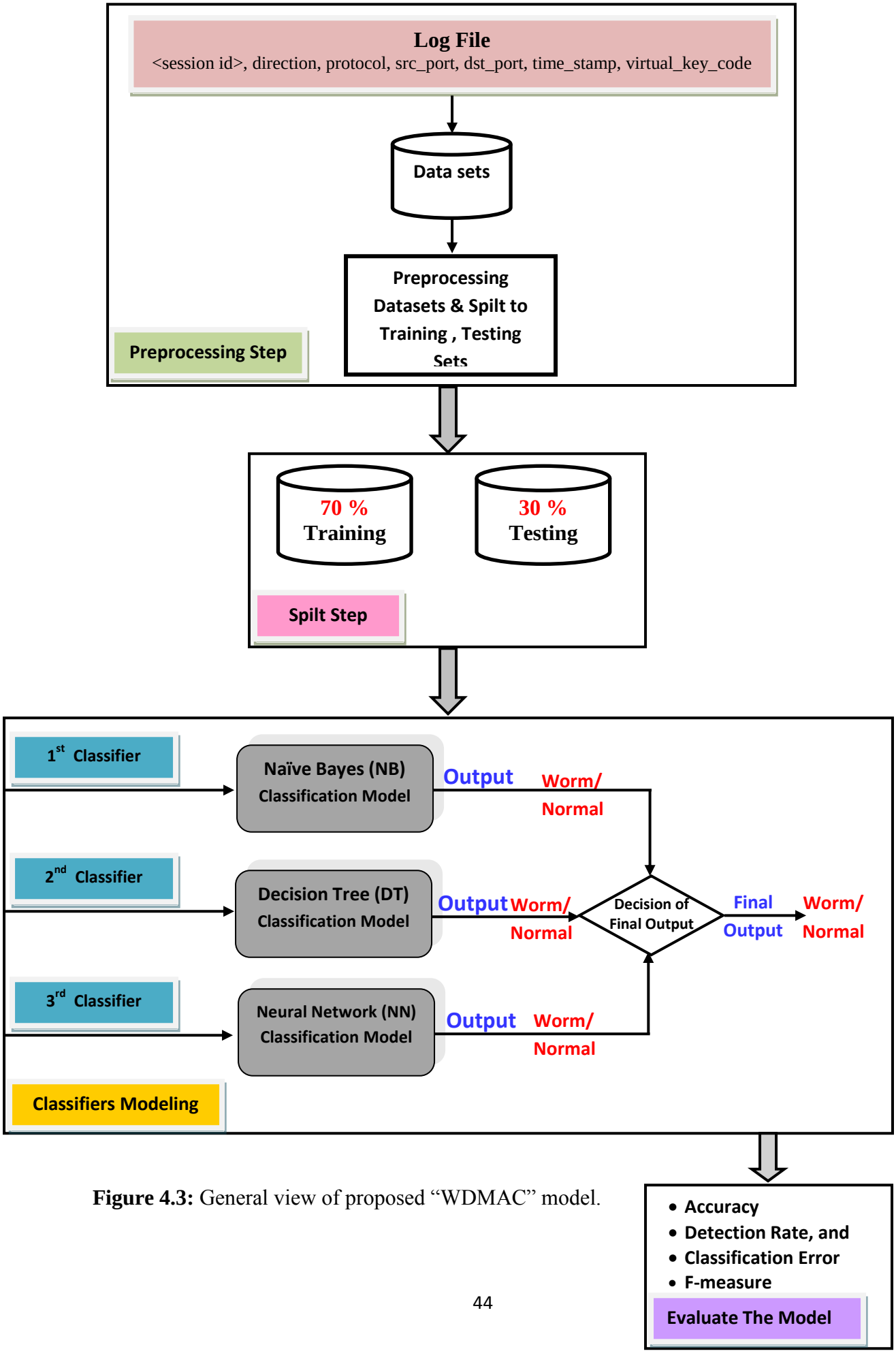

Figure 4.3: General view of proposed "WDMAC" model.

CHAPTER 5: Experimental Results and Evaluation

In this chapter we present and analyze the experiments results. Different machine learning classifiers used by combination as multi classifiers for our experiments, which are Naïve Bayes, Decision Tree, and Artificial Neural Network. We explained the machine environment and tools used in our research. Also we present the evaluation measurements for classifications model during sets of experiments by using the equation of accuracy, detection rate, classification error, and f-measure which are illustrated in section 4.5.

We apply sets of experiments scenarios on case 1, 2, and 3 of data sets, the details about these experiments and their result that have achieved presented and explained in this chapter.

5.1 Experiments Setup

In this section, A description about the experimental environment, tools used in experiments, measures of performance evaluation of classifiers and "WDMAC" model has been provided.

5.1.1 Experimental Environment and Tools

Applied to experiments on a machine with properties that are Intel Pentium Core 2 Duo P7450 @ 2.13 GHz processor and 3.00 GB of RAM. To carry out our thesis (including the experimentation), special tools and programs were used:

- **RapidMiner application program:** used to build our model, and Conduct experiments practical and extracting the required results.
- **Microsoft Excel:** used excel to partition, organize and store datasets in tables, do some simple preprocessing and analyze the results.

5.1.2 Measurements for Experiments

The measures of evaluating the performance of classification are a confusion matrices. Also to perform the comparisons of the tested algorithms, through the performance of each classifier was evaluated using the detection rate, classification error (misclassification) rate, accuracy, and F-measure. Based on the equations in section 4.5, we extracts our experiments results in the next section.

5.2 Experiment Scenarios and Results

In this section, we apply a set of experiments on 3 cases of data sets presented in section 4.2.1. In first experiments set, our model is applied on data sets for each worm types individually. In second experiments set, the "WDMAC" model is applied on data set contain all worm types discussed in our research. In last experiments set, five experiment produced that for each one we applied the "WDMAC" model on data sets contain all worm types except one to be used in test phase. The details of these experiments is explained as follows:

5.2.1 Experiment Scenario I (known worm detection)

The dataset of this experiment is divided into training and testing datasets by inserting normal data and one type of worm (i.e., Blaster). In training dataset, there are **10500** normal profiles and **13213** profiles for "Blaster" worm. The testing dataset has **4500** normal profiles and **5663** profiles for "Blaster" worm. There are **2** output classes are "worm, or normal". In this case, we perform **5** experiments where one worm-type is considered at a time. Table 5.1, illustrates dataset used in these experiments.

For example, in the first experiment, create training and testing dataset with normal and Blaster worm. In the next experiment, a different worm type is used in training and testing dataset, and so on. After completing these 5 experiments, we calculated the average of accuracy, detection rate, misclassification rate, and F-measure. Table 5.2 and Figure 5.1, illustrates experiments results in this case, which show that "WDMAC" model has achieved the best lowest classification error rate, from **0.28%** (in ANN, which is the best one) to **0.24%**, in our model.

Table 5.1: Samples of Dataset for Case 1

Data Set	No. of records	No. of Worm Types	Normal Records	Percentage of Normal	Worm Records	Percentage of Worm
Training	23713	1 per exp.	10500	44.28 %	13213	55.72 %
Testing	10163	1 per exp.	4500	44.28 %	5663	55.72 %

Table 5.2: Experiments results of case 1

Classifier	Overall Accuracy	Classification Error	Detection Rate	Recall	Precision	F-measure
Naïve Bayes	97.02	2.98	97.45	97.12	96.31	96.71
Decision Tree	98.72	1.28	98.99	98.85	98.56	98.70
Neural Network	99.72	0.28	99.75	99.72	99.73	99.72
WDMAC Model	99.76	0.24	99.74	99.76	99.80	99.78

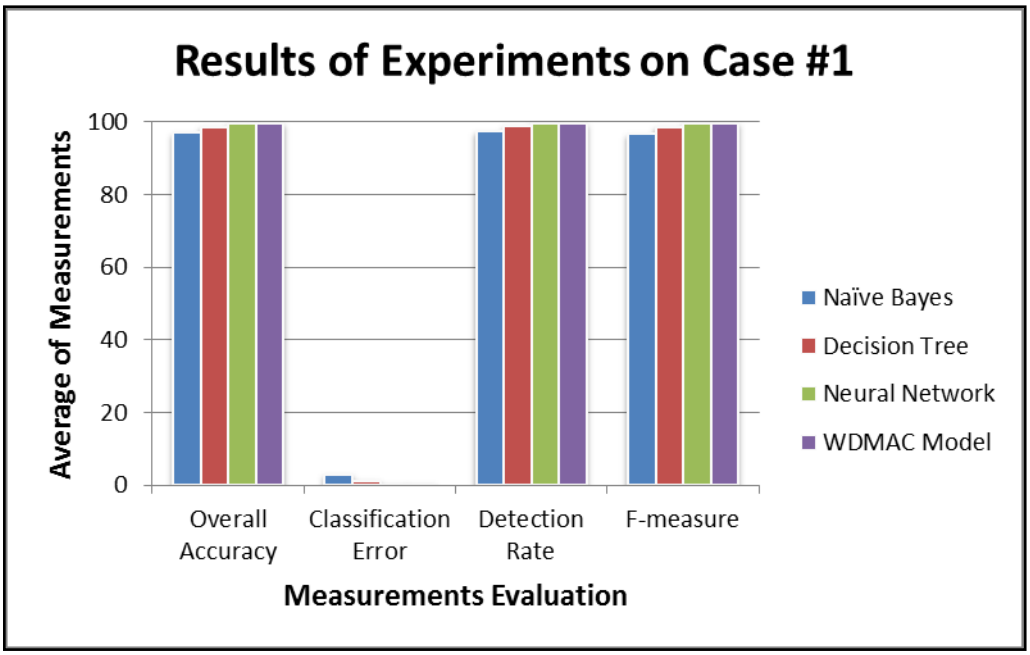

Figure 5.1: Experiments Results of case 1

5.2.2 Experiment Scenario II (known worms detection)

Dataset is divided the into training and testing datasets by inserting normal data and all **5** types of worm which are (Blaster, CodeRedII, Forbot-FU, Rbot.CCC, and Zotob.G). In training dataset, there are **14000** normal profiles and **3119** profiles for each type of worms. With **5** types of worms, the total number of worm profiles used for training is **15593** profiles. The testing dataset has **6000** normal profiles and **1336** profiles for each type of worms. There are **2** output classes **"worm, or normal"**. Table 5.3, illustrates dataset used in these experiments, and Table 5.4 and Figure 5.2, illustrates experiments results in this

case, which shows that the "WDMAC" model has achieved the lowest classification error rate with **1.70%**, while **1.07%** in ANN, which is the best one.

Table 5.3: Samples of Dataset for case 2

Data Set	No. of records	No. of Worm Types	Normal Records	Percentage of Normal	Worm Records	Percentage of Worm
Training	29593	5	14000	47.31 %	15593	52.69 %
Testing	12682	5	6000	47.31 %	6682	52.69 %

Table 5.4: Experiments results of case2

Classifier	Overall Accuracy	Classification Error	Detection Rate	Recall	Precision	F-measure
Naïve Bayes	95.73	4.27	95.91	95.82	95.71	95.76
Decision Tree	97.79	2.21	97.74	97.71	97.95	97.83
Neural Network	98.93	1.07	98.83	98.88	98.98	98.93
WDMAC Model	98.30	1.70	98.30	98.03	98.23	98.13

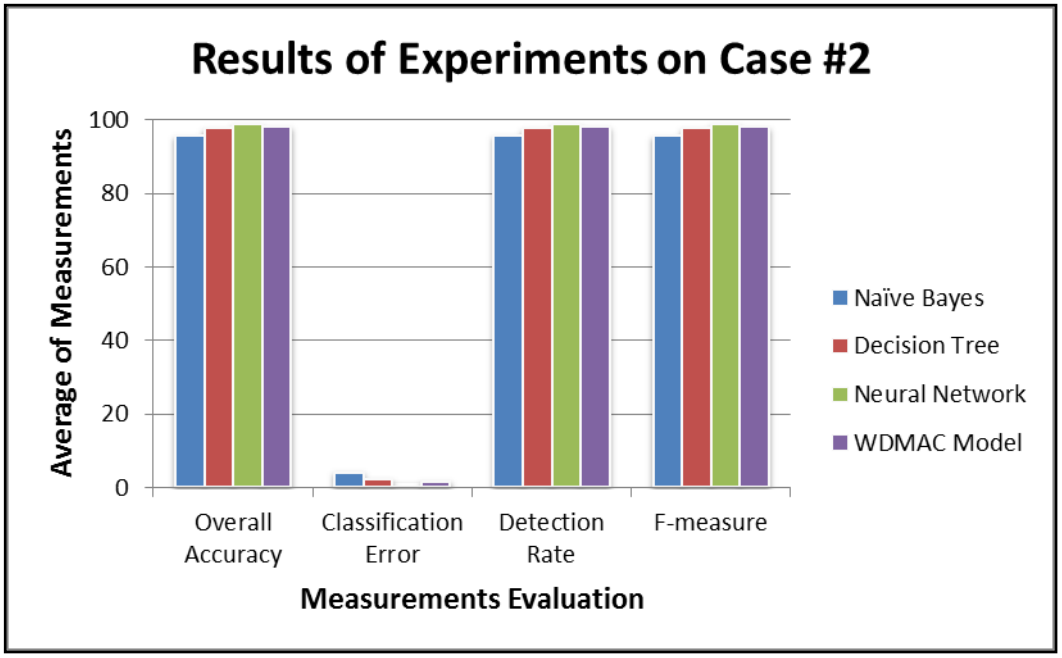

Figure 5.2: Experiments Results of case 2

5.2.3 Experiment Scenario III (unknown worm detection)

In this case, the training dataset is composed of **14000** normal profiles and **12474** worm profiles by sampling 5 worm types, except one worm which is used as unknown worm in the testing dataset. The testing dataset is composed of **6000** normal profiles and **6682** worm profiles . We add one unknown/untrained worm-type profiles into the testing dataset. There are 2 output classes which are normal and worm. In this case, we perform **5** experiments where one unknown worm-type is considered at a time. Table 5.5, illustrates dataset used in these experiments.

For example, in the first experiment, we make a training dataset without Blaster worm. Then we added the Blaster worm into the testing dataset. In the next experiment, a different worm-type is excluded from the training dataset but is included for testing, and so on. After completing these 5 experiments, we calculated the average of accuracy, detection rate, misclassification rate, and F-measure. Table 5.6 and Figure 5.3, illustrates experiments results in this case, which are appear that "WDMAC" model has achieved the best lowest classification error rate from **3.35%** (in ANN, which is the best one) to **1.95%** in our model.

Table 5.5: Samples of Dataset for case 3

Data Set	No. of records	No. of Worm Types	Normal Records	Percentage of Normal	Worm Records	Percentage of Worm
Training	26474	4	14000	52.88 %	12474	47.12 %
Testing	12682	5	6000	47.31 %	6682	52.69 %

Table 5.6: Experiments results of case 3

Classifier	Overall Accuracy	Classification Error	Detection Rate	Recall	Precision	F-measure
Naïve Bayes	95.75	4.25	95.92	95.83	95.74	95.78
Decision Tree	96.29	3.71	96.59	96.78	96.49	96.63
Neural Network	96.65	3.35	96.82	96.74	96.97	96.86
WDMAC Model	98.05	1.95	97.99	98.03	98.10	98.06

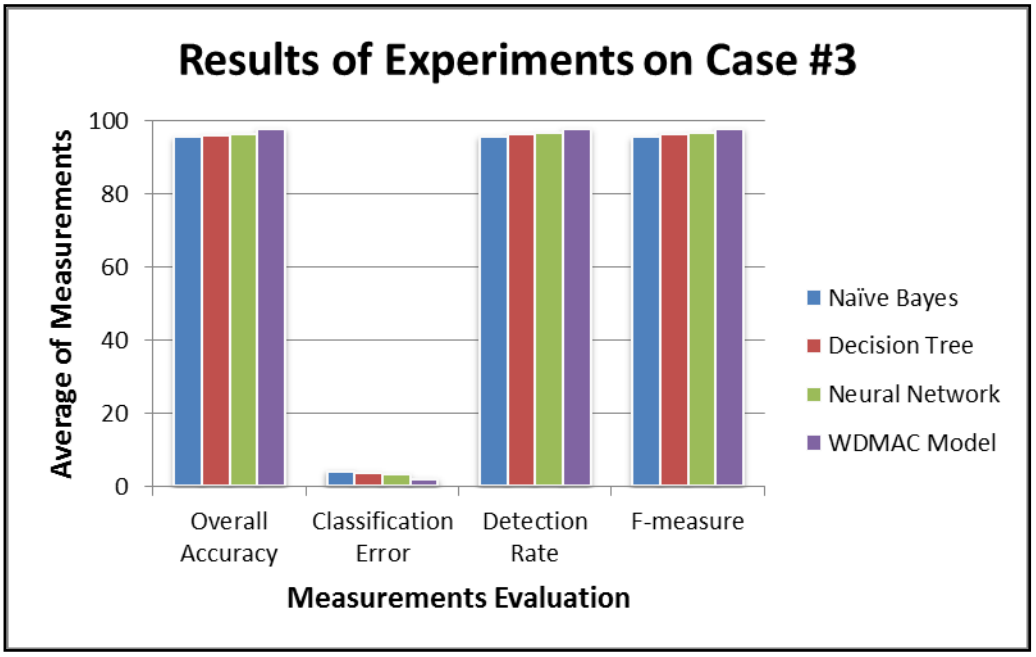

Figure 5.3: Experiments Results of case 3

Based on experiments case 3, the detection rate for each experiment in Table 5.7. We noted that the average for detection rate of all classifiers algorithms ranges between 92.29% and 98.19%, while the "WDMAC" model achieved detection rate ranges between 96.47% and 98.85%. So, as a result we can say that "WDMAC" model has achieved a better detection rate and classification error rate. The average detection rate for all classifiers algorithms is 96.44% with 3.77% classification error, Where "WDMAC" model achieved 97.99% to detection rate, and 2.01% for classification error.

Table 5.7: Experiments results of case 3 For each worm

Worm / Model	Detection Rate for each worm					
	Blaster	**CodeRedII**	**Forbot-FU**	**Rbot.CCC**	**Zotob.G**	**Average**
Naïve Bayes	95.98	96.00	95.84	95.94	95.84	95.92
Decision Tree	99.95	98.19	90.54	94.31	99.95	96.59
Neural Network	98.64	98.82	90.50	98.58	97.56	97.02
Average	98.19	97.67	92.29	96.28	97.78	96.44
WDMAC Model	98.64	98.85	96.47	98.39	97.60	97.99

5.2.4 Forbot-FU Worm Results in Case 1,and 3

For example, we present the result for Forbot-FU worm in case 1, and 3 which reported a lower results. Where in case 1, were training, and testing on samples of normal, and Forbot-FU worm instances. So the number of misclassification ranges between (9-649) instances. While in case 3, were the training on samples of normal, and worms instances except Forbot-FU instance, which are inject in testing instances. The numbers of misclassification was ranges (576-707). The "WDMAC" model able to reduces classification error from 10.58% to 3.47 % in the worst case3. The result illustrates in Table 5.8.

Table 5.8: The results for Forbot-FU worm in case1, and 3

Case of Experiments	Classifier	Overall Accuracy	Classification Error	Detection Rate	F-measure	No. of Misclassification
Case 1	Naïve Bayes	99.91	0.09	99.93	99.91	9
	Decision Tree	93.61	6.39	94.93	93.52	649
	Neural Network	99.9	1.31	99.92	99.90	10
	WDMAC Model	99.89	0.11	99.91	99.88	11
Case 3	Naïve Bayes	95.66	4.46	95.65	95.69	576
	Decision Tree	89.47	10.53	89.47	90.45	704
	Neural Network	89.42	10.58	89.42	90.41	707
	WDMAC Model	96.53	3.47	96.47	96.57	434

5.2.5 Experiment Scenario IV (for comparison case)

Sarnsuwan et-al [8], used the data sets from [37] in their experiments. The training set was 40%, and testing set 60% , we used the same numbers of data sets as the following:

Case 2:

Inserted **750** normal profiles and **1200** worm profiles for the training dataset and **1750** normal profiles and **2800** worm profiles for the testing dataset. The worm class consists of all types of worm profiles (i.e., Zotob.G, CodeRedII, Blaster, Rbot.CCC, Rbot.AQJ, Sdbot-AFR, SoBig.E and Forbot-FU profiles). There are **2** classes in these datasets as normal and worm class.

We calculated the results of case 2 in Table 5.9 depend on Table 6 from [8]. The experiments results explained that "WDMAC" model has achieved the best lowest classification error rate 0.42%, highest accuracy 99.58%, and detection rate with 99.33%.

Table 5.9: Experiments results of case 2

Classifier	Overall Accuracy	Classification Error	Detection Rate	Recall	Precision	F-measure
Naïve Bayes	99.25	0.75	99.20	99.10	99.40	99.25
Decision Tree	98.65	1.35	98.70	98.40	98.89	98.65
Random Forest	98.85	1.15	99.00	98.40	99.29	98.84
WDMAC Model	99.58	0.42	99.33	99.45	99.88	99.66

Case 3:

The training dataset is composed of **750** normal profiles and **1200** worm profiles by sampling **7** worm types, except one worm which is used as unknown worm. The testing dataset is composed of **1700** normal profiles and **2800** worm profiles. Injected one unknown/untrained worm-type profiles into the testing dataset. There are 2 output classes which are normal and worm.

We calculated the results of case 3 in Table 5.10 depend on Table 7 from [8]. The experiments results showed that "WDMAC" model has achieved the best average detection rate with 99.79%, while average detection rate for classifiers algorithms was 91.60%.

Table 5.10: Experiments results of case 3 For each worm

Worm Detection Rate / Model	Blaster	CodeRedII	Forbot-FU	Rbot.CCC	Zotob.G	Rbot.AQJ	Sdbot-AFR	SoBig.E	Average
Naïve Bayes	81.30	96.40	98.70	99.20	40.30	94.30	38.30	95.8	80.50
Decision Tree	98.20	99.3	98.80	93.80	98.6	94.40	99.00	98.8	97.60
Random Forest	98.7	86.50	98.90	99.3	94.50	98.8	99.50	97.9	91.60
Average	94.10	77.80	97.50	78.90	95.80	97.40	98.8	92.7	91.60
WDMAC Model	97.27	98.14	97.34	98.27	97.05	95.97	99.79	97.23	97.63

5.3 Discussion and summary

The following Table 5.11 and Figure 5.4 show the summary of all experiments results.

Table 5.11: The Average of accuracy, Detection Rate, Misclassification Rate, and F-measure comparison of The Models: Baseline and "WDMAC" Model For all Data Sets in Case 1, 2, and 3.

Classifier	Overall Accuracy	Classification Error	Detection Rate	Recall	Precision	F-measure
Naïve Bayes	96.17	3.83	96.43	96.26	95.92	96.08
Decision Tree	97.60	2.40	97.77	97.78	97.67	97.72
Neural Network	98.43	1.57	98.47	98.45	98.56	98.50
WDMAC Model	98.70	1.30	98.68	98.61	98.71	98.66

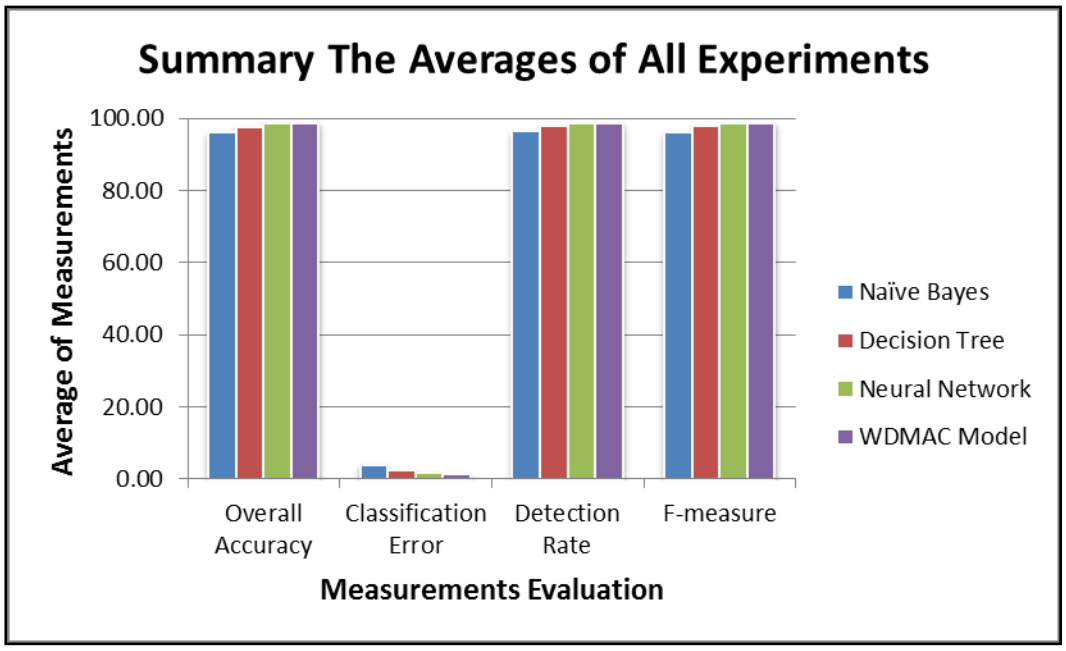

Figure 5.4: Summary The Averages of All Experiments

From our experiment, we consider our classification results in terms of accuracy, detection rate and misclassification rate. Three different data mining models (Naïve Bayes, Decision Tree, and Artificial Neural Network) are evaluated one by one, also our model has been evaluated. From Table 5.2, 5.4, 5.6, and 5.8, each of data mining classification models and our model can classify normal and worm profile in accuracy, detection rate, misclassification, and F-measure.

We can summarize our experiments results as follows:

a) The experiments on datasets of case 1 achieved the highest accuracy, detection rate, F-measure results (99.76%), and lowest misclassification rate (0.24) were in our model.

b) The experiments on datasets of case 2 achieved the highest accuracy (98.30%), highest detection rate (98.30%), F-measure (98.13%), and lowest misclassification rate (1.70) were in our model.

c) The experiments on datasets of case 3 achieved the highest accuracy (98.05%), highest detection rate (97.99%), F-measure (98.06%), and lowest misclassification rate (1.95) were in our model.

d) In general, we can say that our model has achieved good results from the all experiments on datasets of case 1, 2, and 3 where the highest accuracy was (98.70%), and detection rate results was (98.68%), lowest misclassification rate (1.30), and F-measure (98.66%) were in "WDMAC" model. The reason for this misclassification that refer to the classifiers unable to detected for these instances, because some worms have port profiles similar to those in the normal data profiles that may cause difficulty for worm detection.

e) In addition, we achieved good results for accuracy between (98.05 % - 99.76 %), misclassification rate between (0.24 % - 1.95 %), detection rate between (97.99% - 99.74%), and F-measure between (98.06% - 99.78%).

CHAPTER 6: Conclusion and Future work

Currently, internet worm is a critical threat on computer network. To detect internet worm, several research approaches have been proposed, where worm detection is based on intrusion detection system (IDS). Internet worm based IDS can be separated into two categories. The first one is network-based and another one is host-based. The network-based internet worm detection considers network packets before they reach to an end-host, while the host-based internet worm detection considers network packets that already reached to the end-host. Most recent researches were presented "Worms Detection" approaches based on classification techniques in data mining as an efficient ways to increase the security of networks.

This chapter concludes the work, its results and discussion. Finally the future work directions were remarked.

6.1 Conclusion

In our research, we present three efficient classification techniques in data mining, which are Naïve Bayes **(NB)**, Decision Tree **(DT)**, and Artificial Neural Network **(ANN)**. These techniques were used in applying "WDMAC" model. We proposed "WDMAC" which is an adaptive model based on multi classification that able to be detecting known and unknown worms. The purpose of used multi classification was to obtain the highest accuracy and detection rates, and reduced misclassification rates. This dissertation research composed of 4 phases:

Phase 1: collection of data sets from [37], and selection of some attributes such as Direction, Protocol, Src Port, Des Port, Time Stamp and Key Code columns to be used as our datasets. Then, replacing the worm key code from "d9" to "Worm" and replacement of normal data key code to "Normal", in order to facilitate conducting experiments practical. The collected datasets for 3 cases of experiments by sampling all datasets from the 13 end-points.

Phase 2: we used RapidMiner program to apply our model, we have conducted a series of experiments to determine the three classifiers used in our "WDMAC" model which are Naïve Bayes (NB), Decision Tree (DT), and Artificial Neural Network (ANN).

Phase 3: we used accuracy, misclassification rate, detection rate, and F-measure to evaluate the classification performance of our model. The accuracy of the three

classification models achieved 96.17%, 97.60%, and 98.43% , and misclassification rate with 3.83%, 2.40 %, and 1.57 % for Naïve Bayes, Decision Tree, and Artificial Neural Network respectively. While "WDMAC" model achieved the highest accuracy with 98.70%, detection rate with 98.68%, F-measure with 98.66%, and lowest misclassification rate with 1.30 %.

Phase 4: we have some comparisons to prove that the "WDMAC" model has achieved high results in the accuracy and detection rates and reduced the misclassification rates, through the comparisons of performance by using for each classifier algorithm as independently, and by using the "WDMAC" model for each case of dataset, another comparisons that between our model and other research which can be used for worm detection.

To confirm our conclusion, the first comparison in Table 6.1 (a, and b) on case 2, and 3 of experiments, compares our research with published research Sarnsuwan et-al [8] in the field of worms detection domain.

Table 6.1(a): Summary table for compare between related work [8]

Research	Case	Overall Accuracy	Classification Error	Detection Rate	Recall	Precision	F-measure
WDMAC Model	2	99.58	0.42	99.33	99.45	99.88	99.66
Sarnsuwan et-al [8]		98.92	1.08	98.97	98.63	99.20	98.91

Table 6.1 (b): Summary table for compare between related work [8]

Detection Rate / Research	Case	Blaster	CodeRedII	Forbot-FU	Rbot.CCC	Zotob.G	Rbot.AQJ	Sdbot-AFR	SoBig.E	Average
WDMAC Model	3	97.27	98.14	97.34	98.27	97.05	95.97	99.79	97.23	97.63
Sarnsuwan et-al [8]		94.10	77.80	97.50	78.90	95.80	97.40	98.8	92.7	91.60

The second comparison in Table 6.2 compares our research with published researches in the field of worms detection domain based on anomaly-behavior detection approaches.

Table 6.2: Summary table for compare between related works

Research	Model	IDS Approach	Detection Rate	Accuracy	Classification Error
WDMAC Model	NB, DT, and ANN	Host-Based	98.68 %	98.70 %	1.30 %
Sarnsuwan et-al [8]	NB, DT, and RF	Host-Based	97.52 %	-	-
Sarnsuwan et-al [28]	NB, DT, and RF	Network-Based	96.6 %	-	-
Wang et-al [15]	NB, and SVM	Host-Based	97.95 %	-	4.45 %
Ismail et-al [13]	NB, J48, SMO, and Winnow	Host-Based	-	96.50 %	3.50 %
Moskovitch et-al [10]	ANN, NB, DT, and BN	Host-Based	-	94.5 %	5.5 %
Farag et-al [25]	ANN	Host-Based	-	99.96 %	0.04 %
Siddiqui et-al [11],	DT,RF, and Bagging	Host-Based	95.6 %	96.2 %	3.8 %

We can concluded that "WDMAC" model achieved the best results for performance measurements which are detection rate, accuracy, classification error, and F-measure, by using large data sets from different environments which are homes, offices, and universities over 12 months, in addition using different worms types.

6.2 Future Work

- Generally, most worms have behaviors similar to those of the Port Scan and Denial of Service (DoS) attacks. So, to measure the evaluation of the model using these types of attacks. Also additional types of malwares such as Viruses, Trojans and so on.
- Some worms have port profiles similar to the normal data profiles that may cause difficulty for worm detection. So, we suggest to increasing the types of worms, and increase in data sets volumes in training, and testing set.
- There is still a classification error rate, there are some suggested solutions in an attempt to solve it, including increase numbers of layers with other classifiers, or modify in the parameters of classifiers model, or possibly using multi classifiers in each layer.
- Try to find a new method to detect worms by using collaborative detection (misuse and anomaly-behavior detection) in conjunction with one another.
- Modify the model to classify many types of the worm at network end point.
- Try to using clustering methods to build the model to detect known and unknown worms.
- Try to find a new method to detect worms by using collaborative methods classification and clustering in conjunction with one another.

References

[1] Ye N., “The Handbook Of Data Mining”, Lawrence Erlbaum Associates, Inc., 2003.

[2] Kruegel Ch. Valeur F.; and Giovanni ; “Intrusion Detection and Correlation, Challenges and Solutions”. Book on Advances in Information Security (Volume 14), Springer.

[3] Law K.; and Kwok L.; "IDS False Alarm Filtering Using KNN Classifier", Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 2005.

[4] Olson D., and Delen D., “Advanced data mining techniques”, Springer-Verlag Berlin Heidelberg, 2008.

[5] Hand D., Mannila H., and Smyth P., “Principles of Data Mining”, Cambridge, Massachusetts Institute of Technology, 2001.

[6] Witten L., and Frank E., ” Data Mining: Practical Machine Learning Tools and Techniques”, The Morgan Kaufmann Series in Data Management Systems, (2nd Edition), 2005.

[7] Han J., and Kamber M., “Data Mining: concepts and techniques”, (2nd Edition), the Morgan Kaufmann Series in Data Management Systems, 2006.

[8] Sarnsuwan N.; Wattanapongsakorn N.; and Charnsripinyo Ch., "Internet Worm Detection and Classification with Data Mining Approaches", The 13th National computer science and engineering conference: green computing technology,2009.

[9] Aiello M.; Avanzini D.; Chiarella D.; and Papaleo G.; "Worm Detection Using E-mail Data Mining", Primo Workshop Italiano su PRIvacy e SEcurity, Prise 2006.

[10] Moskovitch R.; Gus I.; Pluderman Sh.; Stopel, D.; Feher, C.; Glezer, Ch.; Shahar, Y.; and Elovici, Y,; “Detection of Unknown Computer Worms Activity Based on Computer Behavior using Data Mining”, Conference of Computational Intelligence in Security and Defense Applications, CISDA, IEEE Symposium, 2007.

[11] Siddiqui M.; Wang M.; “Detecting InternetWorms Using Data Mining Techniques”, Cybernetics and Information Technologies,Systems and Applications (CITSA), 2008.

[12] Ellis D.; Aiken J.; Attwood K.; and Tenaglia S.; "A Behavioral Approach to Worm Detection", Proceedings of the 2004 ACM workshop on Rapid malcode, 2004.

[13] Ismail I.; MarsonoM.; and Nor S.; " Detecting Worms Using Data Mining Techniques: Learning in the Presence of Class Noise ", Sixth International Conference on Signal-Image Technology and Internet Based Systems, 2010.

[14] Sharma O.; Girolami M.; Sventek J.; "Detecting Worm Variants using Machine Learning", CoNEXT’ 07, December 10-13, 2007, New York, NY, U.S.A., 2007.

[15] Wang X.; Yu W.; Champion A.; Fu X.; and Xuan D.; "Detecting Worms via Mining Dynamic Program Execution", Authorized licensed use limited to: The Ohio State University, 2008.

[16] Stopel D.; Boger Z.; Moskovitch R.; Shahar Y.;and Elovici Y.; "Improving Worm Detection with Artificial Neural Networks through Feature Selection and Temporal Analysis Techniques", International Journal of Mathematical and Computer Sciences, 2005.

[17] Ashfaq A.; Javed M.; and Khayam S.; " An Information-Theoretic Combining Method for Multi-Classifier Anomaly Detection Systems", IEEE Communications Society subject matter experts for publication in the IEEE ICC, 2010.

[18] Li P., Salour M., and Su X., "A Survey of Internet Worm Detection and Containment". Communications Surveys & Tutorials, IEEE, 2008.

[19] Stopel, D.; Boger, Z.; Moskovitch, R.; Shahar, Y.;and Elovici, Y.; "Application of Artificial Neural Networks Techniques to Computer Worm Detection", International Conference on Neural Networks ICNN International Journal of Applied Mathematics and Computer Sciences, 2006.

[20] Masud M., "E-Mail Worm Detection Using Data Mining", International Journal of Information Security and Privacy (IJISP), 2007.

[21] Douligeris Ch.; and Serpanos D.; "Network Security Current Status and Future Directions", IEEE Press, 2007.

[22] Pietro R.; and Mancini L.; "Intrusion Detection Systems", Springer Science and Business Media, LLC., 2008.

[23] Weaver N.; "Warhol Worms: The Potential for Very Fast Internet Plagues", The Regents of the University of California, 2001.

[24] Weaver N., Paxson V., Staniford S. and Cunningham R., "Taxonomy of computer worms," Proc of the ACM workshop on Rapid malcode, WORM03, 2003.

[25] Farag I.; Shouman M.; Sobh T.; and El-Fiqi H.;"Intelligent System for Worm Detection", International Arab Journal of e-Technology, 2009.

[26] Mannan M.; and Oorschot P.;"On Instant Messaging Worms- Analysis and Countermeasures," in Proceedings of the 2005 ACM workshop on Rapid Malcode, Fairfax, VA, USA, 2005.

[27] Rasheed M.; Norwawi N.; Ghazali O.; Kadhum M.; "Intelligent Failure Connection Algorithm for Detecting Internet Worms", International Journal of Computer Science and Network Security, 2009.

[28] Sarnsuwan N.; Wattanapongsakorn N.; and Charnsripinyo Ch., “A New Approach for Internet Worm Detection and Classification”, Networked Computing (INC), 6th International Conference, 2010, and The 13th National computer science and engineering conference: green computing technology,2009.

[29] Schneider A., “Methods of Internet Worm Propagation”, Wake Forest University, Department of Computer Science, Honors Thesis, 2009.

[30] Tang Y.; Doctor’s Thesis in " Defending Against Internet Worms ", University Of Florida, 2006.

[31] Dey Ch.; Master's Thesis in " Reducing IDS false positives using Incremental Stream Clustering (ISC) Algorithm ", Dept of Computer and Systems Sciences, Royal Institute of Technology, Sweden, 2009.

[32] Fosnock C.; “Computer Worms: Past, Present and Future”, East Carolina University, 2005.

[33] Source: http://en.wikipedia.org/wiki/Artificial_neural_network, (2011, August), [Online].

[34] Naive Bayes classifier Source: http://en.wikipedia.org/w/index.php?oldid=411315852, (2012, April), [Online].

[35] Source: http://en.wikipedia.org/wiki/Intrusion_detection_system, (2011, August), [Online].

[36] Wireless and Secure Networks (WiSNet) Research Lab at the NUST School of Electrical Engineering and Computer Science (SEECS), http://www.wisnet.seecs.nust.edu.pk/projects/adeval/endpoint_dataset/Readme_EndpointDataset.pdf , (2011, October), [Online].

[37] Wireless and Secure Networks (WiSNet) Research Lab at the NUST School of Electrical Engineering and Computer Science (SEECS), http://www.wisnet.seecs.nust.edu.pk, (2011, October), [Online].

[38] Rapid Miner 5.1, http://www.rapidminer.com , (2011, October), [Online]

[39] Source: http://macs.about.com/od/glossaryuz/g/worm.htm, (2011, August), [Online].

[40] source: http://searchsecurity.techtarget.com/definition/zero-day-exploit, (2012, June), [Online].