An Enhanced Hopfield Neural Network Model for Misuse Intrusion Detection System

نموذج محسن لشبكة هوبفيلد العصبية لكشف التطفل في نظام المعلومات

Prepared By

Ziad Jameel Al-Nuimat

Supervisor

Prof. Reyadh Shaker Naoum

Master Thesis

Submitted in Partial Fulfillment of the requirements of the Master Degree In Computer Science

Department of Computer Science

Faculty of Information Technology

Middle East University

Amman – Jordan

January, 2013

Middle East University

AUTHORIZATION STATEMENT

I, Ziad Jameel Ibrahim Almsimat, authorize Middle East University to supply hardcopies and electronic copies of my thesis to libraries, establishments, or bodies and institutions concerned with research and scientific studies upon request, according to the university regulations.

Name : Ziad Jameel Ibrahim Almsimat

Date : 05 /01/2013

Signature :

جامعة الشرق الأوسط

التفويض

أنا زياد جميل النعيمات أفوض جامعة الشرق الأوسط بتزويد نسخ من رسالتي ورقيا والكترونيا للمكتبات، أو المنظمات، أو الهيئات والمؤسسات المعنية بالأبحاث والدراسات العلمية عند طلبها.

الاسم: زياد جميل النعيمات
التاريخ: ٢٠١٣/٠١/٠٥
التوقيع:

Middle East University

Examination Committee Decision

This is to certify that the thesis entitled **"An Enhanced Hopfield Neural Network Model for Misuse Intrusion Detection System"** was successfully defend and approved on January 5th 2013.

Examination Committee Member		Signature
1- Prof. Reyadh Shaker Naoum Dean of Information Technology College ,Department of Computer Science Faculty of Information Technology (Middle East University)	Supervisor and Chairman	R. Naoum .....
2- Dr. Hussein H.Owaied Associate Professor Department of Computer Science Faculty of Information Technology (Middle East University)	Member	
3- Dr. Mohammad Ali Abbadi Associate Professor Department Of Information Technology Faculty of Science (Mutah University)	Member	

Dedication

I dedicate this work to my father, my mother, my wife and partner in life and my children for their love, understanding and support; they were the light in my path. Without them, nothing of this would have been possible. Thank you for everything. I love you!

Acknowledgment

“In the name of Allah the Most Gracious the Most Merciful”. My .guidance cannot come except from Allah, in Him I trust, to Him I repent, and to Him praise and thanks always go.

First of all ,I would like to thank Prof. Reyadh Shaker Naoum for his guidance and support, both during the stage of developing ideas as well as during the writing of the thesis.To him I offer my sincerest gratitude for his valuable contributions, knowledge, encouragement and helpful advice and vision that brought this work forward, and for being there any time I knocked on his door. I wish to him more and more success and giving.

I am highly indebted to my parents; they taught me the right, encouraged me and gave me hope and unconditional love. I wish to both of them happiness and well health. Thanks for them for supporting me.I thank all my frinds especially Mohammad Al-Ali,who read this thesis and did his best to correct the language mistakes.

My wife ,son and daughter have been more than helpful to push me in the beginning of the graduation project. Furthermore, I would like to thank my family for everything they had to do during writing my thesis and for their support and love throughout these years. They have always made my life meaningful. I am also grateful to the perfect team that helped me in all the stages of writing this thesis.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviations	Meaning
ANN	Artificial Neural Network
CSI	Computer Security Institute
DoS	Denial Of Service
DR	Detection Rate
FAR	False Alarm Rate
FN	False Negative
FP	False Positive
FTP	File Transfer Protocol
HIDS	Host-based Intrusion Detection Systems
HIDS	Host-based Intrusion Detection Systems
HNKMIDS	Hopfield Neural K-means Intrusion Detection System
HNKNNIDS	Hopfield Neural K-nearest neighbor Intrusion Detection System
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDPS	Intrusion Detection and Prevention Systems
IDS	Intrusion Detection System
IDS	Intrusion Detection System
INNS	International Neural Network Society
KDD	Knowledge –Discovery and Data mining
MIDS	Misuse Intrusion Detection System
MSE	Mean Square Error
NIDS	Network-based Intrusion Detection System
PR	Precision Rate
Prob.	Probing
R2U	Remote to User
RR	Recall Rate
RST	Rough Set Theory
SVM	Support Vector Machine
TCP	Protocol Transfer Control
TN TP	True Negative. True Positive
U2R	User to Root
UDP	User Datagram Protocol

الملخص

نظرا للتوسع الكبير في استخدام شبكات الحاسوب على مدى القرن الماضي,أصبحت أنظمة الحماية واحدة من القضايا الاكثر أهمية في انظمة الحاسوب بسبب وجود الثغرات في معظم مكونات أنظمة الحماية مثل نظام ال FIREWALL . في السنوات الاخيرة تم اقتراح وتصميم وتطوير عدة ابحاث تهدف في اساسها الى تصميم أنظمة كشف التطفل وذلك لحماية النظام وتوقع سلوك المستخدمين .تعرف أنظمة كشف التطفل على أنها العملية التي تعتمد على انماط الهجوم في مصدر البيانات لتحديد الحالات من هجمات الشبكة من خلال مقارنة النشاط الجاري مع الاجراءات المتوقعة للمتطفل .لذا يتم استخدام (IDS) كنظام حماية ثانوية لتحديد وتجنب الثغرات أو أي نشاط غير قانوني(غير طبيعي) ,أن مشكلة نظم كشف التطفل بالتعرف على الانماط الطبيعية وغيره الطبيعية (Dos,Prob.,R2L,U2R) هي تدريب الشبكة العصبة للتعرف على هذه الانماط.

في هذه الرسالة تم تطوير نموذجين بأستخدام شبكة هوبفيلد العصبية المحسنة ,الاول (HNKMIDS) ويكون بأستخدام شبكة هوبفيلد العصبية المحسنة وخوارزمية (K-means), والاخر(HNKNNIDS) ويكون بأستخدام شبكة هوبفيلد العصبية المحسنة وخوارزمية (K-nearest neighbor) بحيث يحتوي كل منهما على ثلاث مراحل:-

المرحلة الاولى : (مرحلة التصنيف) تم أستخدام خوارزمية (K-means) للنموذج الاول(HNKMIDS) وخوارزمية (K-Nearest neighbor) للنموذج الثاني (HNKNNIDS).

المرحلة الثانية : (مرحلة التدريب) تم أستخدام شبكة هوبفيلد العصبية المحسنة.

المرحلة الثالثة : (مرحلة الفحص) وتتم بأستخدام (SVM).

وقد أظهرت النتائج أن النموذجين، HNKMIDS و HNKNNIDS كانا قادرين على تصنيف الانماط الطبيعية وانماط الاقتحام مع معدل اكتشاف جيدة و وقت أقل . في النموذجين قمنا باستخدم مجموعة البيانات في KDD Cup'99 لكشف التطفل .

وقد أظهرت النتائج للنموذجين المقترحين مقدرة عالية على التصنيف بالنسبة للنموذج الاول بنسبة 99.38% ومعدل دقة 99.39% و بمعدل تصنيف للنموذج الثاني بنسبة 81.08% ومعدل دقة 94.69% وبالتالي فأن كلا النموذجين المقترحين قدما تحسينات كبيرة الى (FPR,FNR and Accuracy Rate) بالمقارنة مع الخوارزميات الاخرى.

Abstract

According to the rapid expansion of networks over the past century, system protection has become one of the most important issues in Computer Systems due to the existence of gaps in most of the components of protection systems such as FIREWALL systems. In the last past years, several research were proposed, developed and designed to set ideas based on several techniques to design systems intrusion detection to protect the system, analyze and expect the behaviors of users. Misuse intrusion detection is the process that searches attack patterns in the source of data to identify instances of network attacks by comparing current activity against the estimated actions of an intruder. Thus intrusion detection systems (IDS) are used as secondary computer systems protector to identify and avoid illegal activities or gaps. The intrusion detection problem is considered as apattern recognition, and the artificial neural network must be trained to distinguish between normal and unusual patterns (DoS, Prob., R2L, U2R).

In this thesis, two hybrid neural models were developed; Enhanced Hopfield neural network with K-means clustering algorithms(HNKMIDS) and Enhanced Hopfield neural network with K-Nearest clustering algorithms (HNKNNIDS). The two models consist of three phases:

Phase one: - In this phase, K-means clustering algorithms or K-nearest neighbor are used (clustering phase).

Phase two: - Enhanced Hopfield artificial neural network is used in this phase (Training Phase).

Phase three: Multi-class support vector machine is used in this phase (Testing Phase).

Our results have shown that the two models, HNKMIDS and HNKNNIDS, were able to classify normal class and intrusion classes with good detection rate during less time. In the two models, for evaluation, the KDD Cup’99 network used in misuse intrusion detection data set.

The result, from using the two models, demonstrates that the two proposed model have detection rate as follows

- The first model HNKMIDS, has a Classification rate of about 99.38% with Accuracy rate 99.39%.
- The second model HNKNNIDS, has a Classification rate of about 81.08% with Accuracy rate 94.69%.

Thus, the two proposed models produce substantial improvements (FPR, FNR and Accuracy Rate) over other algorithms.

Keywords: **Intrusion Detection System, Misuse Intrusion Detection System, Information Systems, Hopfield Neural Networks and Computer Security**

Chapter One

Introduction

Chapter one
Introduction

1.1 Introduction.

Security has become more and more important in our life according to development. During the past years, the concepts of security involved considering the process of assessing computer system, network and file, scanning, analyzing system information from various areas, observing and analyzing both user and system activities to identify possible security violations which include both intrusions (attacks from outsider) and misuse (attacks from inside the organization).

A technology that is developed to assess the security of computer systems or network is one of the most popular types of security management system for computers and networks which is defined as intrusion detection system. So, the increasing numbers of various attacks on major sites and networks construe ID systems are being developed.

The preservation of security has become more difficult by time because the possible technologies of attack are becoming more superior. At the same time, less technical ability is required for the novice snoopier because the verified past methods are easily accessed through the organization.

The main idea of protecting the information through the encrypted channel for data and also confirming the identity of the connected device through the firewall, which will not accept any connection with a stranger, firewalls do not provide full protection for the system (Rung-Ching , Kai-Fan and Chia-Fen ,2009).

So, it is needed to extend the network security capabilities by complementing with other tools or intrusion detection system (IDS is not a replacement for either a good antivirus program or firewall). In table 1.1, we present a comparison between IDS and firewall protection.

Table 1.1 (Comparison between IDS and Firewall (Wikipedia: The Free Encyclopedia, 2005)).

IDS	FIREWALL
Observing for intrusion that begins within a system	Look out for intrusion in order to stop them from taking place
Evaluate a suspected intrusion once it has takes place and signals an alarm	limits the access between networks in order to prevent intrusion Does not signal an attack from inside the network

(Kozushko, 2003) explained that Intrusion detection has grown to be the mainstream of information assurance. While firewalls do provide some protection, they do not provide full protection and still need to be developed and complimented by an intrusion detection system. The purpose of intrusion detection is to help computer systems deal and get ready for different types of attacks.

1.2 Problem Statement

The spread of the Internet and network everywhere raised the chance for using it as an intermediary transfer of information between client users, which decrease transfer information effort and expenses from one place to another compared to the conventional transfer way.

There are many problems associated with IDS. In this thesis, the following problems have been identified:-

1- How to protect information systems environment and keep it away from intruders by using Hopfield neural network and its variant.
2- How to increase the ability of variant Hopfield neural network to detect a new attack depending on clustering and classification methods.
3- How to estimate the performance of learning in the proposed model.

1.3 Objectives Of the Study

The aim of this research is to show the success of the proposed model in terms of different accurate measurement of attack detection, Detection Rate (DR), low False Alarm Rate (FAR), Precision Rate (PR), Recall Rate (RR) and to evaluate the results.

This will be achieved through the following objectives:-

1- Classification of the misuse of information.

2- Feature extraction on data codification

3- Designing a model based on an Enhanced Hopfield Neural Network model.

4- Application of the designed model in an appropriate environment data with specific parameters and comparing the estimated results with the others.

1.4 Significance of the Study.

This thesis develops and applies two types of IDS, Hopfield with K-means (HNKMIDS) and Hopfield with K-Nearest Neighbor (HNKNNIDS), and then it evaluates the performance of each model with the comparison between them.

1.5 Motivation

Searching for a model that secures the computer environment for trained agents can be a very difficult and time consuming task. The goal of this thesis is developing a system that can help workers detect misuse intrusion detection system in the environment.

Searching in an environment for a particular system requires big efforts and usually faces many problems. There are many intrusion detection systems that are usually used by individuals and organizations.

The main reason behind using misused methodology in this thesis is rare researches that use misused methodology to detect intruders.

1.6 Limitation of the Study.

As for many studies; there are some different challenges viewed in the intrusion detection systems. In this study, some limitations were faced. They can be summarized as follows:

1. Intrusion detection systems need a periodic update to the training set and profiles.
2. Using a static training data might become outdated and deficient for prediction.
3. The accuracy of classification for the data do not 100%.

1.7 Thesis Organization.

This thesis consists of six chapters organized as follows:

Chapter two: this chapter will focus on the related works in the field of intrusion detection using either neural networks or machine learning algorithms. The chapter also discusses the hybrid system models of supervised and unsupervised training algorithms that have been designed by other researches.

Chapter three: reviews intrusion detection approaches, presents an overview of intrusion detection system, gives an overview of attacks and learning methods, and finally discusses the related work**.** The difference between Host Intrusion Detection System architecture (HIDS) and Network Intrusion Detection System architecture (NIDS) will be discussed.

Chapter four: Artificial Neural Networks are the main subject of the thesis work; therefore, this chapter will discuss neural networks including the biological neural and artificial neural network model, advantage and architecture.

Chapter five: outlined research methodology used by this thesis. It also presents the proposed Enhanced model architecture and the software that is used for the evaluation of our model. It also describes the dataset used for experiments in this study, experiments environment and procedures, It also presents the evaluation measures and experimental results and finally gives a comparison with other studies' results.

Chapter six: the two proposed systems are implemented and tested (using the k-Nearest Neighbor with the enhanced Hopfield artificial neural network and using the K-means with the enhanced Hopfield artificial neural network). The results of using the two separated models are demonstrated in tables. Our results are also compared with other research results. We present conclusions and future recommendations.

Chapter Two

Literature Review and Related Work

Literature Review and Related Work.

2.1 Introduction

In this chapter, we will review some of the related work in the areas of intrusion detection and partially true data makes them attractive to be applied in intrusion detection.

Since there are many researchers from many different nationalities who have concentrated on the field of intrusion detection, the notion of intrusion detection was born in 1980 -with the publication of John Anderson (Anderson, 1980) - and it has been an active topic till now. Here, we will introduce some of the previous and the most recent researches in order to find the model that is more accurate and has better results of detection intruder rate. Then, we will compare our results with the previous researchers' result. Figure 2.1 shows the evolution of Intrusion Detection System (Paul, 2001).

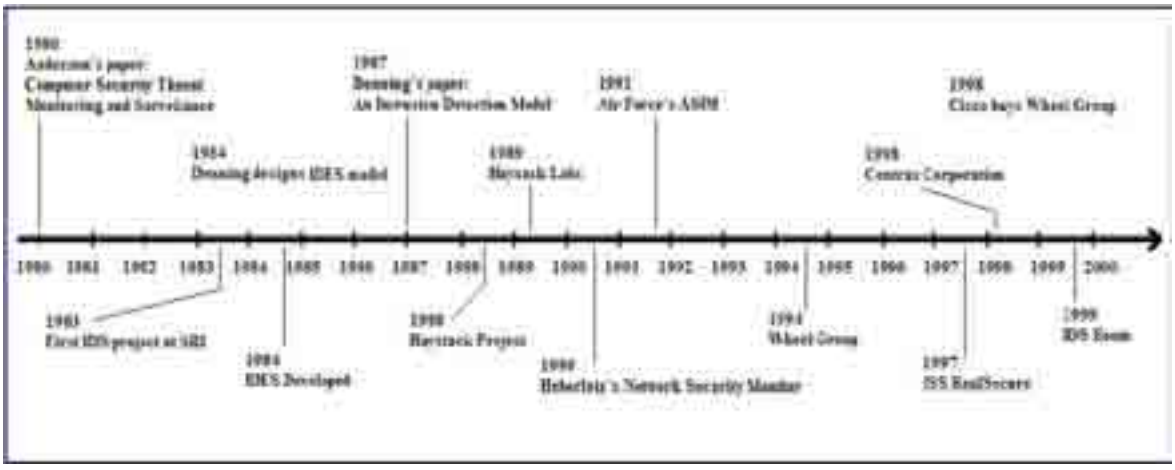


Figure 2.1 (Evolution of Intrusion Detection System (Paul, 2001).)

2.2 Intrusion Detection System

Intrusion detection has been the most popular topic as an effective countermeasure for various attacks. IDS are usually built to identify unauthorized behaviors of outside or inside intruders and to enforce the security policy of computer systems.

Scarfone and Mell (2007) in their paper, they presented Intrusion detection as a process of monitoring the events that occur in a computer system or network and analyzed them for signs of possible incidents, which are violations or imminent threats of violation of computer security using acceptable policies, or standard security practices.

Intrusion prevention is the process of performing intrusion detection and attempting to stop detecting possible incidents. Intrusion detection and prevention systems (IDPS) primarily focus on identifying possible incidents, logging information about them, attempting to stop them, and reporting them to security administrators.

In addition, organizations use IDPSs for other purposes, such as identifying problems with security policies, documenting existing threats, and deterring individuals from violating security policies. IDPSs have become a necessary addition to the security infrastructure of nearly every organization.

Kaxienko and Dorosz (2004) designed an overview of intrusion detection systems. They considered a type of detectable attacks, which is not associated to IDS, and different numeral terminology associated to IDS. (Kaxienko and Dorosz, 2004). They also introduced a short review about architecture of IDS.

2.3 Clustering and classification method

Brifcani & Issa (2011) presented a comparative study where three different classifiers were used; Data Mining Association Rules (DMARs), Decision Trees (DTs) and Artificial Neural Networks (ANNs). A Feed forward neural network was trained using backpropagation algorithm, and the type of DT was the Interactive Dichotomizer3 (ID3). Their experiments demonstrate that DMARs gave the worst results in terms of classifications, and their neural network training time took about 23.5 days while ID3 took 2 minutes. ID3 classification rate was 92.2%, which was the best result among the proposed methods. The main downstream of this method is the training time for ID3 because 2 minutes training is considered large in comparison to other systems, especially when using neural network properly; the training time is measured in seconds even for a large dataset.

Nieves (2009) used data clustering for anomaly detection in network intrusion detection system. The author used k-means algorithm to evaluate the performance of an unsupervised learning method for anomaly detection using KDD Cup 1999 network dataset. In this paper, the author converted the three symbol columns feature to binary format, and the continuous columns were normalized, so their maximum was one. Therefore, the number of feature columns expanded to 80 features instead of 41. The results of the evaluation confirm a good detection rate about 89% for 5 clusters while maintaining false rate of about 4.8%. This method has the advantage of using unsupervised method; therefore, the false positive rate was reasonably good, but on the other hand the system detection rate was not very high in comparison to other systems.

(Katos , 2007) spoted the light on statistical methodologies (cluster analysis, discriminant analysis, and Logic analysis) by using the same intrusion detection data for the examination. The research was based on a random sample of 1200 observations for 42 variables of the KDD-99 database, that contains 'normal' and 'bad' connections.

According to the Kappa statistics that make full use of all the information contained in a confusion matrix, the results indicate that Logic analysis is a more effective method than cluster or discriminant analysis in intrusion detection ; Logic analysis (K = 0.629) has been ranked first, with second discriminant analysis (K = 0.583), and third cluster analysis (K = 0.460).

Faraoun and Boukelif (2006) this paper presented a new technique to increase the learning capabilities and, it also reduced the computation intensity of a competitive learning multi-layered neural network using the K-means clustering algorithm. The proposed model used multi-layered network architecture with a back propagation learning mechanism.

The K-means algorithm was firstly applied to the training dataset to reduce the amount of samples to be presented to the neural network by automatically selecting an optimal set of samples. The obtained results demonstrate that the proposed technique performed exceptionally in terms of both accuracy and computation time when applied to the KDD99 dataset compared to a standard learning schema that used the full dataset. Table 2.1 summarizes the obtained performances results

Table 2.1 (Performances Results for the K-means Based Neural Network Approach (Faraoun and Boukelif , 2006))

Parameter	Value
Detection Rate	92%
False Alarm Rate	6.21%
Execution Run Time	28m 21s

2.4 IDS Based on Artficial Neural Network

ANNs are based on the neural structure of the human brain, which processes information by means of interaction between many neurons. In the last few years, there has been a constant increase in interest of neural network modeling in different fields of scientific materials. The basic unit in the ANNs is the neuron. The neurons are connected to each other with weight factor.

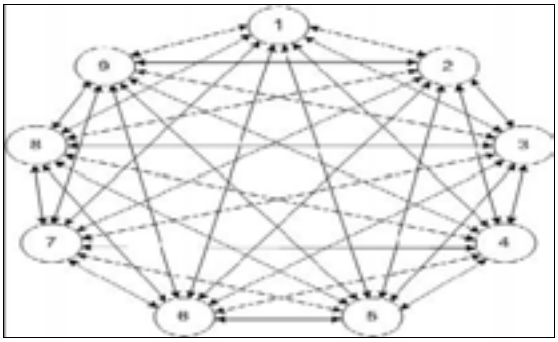


Figure 2.2 (Layout of a Single-layer 9-neuron Hopfield Network)

Lippmann (1987) in his paper the definition of Hopfield ANN is called Thermo Dynamic Models. Figure 2.2 consists of a single layer feedback neural network; there is no difference between input and output neurons. It is considered as the main integral part of day to day activities. The main application of Hopfield ANN is the storage, patterns recognition and classification problems with binary pattern vector (Hopfield,1982) .

If the input values are continuous, it must discretise, which means analog quantities must be converted to binary values. Artificial neural networks (ANNs) are networks of highly interconnected neural computing elements that have the ability to respond to input stimuli and to learn how to adapt to the environment.

2.5 IDS Based on Support Vector Machine

Rung-Ching , Kai-Fan and Chia-Fen (2009) the proposed IDS method which use Support Vector Machine(SVM) and based system on a Rough Set Theory (RST) decreases the space density of data effictively and reduces the number of features

from 41 to 29 that will be tested and manipulated to categorize the data (normal or attack). The experiments results compared between different methods.The study showed that RST and SVM schema can work together and improve the false positive rate and accuracy.

2.6 IDS based on Hybrid Methods

Islim (2012) designed an intrusion detection system based on human immune system. He presented a model for intrusion detection system that consisted of four components depending on innate/adaptive human immune system approaches and self/non-self theory of human immune system. He presented a model which was divided into two subsystems; the first one was an attack response system which was similar to innate human immune system, and the second was learning system which was similar to adaptive immune system. The learning system was the core of the model; it presented a hybrid approach of machine learning through hybridization between k-Means clustering algorithm and Naive Bayes as a classifier. The goal was to keep information systems environment safe against intrusions and attacks through applying human immune system mechanism and properties to intrusion detection system. Experimental results illustrated that our proposed model provided a higher detection rate in both DoS attacks and U2R attacks, which gave the power to our proposed hybrid model and increased the security of information systems, especially in the critical environments.

Al-Rashdan (2011) proposed an intelligent model using Hybrid Artificial Neural Networks, supervised and unsupervised learning capabilities to detect network intrusions from the KDDCup'99 dataset. She designed three cooperative phases by using an enhanced k-means clustering algorithm in Phase-1, a Hybrid Artificial Neural Network (Hopfield and Kohonen-SOM with Conscience Function) in Phase-2 and a Multi-Class Support Vector Machines in Phase-3. The Hybrid Neural Network Machine Learning Model achieved a detection rate of 92.5% and false positive rate of 3.5%. The main advantage of the proposed system was that the author used both supervised and unsupervised methods, therefore, minimizing the false positive rate. On the other hand, using both supervised and unsupervised should expect to have a higher detection rate than 92.5%.

Chapter Three

Intrusion Detection Systems

Intrusion Detection System

3.1 Introduction

Due to the increasing number of researchers from many different nations who have been concentrating on the field of intrusion detection, intrusion detection has been the most active and widely spread in many fields. John Anderson, was the one who shed light on the notion of intrusion detection in his paper which was published in 1980 (Anderson, 1980), and it has been the most active topic till now.

(Mukherjee, Heberlein, and Levitt,1994) mentioned that intrusion detection can be configured as the detection of outside illegal visitors “who are using a computer system without authorization” and inside intruders “who have acceptable access to the system but are abusing their privileges”.

3.2 Computer System Security

In this section we will give an overview of security types, such as the terms information, computer and network security means and the way to avoid the intruders (unauthorized) away from their goals such as get attention, gain some benefit or harming someone (Mukherjee, Heberlein, and Levitt,1994) .

Day by day, the terms of information security and assurance become more interrelated and share the common goals of protecting the confidentiality (or secrecy), integrity and availability of information against threats.

Intrusion detection systems are usually built in the companies to identify unauthorized behaviors of outside or inside intruders and to enforce the policy of computer systems security. IDS’s main role in telecommunication networks is to enforce the overall security of the network together with existing security measures such as firewalls and antivirus scanners products.

IDS’s place in the telecommunication networks depends on what it is supposed to monitor and protect. For example, IDS’s could be monitoring intrusions from either inside or outside the core network.

(Bishop, 2005) Computer System Security can be defined as the operation of protecting the main factors for any computer system security. Those factors are: confidentiality, integrity, and availability.

These three concepts are defined as follows:-

- ***Confidentiality*** (or secrecy):- means that computer related assets such as information is disclosed only according to policy; that is, only those who should have access to something will actually get that access.
- ***Integrity***: - means that information is not destroyed or corrupted through transferring and that the system performs correctly only by authorized parties and an authorized way.
- ***Availability***: - means that system services are available when they are needed and information is accessible to authorized parties at appropriate times.

3.2.1 Network Security

The collection of nodes and links perform network; that is, we can define the computer network as a collection of computers interconnected with each other by exchanging information by a single technology in order to secure the network and prevent it from any unauthorized access from both inside and outside the network (Tanenbaum, 2003). Network security is made up of the following:-

- Authentic users.
- A firewall accesses allow services and policies to the employee inside the network.
- A Network Intrusion Prevention System that monitors any illegal traffic.

3.2.2 Firewall

Firewall is a protective system which can be implemented in both hardware or software or a combination of both (Kurose & Ross, 2010). It lies between computer network and the internet in order to avoid illegal employ and right to use to the network. Firewalls play an important role on any network as they provide a protective barrier against most forms of attack coming from the outside world.

The job of a firewall is to carefully analyze entering data and exiting the network based on predefined rules for the system which are matched against network traffic. If the traffic is not in conflict with these rules, it is then allowed to pass through normally. Everything that is against these rules is dropped and defined as type of attack.

3.3 Intrusion Detection Systems Methods

There are two common methodologies that have been used to prevent snooper (attacker) from notifying the networks; intrusion and prevention (avoidance). IDS systems vary in manner from one to another. The differences between these systems are based on the kind of intrusion and how the system can be defined or the organization from snooper (attacker).

Intrusion detection systems can be classified into either host-based, network-based or a hybrid of the two according to the source of data they monitor, and on the area that will be used to process IDS on it.There are two general types of intrusion detection systems :

3.3.1 Host-based intrusion detection system (HIDS).

Host-based systems are resolved to prevent insiders' misuse, but cannot effectively prevent outsiders' misuse. It mainly runs on client or a single workstation or host to protect that one single object (host) (Pfleeger , 2003) .To do so, the host-based IDS should monitor system calls, log files , and other operations on the host in order to detect intrusions .

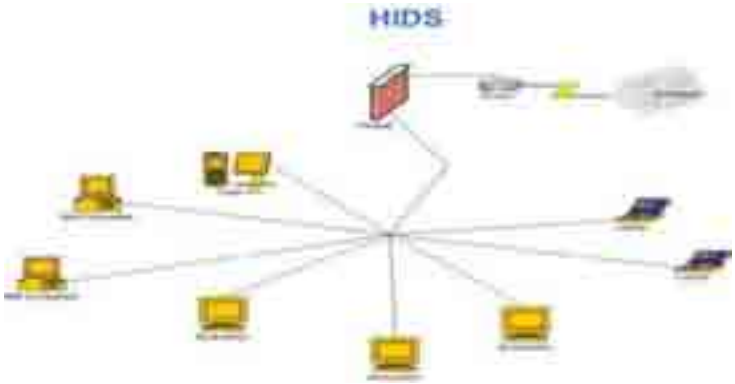


Figure 3.1 (HIDS Scenario (Pfleeger , 2003))

In figure 3.1, it does not matter where the machines are. Even if they are away from the network, they will still be protected at all times (Magalhaes, 2003). The machines which are colored with orange represent where The HIDS is installed.

3.3.2 A Network-based intrusion detection system (NIDS).

An IDS is resolved to prevent outsiders' misuse as "a stand-alone device attached to the network to monitor traffic throughout that network" (Pfleeger, 2003).

This type of IDS generally hooks into the network by connecting to a network hub or switching and typically does so at network borders.

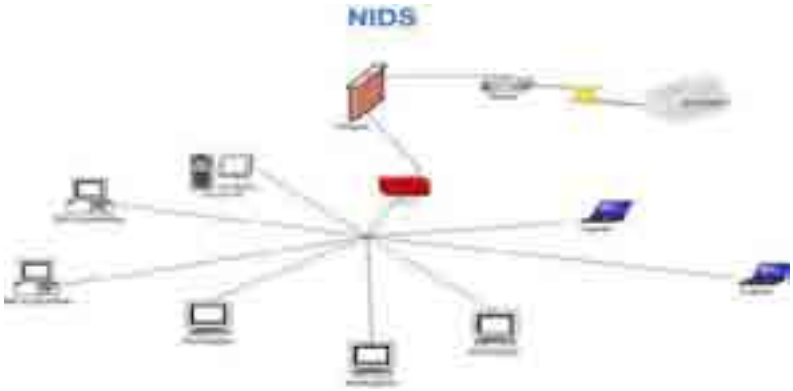


Figure 3.2 (NIDS Scenario (Pfleeger , 2003))

Figure 3.2 represents The NIDS scenario where the red device represents where the NIDS has been installed (Magalhaes, 2003).

3.3.3 Network –Based IDS versus Host-Based IDS

The difference between host-based and network-based intrusion detection is that HIDS is concerned with what occurs on the hosts themselves while NIDS deals with data transmitted from host to host. Today, a corporate network continues to evolve to mobile/wireless environments where computers are used outside the network.

It is essential to have NIDS in place to detect any malicious activity as a connection to the network is being made. The most common question IT infrastructure groups ask can be “what type of product is necessary for their environment” (Paul ,2001) .

A proper IDS implementation would be advantageous to fully integrate the network intrusion detection system because it would filter alters and notifications. In addition, it is also important to the host-based portion of the system, controlled from the same central location (Rebecca ,2006).

3.4 Intrusion Detection Techniques

An intrusion detection methodology is conventionally classified into three techniques which are represented in figure 3.3 (Bishop, 2005):-

1- Anomaly Detection Methodology *"Anomaly detection looks for unexpected behavior"*.
2- Misuse Detection Methodology: *"Misuse detection looks for sequences of events known to indicate attacks*".
3- Specification Detection Methodology*:"Specification-based detection looks for actions outside the specifications of key programs"*.

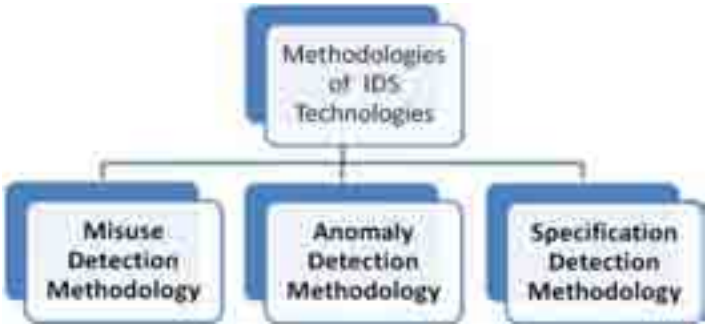


Figure 3.3 (Methodologies of IDS Technologies)

A multiple intrusion detection technique, can be used either individually or incorporation to afford more broad and precise detection, and they will be discussed separately, and a comparison between them will be made according to the advantages and disadvantages of each one of them (Scarfone and Mell , 2007) . The primary classes of detection methods are as follows:

3.4.1 Anomaly Detection Methodology: In 2005, Dorosz and Kazienko defined anomaly detection as one of the methods of intrusion detection as a warping behavior of the system; any conduct that ominously a deviate from the routine pattern is considered intrusive (Dorosz and Kazienko , 2005). Figure 3.4 presents the processes flow for typical anomaly intrusion detection.

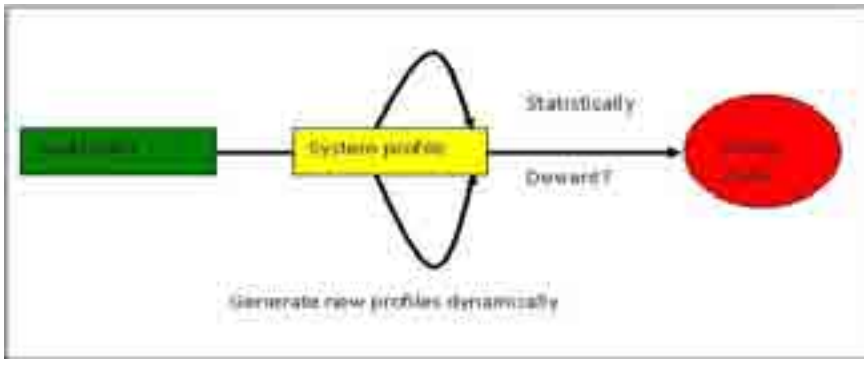


Figure 3.4 (A Typical Anomaly Intrusion Detection System (Kurdi ,2011))

Advantages:

- possibility of detection of novel attacks as intrusions,
- anomalies are recognized without getting inside their causes and characteristics,
- less dependence of IDSs on operating environment and
- ability to detect abuse of user privileges.

Disadvantages

- A substantial false alarm rate. System usage is not monitored during the profile construction and training phases. Hence, all user activities skipped during these phases will be illegitimate.
- User behaviors can vary with time; thereby requiring a constant update of the normal behavior profile database. This may imply the need to close the system from time to time and may also be associated with greater false alarm rates.
- The necessity of training the system for changing behavior makes a system away from the anomaly detected during the training phase (false negative) (Dorosz and Kazienko,2005).

3.4.2 Misuse Detection Methodology: (Dorosz and Kazienko,2005) defined Misuse Detection as depending on searching and comparing for ambiguous pattern or signature of system snoops, and any pattern that corresponds with a known attack is considered as an intrusive. In figure 3.5, we present the processes flow for typical misuse intrusion detection.

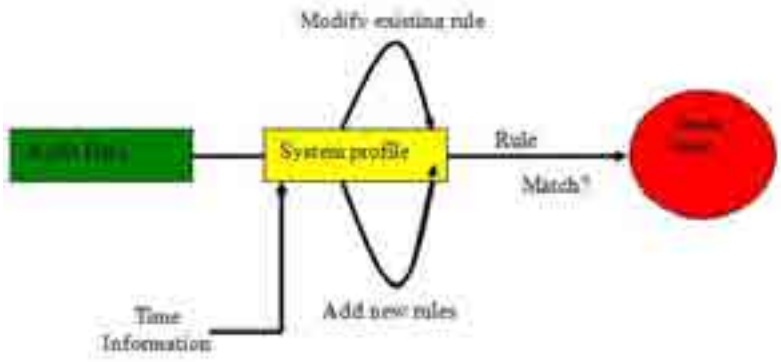


Figure 3.5 (A Typical Misuse Intrusion Detection System (Kurdi,2011))

Advantages :

- Very low false alarm rate.
- Simple algorithms.
- Easy creation of attack signature databases.
- Easy implementation.
- Typically minimal system resource usage.

Disadvantages:

- Difficulties in updating information on new types of attacks.
- Inherently unable to detect unknown novel attacks.
- Maintenance of an IDS is necessarily connected with analyzing and patching of security holes which is a time-consuming process.
- Attack knowledge is operating environment-dependant, so misbehavior signature-based intrusion detection systems must be configured in strict compliance with the operating system.
- Difficulty handling internal attacks.

3.4.3 Specification Detection Methodology (Stateful Protocol Analysis):

One of the methods of intrusion detection which occurrs when the system is unable to take an action or when the system has entered unable state is considered intrusive. In addition, the Specification Detection Method is comparatively new in its development and use (Bishop, 2005).

Advantages:

- Formal stating of what should happen; i.e., intrusions using unknown attacks will be detected.

Disadvantages:

- Relatively new model.
- Effort to identify programs that could cause a security threat.

3.5 IDS Attack Types Overview.

This part is an overview of the different types of attack which can be classified into four groups, and also introduces several names of attack that belongs to each one of them. (Das, 2000).

3.5.1 Denial of Service (DOS).

This type of attack can start from a single host and trying to block access. The authorized access (user) to services offered by a single host or network can be done by overloading services or by crashing a single host or network and denying users access to a machine.

The following attacks are examples of Denial of Service attacks:

\Back", \Land", \SYN Flood" (Neptune),\Ping of Death"' (POD), and \Smurf",\mail bomb",\UDP storm", etc.

3.5.2 Remote to User (R2U). :

Attacks of this group, Remote to User, aim at achieving access to a users' account from another host or network ;a user sends packets to a machine –does not have an access on it- over the internet in order to expose the machines vulnerabilities and impose benefit which a local user would have on the computer.

The following attacks are examples of Remote to User:

\Xnsnoop",\Dictionary", \Ftpwrite", \Guest",\Xlock" ,\Imap" and \Phf", etc.

3.5.3 User to Root (U2R):

Having normal user privileges, User to Root, aim at obtaining root accesses (system administrator privileges). Intruders firstly try to get normal user privileges before they try to exploit different security flaws in order to gain root access.

The following attacks are examples of User to Root attacks:

\Xterm",\Eject", \Ffb", \Loadmodule" and \Perl".

3.5.4 Probing:

To gain valuable information about a host or network, the target is scanned in order to determine weaknesses. This information is used to find security flaws and services which are running on the target.

Probing is not considered as an attack, but most sophisticated attacks use scanning as a first step.

The following attacks are examples of \Probing/Scanning": \Nmap", \Satan", \Saint",\Mscan", \Ipsweep" and \Portsweep". The first three tools \Nmap", \Satan" and \Saint" are sophisticated security scanners, able to scan the target for security flaws. These tools are also used by system administrators. \Ipsweep" and \Portsweep" are simpler variants.

3.6 Comparison between IDS and firewalls

Although they both relate to network security, IDS differs from a firewall in that a firewall looks outside for intrusions in order to prevent them from occurring. Firewalls limit access between networks to avoid intrusion and do not signal an attack from inside the network. Once suspected intrusion has taken place, IDS evaluates and signals an alarm. IDS also watch for attacks that originate from within a system. This is traditionally achieved by examining network communications, identifying heuristics and patterns (often known as signatures) of common computer attacks, and taking action to alert operators.

3.7 Clustering

Clustering is a process of handling a combination of similar data into clusters so that data within a cluster; a specific related type; which has a high match in comparison to one another, and the dissimilar data in other clusters. Clustering involves dividing a set of objects into a specified number of clusters. The incentive behind clustering a set of data is to find inherent structure in data and to expose this structure as a set of groups. Two main types of clustering algorithms (Elkan , 2011):-

- ***Hierarchical algorithms***: Create a hierarchical decomposition of the set of objects using some criterion. This requires a measure of similarity between groups of data points. The main idea behind hierarchical clustering is to build a binary tree of the data that successively merges similar groups of points. Visualizing this tree provides a useful summary of the data.

- ***Partitioning (Nonhierarchical) algorithms***: Construct various partitions and then evaluate them by some criterion. The K-means algorithm is an example of the partitioning based, nonhierarchical clustering methods.

Partitioning algorithms require the number of clusters k, an initial assignment of data to clusters and a distance measure between data. Partitioning algorithms relocate instances by moving them from one cluster to another, starting from an initial partitioning.

In this thesis, we will use the partitioning clustering algorithms (K-means and k-Nearest Neighbor clustering), hence it can estimate the number of clusters (K), where K represents the number of attack types; initial classes or subclasses of attack.

Chapter Four

Artificial Neural Networks

Artificial Neural Networks

4.1 Historical Overview of Artificial Neural Network

(McCulloch and Pitts ,1943) represented in their paper the valuable idea about research and study in AI field as an important field in the age of information. The history of the neural network field is often considered to have begun with their paper. In 1949, Donald Hebb wrote a book entitled "*The Organization of behavior"* (Hebb , 1949) where he described a learning paradigm that now bears his name, with a lot of different applications which include Expert System, ES, and artificial neural technology.

There is a great development in the field of AI software which is used to simulate the human reasoning. However, the method used in processing AI is sequential with the use of knowledge representation and questions. In neural computing, we use parallel processing, and such technology offers a great speed and can store a large amount of information.

In 1987, the first bigger conference specialized on neural networks in modern times, the IEEE International Conference on Neural Networks with 1700 participants, was held in San Diego, and the International Neural Network Society (INNS) has been established.

One year later the INNS began to publish its journal *Neural Networks*, followed by *Neural Computation* (1989), *IEEE Transactions* on Neural Networks (1990) and many others. Beginning in 1987, many prestigious universities founded new research institutes and educational programs in neurocomputing. This form represents a new class of computers, optimized for running neural networks. Here, the information-processing mechanisms are designed for implementing the systems of differential equations associated with neural networks (Marilyn and Illingworth, 1991) .

This trend has continued up to the present since there are dozens of specialized conferences, journals and projects based on neural networks. It turns out that a wide range of research and investment in neurocomputing may not correspond to the quality of achieved results. It is hoped that in the near future the vitality of neural network field will be proven.

4.2 What Is a Neural Network?

There are many definitions of neural networks. The simplest one is a technology for processing information, and it simulates the human brain modeling and nervous system which works quite differently than conventional computing. An artificial neural network consists of a large number (approximately 10^{11}) of a highly interconnected collection of processing elements (approximately 10^{4} connections per element) called neurons that are transforming a set of inputs to a set of desired outputs.

The result of the transformation is determined by two main components the characteristics of the elements and the weights associated with the interconnections among them. The network is able to adapt to the desired outputs by modifying the connections between the nodes (Kevin, Rhonda, and Jonathan ,1990).

Neural networks have proven themselves as proficient classifiers and are particularly well suited for addressing non-linear problems. It conduct an analysis of the information and introduces a probability estimation matches between the data and characteristics which it has been trained to recognize. While the probability of a match determined by a neural network can be 100%, the accuracy of its decisions relies totally on the experience the system gains in analyzing examples of the stated problem.

The neural networks achieve the experience initially by training the system to correctly identify pre-selected examples of the problem. The response of the neural network is reviewed and the configuration of the system is refined until the neural network's analysis of the training data reaches an agreeable level. In addition to the initial training period, the neural networks also achieve experience over time as they conduct analyses on data related to the problem (Dan, 1993).

4.2.1 Biological Neural Networks

In 1943, McCulloc and Pitts introduced a set of simplified neurons after which artificial neural network was born. These neurons were represented as models of biological networks into conceptual components for circuits that could perform computational tasks. The basic model of the artificial neuron is founded upon the functionality of the biological neuron.

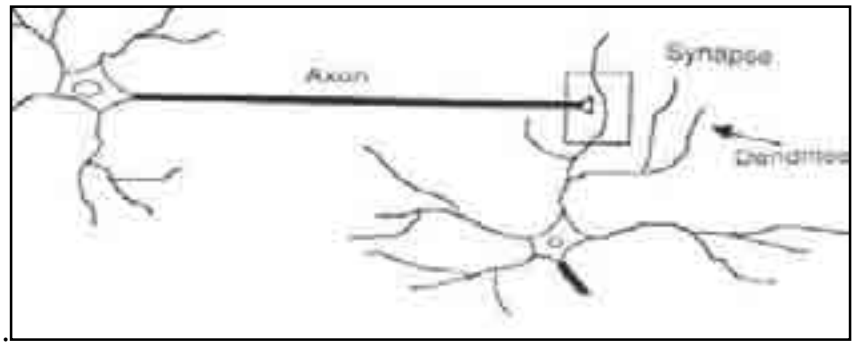


Figure 4.1 (a Biological Neuron and Neuron Network)

The above figure, 4.1, represents a portion of a network which consists of two interconnected cells that have the three principal components of a typical biological neural; these are:

1. The cell body (soma) itself includes a nucleus (the heart of the cell).
2. Dendrites, which look like a tree structure that provides input and carry signals into the cell body.
3. The axon sends output signals away from cell body to other neurons.

These axon terminals merge with the dendrites of the next cell to form Biological neural network. A synapse is the point of contact between an axon of one cell and a dendrite of other cells. A synapse is able to increase or decrease the strength of the connection from neuron to neuron and cause excitation or inhibition of a subsequence neuron. Signals can be transmitted unchanged, or they can be altered by synapses where information is stored. The information processing abilities of biological neural systems must follow from highly parallel processes operating on representations that are distributed over many neurons. One motivation for ANN is to capture this kind of highly parallel computation based on distributed representations as shown in figure 4.2.

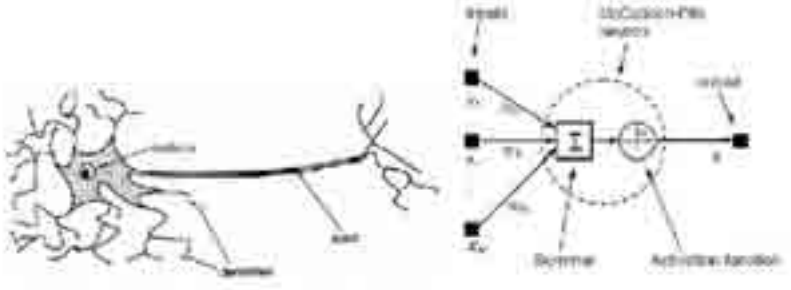


Figure 4.2 (a biological neuron and its model (McCulloch and Pitts, 1943))

4.2.2 Artificial Neural Network Component

The purpose of the artificial neural networks is to simulate only the most basic component of this complicated, versatile, and powerful organism. They do it in a primitive way. But, for the software engineer's aim who is trying to solve problems, neural computing was never about replicating human brains. It is about machines and a new way to solve problems (Reingold and Nightingale, 1999).

Since there are different types of ANN, the component of all ANN is the same component Processing unit, Combination function and transferring unit.

4.2.2.1 Processing unit

Each unit performs a relatively simple two jobs:

- The first task is to receive input from neighbors or external sources and use this to compute an output signal which is propagated to propagate units.

- The second task is the adjustment of the weight. The system is inherently parallel in the sense that many units can carry out their computations at the same time (kukielka and kotulski,2008). The architecture of this ANN consists of three types of layers:

    1. Input Layer; the responsibility of this layer is receiving data from outside the ANN.

    2. Hidden Layer; signals from the input and output remain within the neural network in this layer.

    3. Output Layer; sending data out of the neural network is the main function of this layer.

    During operation, units can be updated either

    -Synchronously: all units update their activation simultaneously

    - Asynchronously: each unit has probability of updating its activation at a time *t*, and usually only one unit will be able to do this at a time (Muthukkuumarasamy and Birkely,2004).

4.2.2.2 Combination function

Each non-input unit in an ANN combines values that are fed into it via synaptic connections from other units, producing a single value called net input. The function that combines these values is called Combination function, which is defined by a certain propagation rule. In most neural networks, we assume that each unit provides an additive contribution to the input of the unit with which it is connected. The total input to unit *j* is simply the weighted sum of the separate outputs from the connected units plus the threshold or bias term *Θj*.

4.2.2.3 Activation function

Units in ANN have a rule for transforming their input value to an output value that will be transmitted to other units or for presenting to the environment the end result of computation. This rule is known as an Activation function, and the output value is referred to as the activation for the unit. In figure 4.3 a diagram of a neuron's operation, the activation may be: a real number, that is restricted to some interval such as [0,1] , or a discrete number such as {0,1} or {+1,-1}. The value passed to the activation function is the net combined input to a unit (Naoum, 2011).

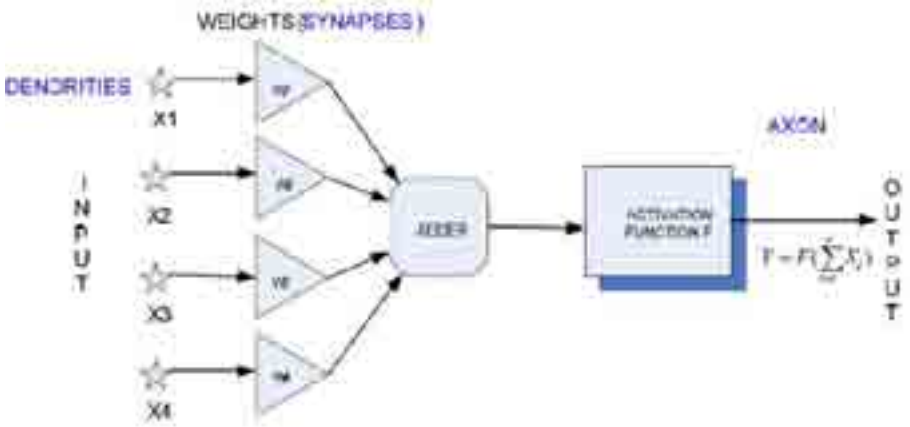


Figure 4.3 (artificial neuron's schema (Naoum, 2011))

(Naoum, 2011) mentioned that, as shown in figure 4.3, various inputs to the network are represented by the mathematical symbol, x_n. Each of these inputs is multiplied by a connection weight. These weights are represented by w_n. In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then neural output is provided.

(Reingold and Nightingale, 1999) considered that a neuron may sum its inputs, or average them, or something entirely more complicated. Each of these behaviors can be represented mathematically, and that representation is called the transfer function. Here, the most commonly activation functions are used and a brief overview of each one is given:

- **Identity function**: it assigns every real number x to the same real number x; the activation (signal sent on to other units) is the same as the net input is described by the following equation:

$$f(x) = x \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 4.1$$

- **Binary step 'threshold" transfer function** is a function which is like that used by the original perceptron. A binary threshold function will limit the activation to 1 or 0 depending on some threshold Θ. The output is a certain value 0 or 1.

$$f(x) = \begin{cases} 1 & x > 1 \\ x & -1 < x < 1 \\ -1 & x < -1 \end{cases} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 4.2$$

- **Sigmoid (logical) activation function**: is a commonly used function. The output from a sigmoid function falls in a continuous range from 0 to 1 but not linearly as the input change figure 4.4.

$$f(x) = \frac{1}{1 + e^{-ax}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 4.3$$

Where a: is a real number

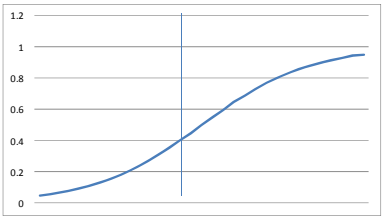


Figure 4.4 (Sigmoid Functions)

- **Hyperbolic Tangent Sigmoid Transfer Function** (MathWorks,2012)

$$f(x)=\frac{2}{1+e^{-2x}}-1 \qquad \text{4.4}$$

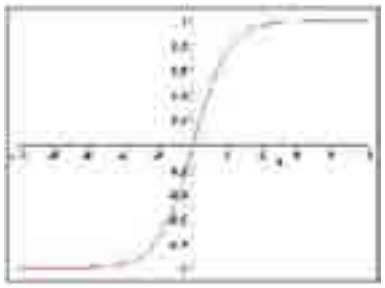


Figure 4.5 (Hyperbolic Tangent Sigmoid (Willamette University, n.d.))

In 2011, Naoum mentioned that the output of this function changes continuously non-linearly as the input changes. The sigmoid function is bounded and differentiable real function and has positive derivative, and it has lower limit bound (0 or -1) and upper limit bound (+1) ,as shown in figure 4.5 above (Naoum, 2011).

4.3 Artificial Neural Network types

There are several types of neural network. Every month, new type or at least new modifications of old types are generated so that nobody knows exactly how many types there are. ANN can be classified according to:

1. Topology

    a. Feed Forward: in this type of ANN, a quickly response to an input is usually produced. The connection between units does not form cycles or loops. Training is usually easy i.e. the output of any layer does not affect that same layer.

    b. Feed Back or Recurrent: there are cycles (loops) connections between units. An input is presented each time. The ANN must iterate for a

potentially long time before it produces a response. Training is usually difficult. We will consider this topology type.

2. kind of data they accept

   a. Categorical Variable: may take symbolic values such as colors, shapes, etc….a finite number of possible values, and there are usually more cases falling into each category. It must be encoded into numeric format before passing through the network.

   b. Quantitative Variable: it represents numerical measurements of some attribute, such as weight, which reflects analogous relations among our object attribute. We will consider this type.

3. Learning algorithm

   a. Supervised: the correct result that includes target values and desired output are known to the ANN during training so that it can adjust its weight to try much output to the target values. After training, the ANN is tested by giving it input values and check how close the output to the correct target values. Supervised methods fall into two sub-varieties:

      - Auto-associative: the input and the target are the same.
      - Hetero-associative :the input and the target are different.

   b. Unsupervised: the ANN is not provided with the correct results during training. It usually performs some kind of data compression such as clustering.

4.4 The Learning Algorithms in Neural Network

An important property of ANN is their ability to learn from input data with or without a teacher. Learning is a process by which the free parameters of ANN are adapted through a process of simulation by the environment in which the network is embedded (Gupta, 2006).

The type of learning in ANN is determined by the manner in which the parameter-changes following the way learning is performed; we can distinguish two major categories of ANN (Naoum ,2011):

1. **Fixed Networks**: in which the weights cannot be changed that is $\frac{dw}{dt} = 0$, where $\boldsymbol{w}$ is the weight vector. In such a network, $\boldsymbol{w}$ is fixed from the beginning and according to the problem we want to solve.
2. **Adaptive Network**: in which the weight can be changed; dw / dt ≠ 0. The type of learning is determined by the manner in which the parameter changes take place (Bridges and Vaughn ,2000).

Naoum (2011) declared the rules that can be used in the learning processes of ANN and classified them into five numbers of basic rules. These rules are:

1. Error – Correction learning rule (Delta Rule)

The goal is to minimize the cost to correct the errors. This leads to the delta rule, which is stated as the adjustment made to a synaptic weight of a neuron. It is proportional to the product of the error signal and the input signal of the synapse in question, as shown in figure 4.6.

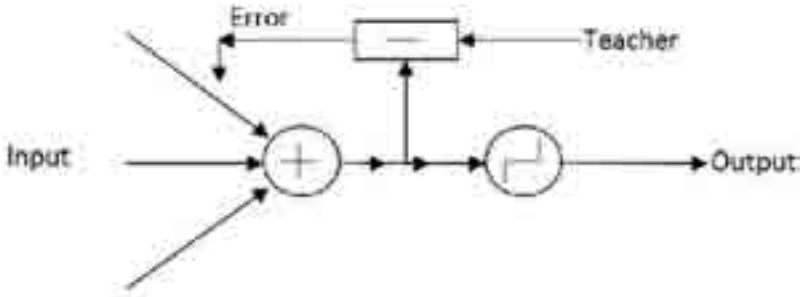


Figure 4.6 (Delta Rule (Learning Process))

2. Hebbian Learning rule

This is the oldest and most famous of all learning rules. Hebbian learning rule comes from Hebb's postulation that if two neurons were very active at the same time, which is illustrated by the high values of both its output and one of its inputs, the strength of the connection between the two neurons will grow or increase.

3. Boltzmann Learning rule

The neurons consist of a recurrent structure, and they operate in a binary manner. The machine is characterized by an energy function *E*. Machine operates by choosing a neuron at random then flipping the states of neuron k from x_k state to $-x_k$ at some temperature with probability $P(x_k \rightarrow -x_k)$

4. Competitive learning rule:

Naoum (2011) declared the Competitive learning rule as a process in which output layer neurons compete among themselves to acquire the ability to fire in response to give input patterns. A winner-take-all CLN (Competitive Learning Network) consists of an input layer and a competition, or output layer (e.g. kohonen network).

5. Memory-based learning rule:

All algorithms in this category involve two essential ingredients:

- Criterion used for defining the local neighborhood of the test vector ***x***.
- Learning rule applied to the training examples in the local neighborhood of the vector ***x***.

All learning methods used for adaptive neural networks can be classified into two major categories, and they will be discussed in the following subsections:

4.4.1 Supervised Learning.

It is also called classifier because it aims at building a predictive model (classifier) to classify the incoming patterns. This classifier should be trained with labeled patterns, so it can be able to classify the new unlabeled pattern later (MacQueen, 1967).

Naoum (2011) mentioned about this type of learning that an external teacher is required so that each output unit is told what its desired response to input signals must be. During the learning process, global information may be required. Paradigms of supervised learning include Error–correction learning or reinforcement learning and stochastic learning.

An important issue concerning supervised learning is the problem of error convergence, which is minimization of error ***E***, which is defined as the norm of the difference between the desired responses and computed (actual response of the network) unit values.

$$E = \|Z - Y\|_m \xrightarrow{converge} 0, m = 1,2,\infty \quad \text{..... 4.5}$$

Where Z: desired task (model)

Y: computed model from ANN

Our task is to determine a proper set of weights which minimizes the error *E* .

In supervised learning process, the adjustment of weights is done under the supervision of the teacher; that is, precise information about the desired or correct network output is available from a teacher when given a specific input pattern. In our thesis, we made use of the supervised learning, as shown in figure 4.7.

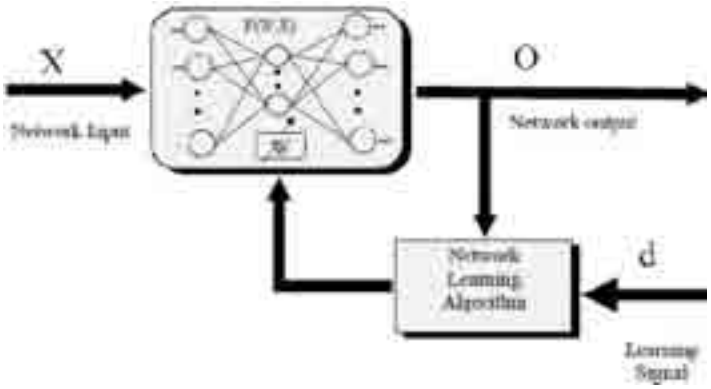


Figure 4.7 (Supervised Learning Rule Diagram (MacQueen, 1967))

4.4.2 Unsupervised Learning.

It is also called data clustering; because it is the separation of a set of objects into groups; each group consists of similar objects that are not similar to objects in other groups (Dunham, 2003). K-means is one of the best simplest clustering techniques to partition (*n)* instances into *(k)* clusters in which each instance belongs to the cluster with the nearest mean (MacQueen, 1967).

Naoum (2011) identified that it's also called self-organization learning, where the network is not given any external indication as to what correct response should be nor whether the generated responses are right or wrong. It is based upon only local information. It is simply exposed to the various input-output pairs, and it is learn by the environment; that is by detecting regularities in the structure of input patterns, as shown in figure 4.8.

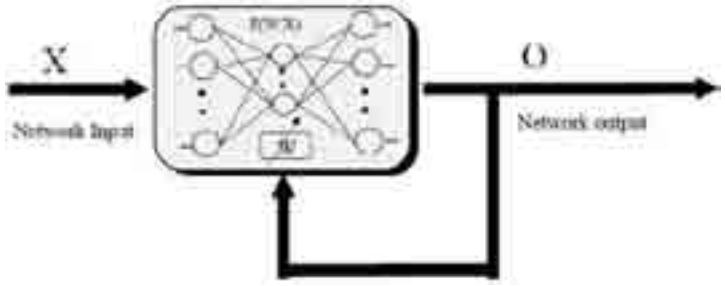


Figure 4.8 (Unsupervised Learning Rule Diagram (MacQueen, 1967))

4.4.3 Reinforcement learning:

It is somewhere between supervised learning and unsupervised learning. In this type of learning, a random search component is necessary since there is no information on what the right output should be, the system receives a feedback that tells the system whether its output response is right or wrong, but no information on what the right output should be is provided. The system must employ some random search strategy so that the space of plausible and rational choices is searched until a correct answer is found.

4.4.4 Semi–Supervised Learning :

It is a combination of supervised and unsupervised learning approach. This approach uses unsupervised learning technique to learn the structure of data, making it easier to identify the most interesting examples in training set. This enables a supervised learning technique to gain better performance with fewer labeled examples.

4.4.5 Hybrid Learning:

It is a combination of supervised learning, unsupervised learning, reinforcement learning and Semi–Supervised Learning.

.

Chapter Five
An Enhanced Hopfield Neural Network

An Enhanced Hopfield Neural Network

5.1 INTRODUCTION

Methodology used in this thesis is based on enhancement of Hopfield neural network with K-means or k-nearest neighbor for Misused IDS (MIDS), using MATLAB tools. The proposed Hybrid system consists of three phases to accomplish the research goal; each phase supplies the next one with a proper input to produce an efficient IDS model and feeds the other phase with the needed data in an appropriate format.

The data set which will be used in this experiment for training and testing is the data that originates from MIT's Lincoln lab the **K**nowledge –**D**iscovery and **D**ata mining, KDD cup 99 (Liu ,Florez and bridges ,2002).

5.2 The Enhanced Hopfield Neural Network Architecture

The Proposed Model Architecture consists of three phases, figure 5.1**.** Each phase sequentially co-operates and feeds the next phase with the needed data in an efficient format.

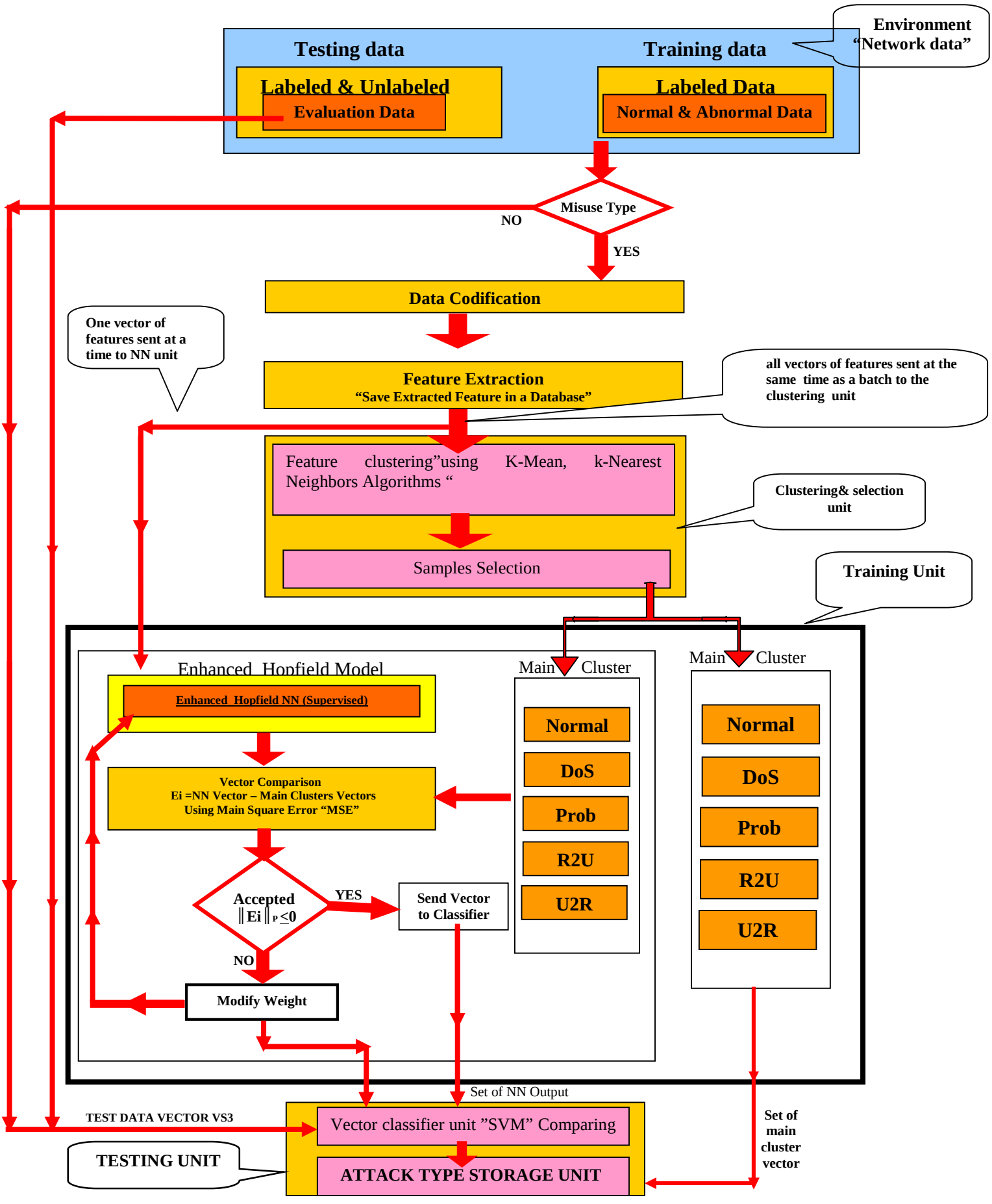


Figure 5.1 (The Main Processes of the Proposed Model)

5.2.1 PHASE 1:-

It is considered as the starting point for our project which is split into four co-operated units as we show in figure 5.2.

1- Environment Unit; that begins with capturing the data.

2- Data codification unit (Data Pre-processing Unit).

3- Feature Clustering Unit and

4- Clustering and Selection Unit.

The data starts flowing through this phase. In the next few pages we will consider the specific job in detail for each unit.

The first unit /Environment unit/ is considered the first unit which contains the collected data record from KDD CUP'99. It is the most widely used data set for the misused detection algorithms. It will be divided into two subsets, training subset and testing subset, and it's record categories labeled as normal or fall into one of four main categories of the attacks were simulated (Lee and Sheu ,1991):

- DoS (Denial of Service) – An attacker tries to prevent legitimate users from using a service, e.g. TCP SYN Flood, Smurf, etc.
- Probe – An attacker tries to collect information about the target host. For example: scanning victims in order to get knowledge of available services, operating system version etc.
- U2R (User to Root) – An attacker has a local account on the victim host and tries to gain root privileges.
- R2L (Remote to Local) – An attacker does not have a local account on a victim host and tries to obtain it.

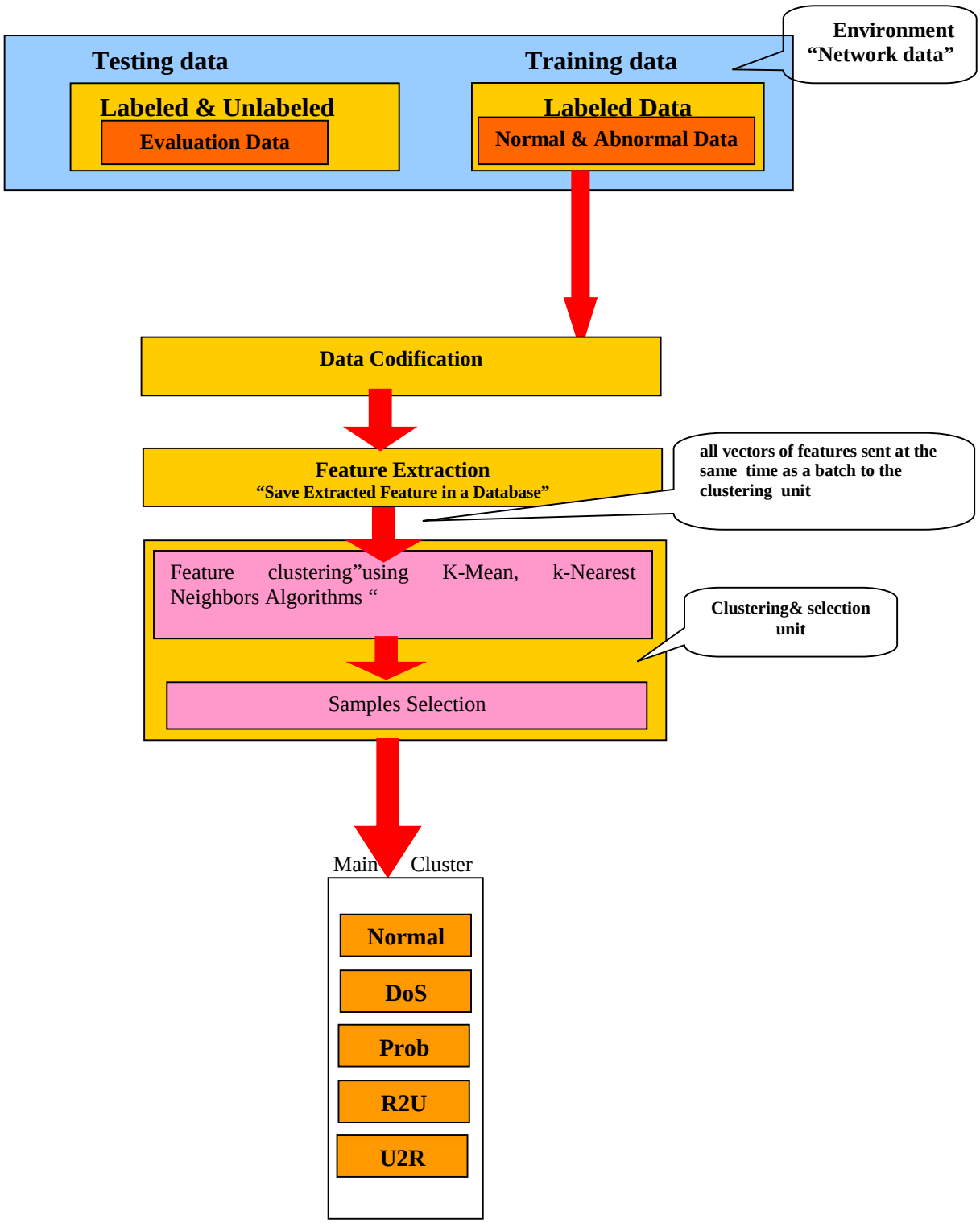


Figure 5.2 (Phase 1: Environment, codification, extraction, clustering and selection)

There are 41 features and each feature is considered in columns and its type is either symbolic or continuous. Each column and its type are represented in the following table, 5.1.

Table 5.1 (KDD Cup '99 Feature Columns Name and Type(KDD'99,1999))

	Feature Name	Feature Type		Feature Name	Feature Type
1	Duration	Continuous.	22	is_guest_login	Discrete.
2	protocol_type	Symbolic	23	count	Continuous
3	Service	Symbolic	24	srv_count	Continuous
4	Flag	Symbolic	25	serror_rate	Continuous
5	src_bytes	Continuous	26	srv_serror_rate	Continuous
6	dst_bytes	Continuous	27	rerror_rate	Continuous
7	Land	Discrete	28	srv_rerror_rate	Continuous
8	wrong_fragment	Continuous	29	same_srv_rate	Continuous
9	Urgent	Continuous	30	diff_srv_rate	Continuous
10	Hot	Continuous	31	srv_diff_host_rate	Continuous
11	num_failed_logins	Continuous	32	dst_host_count	Continuous
12	logged_in	Discrete	33	dst_host_srv_count	Continuous
13	num_compromised	Continuous	34	dst_host_same_srv_rate	Continuous
14	root_shell	Continuous.	35	dst_host_diff_srv_rate	Continuous
15	su_attempted	Continuous	36	dst_host_same_src_port_rate	Continuous
16	num_root	Continuous	37	dst_host_srv_diff_host_rate	Continuous
17	num_file_creations	Continuous	38	dst_host_serror_rate	Continuous
18	num_shells	Continuous	39	dst_host_srv_serror_rate	Continuous
19	num_access_files	Continuous	40	dst_host_rerror_rate	Continuous
20	num_outbound_cmds	Continuous	41	dst_host_srv_rerror_rate	Continuous
21	is_host_login	Discrete		Lable	Symbolic

5.2.1.1 Environment Unit begins by clustering, classification and selection (Chen, Cheng and Hsieh, 2009):-

- Clustering aims at extracting the features of input data in order to reduce the dimension of features space.
- Classifications will be carried out by using K-means or K-Nearest Neighbors clustering algorithms.
- K-means or K-Nearest Neighbors select the most representative samples for each attack type.

By the end of this phase, the preparing data is converted to an acceptable format to the next phase.

5.2.1.2 Data codification unit / Data Pre-processing Unit:

The raw data from the above unit will be converted from its current unacceptable format such as symbolic into an accepted numeric ANN format, and then the coded data will be sent to the clustering unit. To fulfill our demand, this unit

consists of two sequential steps: - Mapping symbolic-valued and implemented scaling.

1. **Mapping symbolic-valued** attributes such as protocol, service, flag and label to numeric-valued attributes. This step should be applied for the testing data set as done in the training data set ,the following tables (table 5.2, table 5.3, table 5.4, table 5.5 and table 5.6) demonstrate each symbolic columns and its numeric transformation

Table 5.2 (Protocol Column B Feature Transformation Table)

Protocol type	No
TCP	1
ICMP	2
UDP	3

Table 5.3 (Flag Column D Feature Transformation Table)

No	Flag	No	Flag
1	OTH	**7**	S1
2	REJ	**8**	S2
3	RSTO	**9**	S3
4	RSTOS0	**10**	SF
5	RSTR	**11**	SH
6	S0		

Table 5.4 (Flag Column C Feature Transformation Table)

No	Flag	No	Flag	No	Flag
1	Auth	**23**	Private	**45**	klogin
2	Ctf	**24**	remote_job	**46**	kshell
3	Daytime	**25**	Rje	**47**	ldap
4	domain_u	**26**	Smtp	**48**	netbios_dgm
5	eco_i	**27**	Ssh	**49**	netbios_ns
6	ecr_i	**28**	Sunrpc	**50**	netbios_ssn
7	Finger	**29**	Systat	**51**	netstat
8	ftp	**30**	telnet	**52**	nnsp
9	ftp_data	**31**	Time	**53**	pop_2
10	Gopher	**32**	Uucp	**54**	printer
11	Hostnames	**33**	Vmnet	**55**	red_i
12	http	**34**	Bgp	**56**	shell
13	imap4	**35**	Courier	**57**	sql_net
14	Link	**36**	csnet_ns	**58**	supdup
15	Login	**37**	Discard	**59**	tftp_u
16	Mtp	**38**	Domain	**60**	tim_i
17	Name	**39**	Echo	**61**	urh_i
18	nntp	**40**	Efs	**62**	urp_i
19	ntp_u	**41**	Exec	**63**	uucp_path
20	Other	**42**	http_443	**64**	whois
21	pm_dump	**43**	IRC	**65**	X11
22	pop_3	**44**	iso_tsap	**66**	Z39_50

Table 5.5 (Sub Attack cluster into Main Attack type)

	Attack Name	Attack Cluster		Attack Name	Attack Cluster
1	Normal	Normal	14	Back	DoS
2	Ipsweep	Prob	15	Land	
3	Nmap		16	Neptune	
4	Portsweep		17	Pod	
5	Satan		18	Smurf	
6	ftp_write		19	Teardrop	
7	guess_passwd	R2L			
8	Imap		20	buffer_overflow	U2R
9	Multihop		21	Loadmodule	
10	Phf		22	Rootkit	
11	Spy		23	Perl	
12	Warezclient				
13	Warezmaster				

Table 5.6 (Label Transformation Table)

Label	Column1	Column2	Coulmn3	Column4	Column5
Normal	**1**	**0**	**0**	**0**	**0**
DoS	**0**	**1**	**0**	**0**	**0**
U2R	**0**	**0**	**1**	**0**	**0**
R2L	**0**	**0**	**0**	**1**	**0**
Prob.	**0**	**0**	**0**	**0**	**1**

The following example illustrates the KDD Cup'99 rows and the transformation of symbolic text word in the dataset according to the last tables present in figure 5.3 and figure 5.4:-

0, tcp, ftp_data, SF, 491, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 1, 0, 0, 150, 25, 0.17, 0.03, 0.17, 0, 0, 0, 0.05, 0, normal

Figure 5.3 (Feature Columns from the original KDD cup99 before transformation)

0, 1, 9, 10, 491, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 1, 0, 0, 150, 25, 0.17, 0.03, 0.17, 0, 0, 0, 0.05, 0, 1,0,0,0,0

Figure 5.4 (Numeric form Feature Columns after transformation)

2. Implemented scaling.

This step should be applied for the testing dataset as done in the training dataset. The Hopfield net is normally used with discrete binary input so that a

binary encoding method (Liu , Florez and bridges ,2002) must be used to generate acceptable inputs for Hopfield net in phase 2 and SVM in phase 3.

5.2.1.3 Feature Clustering Unit:

The data set will be combined into groups (clusters) according to the attack type. The cause of its simplicity and efficiency K-means or K-Nearest Neighbors clustering algorithms will be used as the clustering method.There are 41 features where each feature is considered in a column which has either discrete values or continuous values (Xu , 2006).

5.2.1.4 Clustering and Selection Unit:

Cluster can be defined as a collection of similar data within the same cluster and dissimilar to the other object in another cluster. Clustering and Selection processes which represent the foundation of this phase have two main jobs:-

1. **Classifications**
2. **Sample selection unit**

1. **Classifications** can be carried out by using K-means or K-nearest neighbor classifier that uses statistical properties and distance measures to cluster information into specific groups. The result of K-mean and k-Nearest Neighbors will be saved in a different table.

- **K-means Clustering Algorithms**

The idea for this algorithm is to classify an object based on features into ***K*** number of group. Where ***K*** is non negative integer number, the grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid, which is the average of all the points in the cluster i.e., its coordinates are the arithmetic means for each dimension separately over all the points in the cluster.

The better choice is to place them as much as possible far away from each other. Thus, the purpose of K-mean clustering is to classify the data. The basic step of K-means clustering is simple. In the beginning, we determine the number of cluster K, and we assume the centroid or center of these clusters. We can take any random objects as the initial centroids, or the first K objects can also serve as the initial centroids.

Then the K means algorithm will do the three steps below until convergence iterate until no more changes are done.

1. Determine the centroid coordinate.
2. Determine the distance of each object to the centroids.
3. Group the object based on minimum distance (find the closest centroid).

Figure 5.5 shows that these iterative steps will continue until centroids do not change. Finally, this algorithm aims at minimizing an objective function; in this case a squared error function. The objective function is as follows:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|_p^2 , p = 1,2,\infty \quad \text{5.1}$$

Where $\left\| x_i^{(j)} - c_j \right\|_p^2$, is a chosen as a distance measure between a data point $x_i^{(j)}$ and the cluster centre **C**j, is an indicator of the distance of the ***n*** data points from their respective cluster centres. The algorithm is also significantly sensitive to the initial randomly selected cluster centres. The K-means algorithm can be run multiple times to reduce this effect. The main advantages of this algorithm are its simplicity and speed which allow it to run on large datasets (MacQueen ,1967).

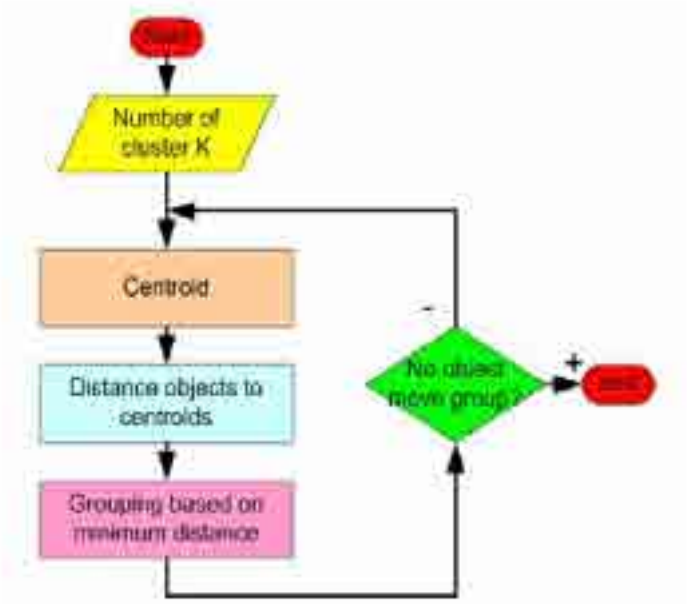


Figure 5.5 (Flowchart of K-means Algorithm)

- **K-Nearest Neighbors Algorithms**

The general principle K-Nearest Neighbors work is to find the k training samples to determine the K-Nearest Neighbors based on a distance measure. To find K-Nearest Neighbors, we sort the distances of all KDD CUP'99 training samples to the new instance and determine the k-th minimum distance.

- Find the majority of the category of K-Nearest Neighbors.

K- Nearest Neighbor clustering algorithm procedure:

1. Store KDD Cup'99 training dataset with its corresponding label.
2. For each connection in the KDD Cup'99 testing dataset, calculate the norm difference (distance) with every connection in the training set using the first norm. The maximum absolute column sum norm $\|A\|_1$ is defined as (Naoum, 2011):

$$\|A\|_1 = max_j \sum_{i=1}^{n} |a_{ij}| \quad \text{.........................} \quad 5.2$$

3. Sort the distances in an ascending order (using sort algorithm such as Q sort). Then with k=3, select the minimum three nearest neighbors.
4. Select the repeated label among these examples. This label is the prediction of this test example.
5. Repeat the above procedure for all the connections in the testing dataset.

After classifying the testing set into 5 classes using k-Nearest Neighbor, the dataset will also be classified using the Enhanced Hopfield neural network.

2. **Sample selection unit,** the main task of this unit is to make a comparison of corresponding class from K-means and K-Nearest Neighbors results. This will take place to establish the main cluster unit by selecting representative sets of significant vector as samples for each class (Normal, DoS, Prob, R2U and U2R). Then it will make two copies of these classes of samples and then will send them to the training unit. One copy will be sent to the neural network model whereas the other one will remain independent to be used later as an input for SVM unit.

5.2.2 PHASE 2:-

It is the training unit phase or can be described as a re-classifying (clustering) unit. The main role of this unit is to train the system for misuse intrusion detecting. It is built of a supervised neural network that will act as a classifier to determine the type of attacks, by using enhanced Hopfield neural network (as a supervised learning with a fixed weight) to generate new representative sets for each class of intrusion according to attack types (DoS, Prob, R2U and U2R), and receives a copy of main attack types vector from the features that have been extracted from the output of selection unit (phase1) and then to enhance them to get new representative samples of attacks type.

These new vectors will be carried out using Enhanced Hopfield to increase the comparable set of attacks representative samples to expand the ability to catch novel patterns of attacks. Test data will pass through the saved trained model to detect intrusions in the testing phase.

- **Hopfield Neural Network Learning Algorithm**

The general learning idea behind the Hopfield network is that the weights between these nodes which produce an output of 1 (active nodes) will increment while those between all other nodes will decrement. This process is repeated for all patterns until the weights reach a stable point (stop changing). Hence all weights must be normalized to guarantee that this is accomplished (Hopfield, 1988).

Step 1: Initialize and assign connections Weights

$$Wij = \begin{cases} \sum_{n=0}^{N-1} x_i^n x_j^n & i = j \quad ; x_i^n \text{ can be } -1 \text{or } +1 \\ 0 & i - j \quad , 0 \le i, j \le M-1 \end{cases} \qquad \text{5.3}$$

In this formula, $\mathbf{W}_{ij}$ is the connection weight between node **i** to node **j** and x^{n}_{i} , can be **+1** or **-1** ,is an element ***i*** of exemplar for class **n**

Step 2: Initialize with unknown input pattern

$$\mu i(t) = xi \qquad 0 \le i \le N-1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.4$$

In this formula, $\mu_i(t)$ is the output of node i at time t and x^n_i , which can be +1 or -1, is the element i of the input pattern.

Step 3: Iterate until convergence

$$\mu i(t+1) = fn\left(\sum_{i=0}^{N-1} w_{ij}\ \mu(t)\right), 0 \le j \le M-1 \quad \ldots\ldots\ldots\ldots 5.5$$

The function fn is the hard limiting nonlinearity. The process is repeated until output remains unchanged with further iterations. The node outputs then represent the exemplar pattern which is the best matches to the target.

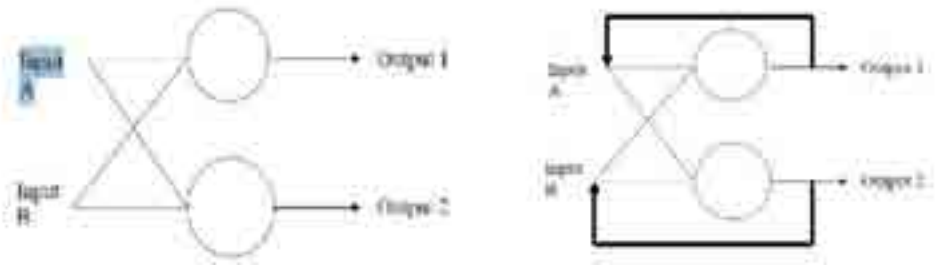
Step 4: repeat by going to step 2 otherwise until the nodes output remains unchanged with further itetrations.

$$if\ \|\mu_i(t+1) - \mu_i(t)\| < \epsilon \quad i = 1,2,\ldots\ldots \alpha \to 0 \ or\ w_{ij} \to 0, \qquad \ldots\ldots\ldots\ldots 5.6$$
$$\epsilon\ is\ a\ small\ real\ number\ say 10^{-6}$$

Step5: END.

- ## Hopfield Neural Network Model and its Enhancement

The Hopfield is a form of feedback (recurrent networks) ANN. Hopfield added feedback connections from the output to the network as inputs again, when the outputs do not change any more, we stop. Figure 5.6 shows what the addition of feedback means. In figure 5.6(b) the difference and the addition to the Feed forword networks shown in figure 5.6 (a).



(a) Feed forword network (b) after we add feedback connection

Figure 5.6 (Different Between Feed Forword and Feedback)

Hopfield also represents a supervised ANN engine, which was invented by John Hopfield in 1982 (Hopfield, 1982) in his highly famous readable research paper "*Neural networks and physical systems with emergent collective computational abilities"*. This type of ANN has the following properties (Naoum, 2011):

1- It is auto-associative (the target values are the same as the inputs).
2- Always settles for any input.
3- Can be settled by modifying the weights on the connections or by changing the activation function.
4- It is a recurrent type ANN.

It's also called Thermo Dynamic Models. It is considered the simplest architecture amongst other ANN. It consists of a single layer Feedback neural network. There is no difference between input and output neurons Figure 5.7; a set of neurons, where each neuron is connected to the other neurons (there is no self-feedback). The weight between these nodes which produces an output 1(active node) will increment while those between all other nodes will decrement. This process is repeated for all patterns until the weights stop changing (that is converge to stable value). Hence, all weights must be normalized to guarantee that this is accomplished.

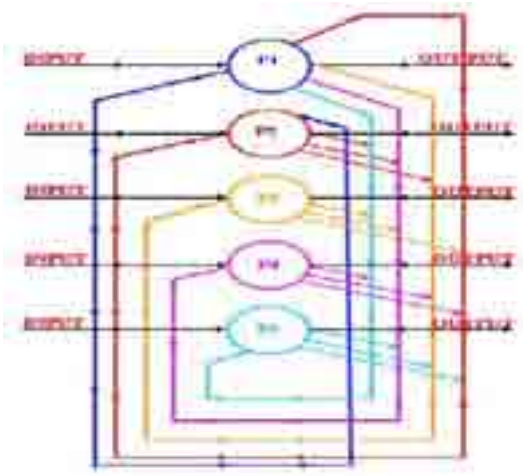


Figure 5.7 (Architecture 0f Hopfield Net with 5 Neurons)

It serves as a content –addressable memory system with binary threshold, which means that these units take only one of two different values for their states. The value is determined by whether the units input exceed their threshold or not. Hopfield nets have units that take on values of 1 or -1, or units that take on values of 1 or 0 (Hopfield, 1982). The two possible definitions for unit i's activation, a_i , are:

$$ai \leftarrow \begin{cases} 1 \text{ if } \sum_j w_{ij} s_j > \theta_i \\ -1 \quad otherwise \end{cases} \quad \text{Or} \quad ai \leftarrow \begin{cases} 1 \text{ if } \sum_j w_{ij} s_j > \theta_i \\ 0 \quad otherwise \end{cases} \qquad \ldots\ldots\ldots\ldots 5.7$$

Where:-

- W*ij* : weight of the connections from unit j to unit i
- S*j* : is the state of unit j
- Θ*i* : is the threshold of unit i

The connections have the following restrictions in a Hopfield net

- W*ii* = 0 for all $i \leftarrow$ it means that no unit has a connection with itself.
- W*ij* = W*ji* for all $i, j \leftarrow$ it means that connections are symmetric.

In order to guarantee that the energy function decreases monotonically while following the activation rules, asymmetric weights will be used. In other cases, if non-symmetric weights are used, W*ii* ≠ W*ij* ,the ANN may exhibit some cyclic or confused performance. Hopfield ANN net has a scalar value associated with each state of the net work referrers as the energy, ***E***, of the network, where

$$E = -\frac{1}{2} \sum_{ij} W_{ij} s_i s_j + \sum_i \theta_i s_i \qquad i,j = 1,2,3,\ldots \qquad \ldots\ldots\ldots\ldots 5.8$$

E is called the "energy function" which should be minimized so as to ensure that if units are randomly chosen to update their activations the network will converge to a state which is *local minima* in the energy function; thus ,if it reaches the local minima, it is a stable state for the network .The energy function , ***E*** ,of the Hopfield net is reduced at each iteration. The number of iteration is finite, so the Hopfield net achieves a stable state (diagonal elements in the interconnection matrix is 0) (Lee, 1991). Thus, Hopfield nets guaranteed converge to local minima, but convergence to one of stored patterns is not guaranteed. This net is more appropriate when input values are actually discrete. If the input values are continuous, they must

be discredited, and analog quantities must be converted to binary values. Table 5.7 summarized our Enhanced Hopfield Neural Network parameters.

Table 5.7 (Parameter Used In Enhanced Hopfield Neural Network)

NO#	PARAMETER	NAME
1	Network type	Recurrent Hopfield
2	Number on input	42
3	Number of output	42
4	Hidden layers	0
5	Hidden layers size	0
6	Input and output ranges	[-1,0,1]
7	Training function	Sigmoid function
8	Adaptation learning function	Mean square Error(MSE)
9	Transfer function	Threshold
10	Training epochs	9888

In this model we shed light on using Enhanced Hopfield NN as the main part of our model (model engine). We use it mainly to detect intrusion of labeled pattern or data as shown in figure 5.8, and to classify them into the 5-class (Normal.Dos,R2L,U2R and Probe) by comparing the new inputs (data vectors) with those stored in main cluster unit. In this unit, training is done in order to produce new descriptive and more accrued vectors for each class from the main five classes.

First of all, we compute the weight of our matrix in order to start training according to the number of features we want to examine. N, which it is equal to the number of input 41 features, the matrix will be N*N square matrix. In our system, the dimension of our matrix is 41*41 matrixes, then adaption on the connections weights is done repeatedly until a stable state is reached (the diagonal of weights matrix=0), and the matrix is symmetric. Learning a correlation (association) between some input and output pattern needs a separate training session, in which the input pattern is presented along with the target pattern (desired output pattern).

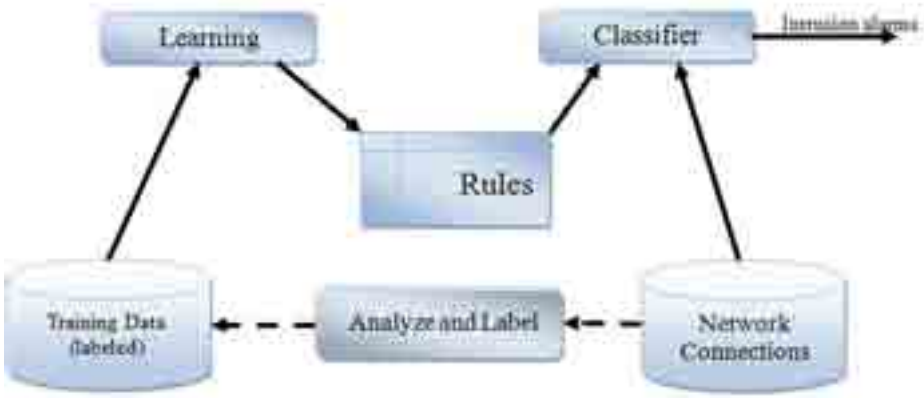


Figure 5.8 (Network Intrusion Detection Using Labeled Data (Al Rashdan W.,2011))

To increase the performance of our model, we use the following two techniques:

1. We make use of Enhanced Hopfield Neural Network for selecting more appropriate packet fields, and considering the relationship of packet sequences in order to support our SVM for learning without pre-existing knowledge.
2. We make use of two clustering algorithm K-means / K-Nearest Neighbor for creating a profile of normal and attack type's packets.

5.2.3PHASE 3 :-

In this phase we represent the Testing Unit: It contains vector classifier tools (SVM). The main object of this tool is to increase the speed of intrusion detection process. The learning phase process is as follow: Firstly, data pattern will be tested to decide if the pattern is normal or not.

Thus, we expose them to both normal and abnormal data. So to get more accurate results in this stage, we use labeled vectors, i.e. in the training subset, and these records are labeled as normal or abnormal. This unit mainly contains Enhanced Hopfield Neural Network and the two types of classifiers, and we form each of them separately and independently.

Our experiment shows that this methodology produces efficient results and accurate training. Hence, the input vectors have been prepared and grouped in a primary clusters in the clustering unit. Such a way of preparing process will help in

minimizing the amount of time needed for classification by Hopfield NN, and so it helps in minimizing FPR and FAR. The main job of this phase is to train the proposed system for detecting intrusion (misuse), and to generate new representative sets for each class of intrusions according to the attack types based on the output of selection unit.

This unit can be described as a re-classifying and re-clustering unit. Thus, SVMs can be considered as learning machines that plot the training vectors in high–dimensional feature space and labeling each vector by its class. SVMs classify the data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in the feature space. However, one of the main disadvantages of the supervised algorithm is that it requires labeled information for efficient learning.

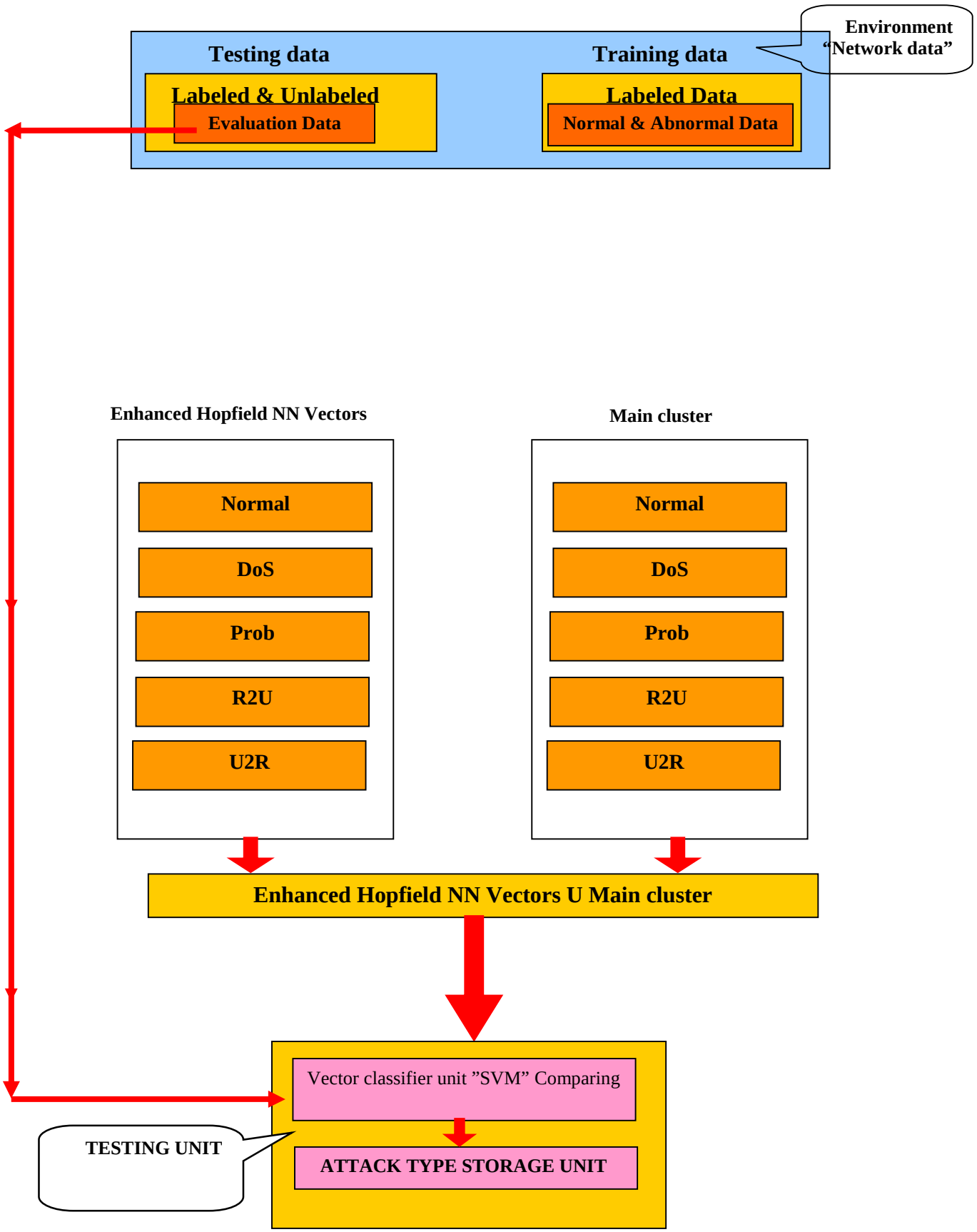


Figure 5.9 (Phase 3 Of The Proposed Model)

Testing phase consists of three units:

After the training phase, we will start the testing phase, figure 5.9, to detect new attacks, and this can be done by applying three units.

5.2.3.1 Processing unit :

The main goal of this unit is to achieve the purposes of our model, which are to detect the misused intrusion and to classify each attack into the main five types of attack. It consists of two subunits, figure 5.11:-

- **codification unit**

By this unit, some useless data will be filtered and modified. There are several text words in the experiment dataset. The system transforms text into numeric values in advance. Every process in the database has 41 attributes shown in table 5.1. In our system, this unit will convert SVM inputs (data patterns) into binary code. We add this unit /codification unit/ to SVM as a support unit to achieve the homogeneity of data format among all phases of our system and their units.

- **feature extraction unit**

It is an attribute reduction process; the process of transforming the input data into a set of features is called feature extraction. It is necessary to achieve a high –performance ID process when using machine learning methods, as we mentioned before in the benchmark dataset of KDD'99, a 41–dimensional features vector was constructed as we showed in table 5.1.

5.2.3.2 Vector classifier unit (SVM)

This unit receive patterns from 3-resourses (Enhanced Hopfield NN, Main Clusters, Environment (Test Data)), then it compares the environment data vector with the combination of two vectors sets (Enhanced Hopfield NN output vectors and Main Clustered vectors) to detect the intrusion and classify its type. Among the variety of misused detection approaches, the Support Vector Machine (SVM) is known to be one of the best machine learning algorithms to classify abnormal behaviors.

The soft margin SVM is one of the well –known basic SVM algorithms using supervised learning (Mukkamala S. and Sung A., 2003).

However, it is not appropriate to use the soft margin SVM algorithm for detecting novel attacks in internet traffic since it requires pre-acquired learning information for supervised learning procedure and such pre-acquired learning information is divided into normal and attack traffic with labels separately.

As the SVMs are only capable of binary classifications, we will employ a one-to-one multi-class SVM. Thus we develop a five SVMs for the five-class (NORMAL, DoS, Prob, U2R , R2L) identification problem in proposed IDS model as shown in figure 5.10 as the most definitive set of features may differ from class to class, using five SVMs becomes a particularity rather than a problem .

We provide a sigmoid function to the kernel feature of SVMs to select support vectors along the surface of this function. These support vectors are members of the set of training inputs and will be used by SVMs to classify data pattern. These support vectors will form a model, by the SVM, representing a category.

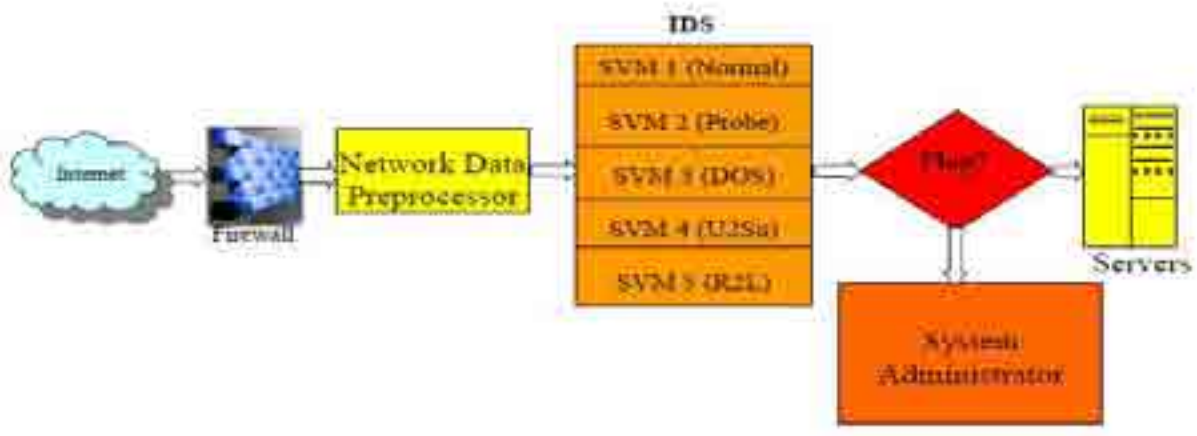


Figure 5.10 (An IDS with 5-SVMs (Mukkamala S. and Sung A., 2003))

5.2.3.3 Storage unit

This unit stores the result (vector) of SVMs and represents it as a table. This subunit keeps the pattern of labeled data by the class name to which it belongs, and this depends on the evaluation process that has been performed by SVMs. The work flow of the system, in the testing phase, can be summarized by figure 5.11.

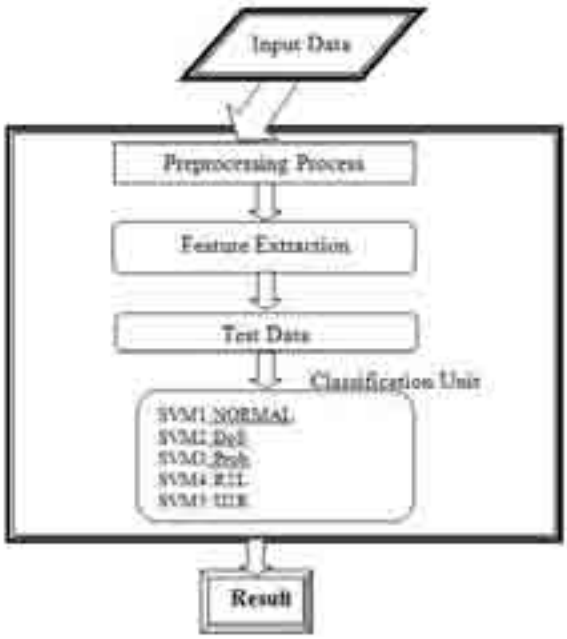


Figure 5.11 (The Workflow of the System in Testing Phase (Mukkamala S. and Sung A., 2003))

Chapter Six

Performance Evaluation and Experimental Results.

6.1 Introduction.

In this chapter, we will present the results that are produced by using the two proposed models HNKMIDS and HNKNNIDS. Section 6.2 contains the evaluation of the proposed IDS model. Section 6.3 contains the Dataset Evaluation. Section 6.4 contains the KDD Cup ‘99 Testing Dataset. Section 6.5 contains the two implementing techniques and their results. Section 6.6 contains the Comparison between Experimental models. Section 6.7 Compares Other Research Result

6.2 Evaluation of Proposed Intrusion Detection System

One of main issues involved in solving problems or trying to find optimal solutions for them is how we can test the results of applying the proposed systems. As for the two IDS proposed models, testing proposed algorithm can provide a good indicator on whether the proposed algorithm can give high performance compared with others or not. Thus, in order to check the performance of the proposed models, we should compare our results with the other results.

Two common ways are used as evaluation measurements for intrusion detection:

1. False-Positive Rate (FPR).

2. Detection rate.

The first is called a false-positive rate (FPR) which appears when the action happened on the system and classified as an abnormal signature (a possible intrusion). Although this type of error may not be completely deleted, a good system should minimize its occurrence to provide useful information to the users. A false-negative (FN) appears when an actual intrusive action has occurred, but the system allows it to pass as non-intrusive behavior. However the true-positives (TP) and true-negatives (TN) are correct classifications. Recall rate measures the proportion of actual positives which are correctly identified. Precision rate is the ratio of true positives to combined true and false positives (Al-Rashdan et al., 2010).

A short summary of each estimated parameter used to evaluate the efficiency of our model and each concept definition to the alarm is described as follows (Al-Rashdan, 2011, Revathi & Ramesh, 2011):

- **True Positive (TP):** Refers to when an attack has occurred and an alarm rises properly
- **True Negative (TN):** Refers to when no attack takes place and no alarm raises.
- **False Negative (FN):** Refers to when an attack has occurred but no alarm raises.
- **False Positive (FP):** Refers to when an alarm rises, but no attack has occurred

The second way which is used to evaluate intrusion detection called detection rate.

Detection Rate (DR): or classification rate for all classes (5 classes) where the system is evaluated by calculating the corrected classified records for each sub class (5 classes) of the total number of records.

DR= (Number of detected attacks / Total number of attacks included in data set) **……6.1**

Accuracy Rate (AR): To estimate the performance of the system is evaluated by calculating the ratio of correctly classified records as attacks (either normal or attack) to the total number of records.

Accuracy Rate (AR) = (TP + TN) / (TP + TN + FN + FP) **……6.2**

Recall Rate: Measures the proportion of actual positives which are correctly identified.

Sensitivity or Recall Rate = TP / (TP + FN) **……6.3**

Error Rate (ER): Incorrectly classified samples divided by the classified samples. Inconclusive results are not counted.

Error Rate (ER) = (FP + FN) / (TP + TN + FN + FP) **……6.4**

Precision Rate (PPV): Measures the ratio of true positives (TP) to combined true and false positives (FP).

Precision Rate = TP / (TP + FP)	**……6.5**

False Positive Rate (FPR): Is the ratio of incorrectly classified normal records (false alarms) to the total number of normal records.

False Positive Rate = FP / (TN + FP)	**……6.6**

False Negative Rate (FNR): Is the ratio of incorrectly classified attacks (when system classifies attacks as normal) records to the total number of attack (intrusions) records.

False Negative Rate = FN / (TP + FN)	**……6.7**

6.3 Dataset Evaluation

Here, we defined higher-level features that help in distinguishing normal connections from attacks. There are several categories for derived features that were used in our experiments. They originate from MIT's Lincoln Lab which was developed for KDD (The Knowledge – Discovery and data mining). KDD Cup' 99 ID dataset provides designers with IDS with a benchmark in order to evaluate separate and different methodologies. A complete listing of the set of 41 features defined for the connection records is given in the three tables below.

Table 6.1 (Basic features of individual TCP connections (KDD, 1999))

feature name	*description*	*type*
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	number of ``wrong'' fragments	continuous
urgent	number of urgent packets	continuous

Table 6.2 (Content features within a connection suggested by domain knowledge (KDD, 1999))

feature name	*description*	*type*
hot	number of ``hot'' indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	number of ``compromised'' conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	discrete
su_attempted	1 if ``su root'' command attempted; 0 otherwise	discrete
num_root	number of ``root'' accesses	continuous
num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous
num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the ``hot'' list; 0 otherwise	discrete
is_guest_login	1 if the login is a ``guest''login; 0 otherwise	discrete

Table 6.3 (Traffic features computed using a two-second time window (KDD, 1999))

feature name	*description*	*type*
count	number of connections to the same host as the current connection in the past two seconds	continuous
Note: The following features refer to these same-host connections.		
serror_rate	% of connections that have ``SYN'' errors	continuous
rerror_rate	% of connections that have ``REJ'' errors	continuous
same_srv_rate	% of connections to the same service	continuous
diff_srv_rate	% of connections to different services	continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
Note: The following features refer to these same-service connections.		
srv_serror_rate	% of connections that have ``SYN'' errors	continuous
srv_rerror_rate	% of connections that have ``REJ'' errors	continuous
srv_diff_host_rate	% of connections to different hosts	continuous

As we mentioned in above tables, features are grouped into three categories (KDD'99, 1999).

1. Basic features: These can be derived from packet headers without inspecting the payload, Table 6.1.

2. Content features: Domain knowledge is used to assess the payload of the original TCP packet, Table 6.2.

3. Traffic features: These features can be grouped into two categories, Table 6.3:

    - Time-based: Features that are designed to capture mature properties over 2 seconds temporal windows.

- Host-based: Features utilize historical window estimated over the number of connections instead of time

6.4 KDD Cup '99 Testing Dataset

Training dataset was used to tune the weights, and testing dataset was used for the network evaluation. Testing dataset contains some novel attacks that are not shown in the training dataset.

Our data sample contains (3975) instances as a Normal class. Also, our data sample contains four attack classes. The first attack class, in our data sample, is denial of service (DoS). It contains (5401) instances. The second attack class, in our data sample, is User to Root (U2R). It contains (52). The third attack class, in our data sample, is Remote to Local (R2L). It contains (97) instances. The forth attack class, in our data sample, is Probes Attack (Probe). It contains (363) instances, as shown in the following table 6.4.

Table 6.4 (Testing Datasets (Labelled) Analysis Details)

Testing (labelled) Datasets	Class Size
Normal	3975
Denial of Service (DoS)	5401
User to Root (U2R)	52
Root to Local (R2L)	97
Prob.	363
Total	9888

The testing dataset (unlabeled) details are represented in table 6.5 below:

Table 6.5 (Testing Datasets (Unlabeled) Analysis Details)

Testing Dataset (Unlabeled)	Class Size
Unknown	4500

6.5 Implementing Technique and results

As we declared earlier, through our two models, we used different algorithms (K-means or K-nearest neighbor), Enhanced Hopfield ANN, and multi class SVMs (5-Class SVM) for each model. The performance outputs of all possible combinations from those techniques were compared with other researches' results, such as Hybrid

(K-means & Naïve Bayes Classifier,SOM) which is considered an indicator of IDSs performance and efficiency.

6.5.1 K-Nearest Neighbor classification Results

Here, we will apply K-Nearest Neighbor algorithm as a classification to the labeled data used for training, with use MATLAB (Version 7.13.0.564 (R2011b)). As shown in table 6.6, we represent the parameter that was used in the experiment. The K-Nearest Neighbor classifier will be used to classify the testing dataset into five classes; these five classes are: Normal, DoS, R2L, R2U and Probe.

Table 6.6 (KNN Parameters)

K- Nearest Neighbor value	1
Distance Measure	Norm -1

K-Nearest Neighbor results for labeled testing dataset are given in the table 6.7 below:

Table 6.7 (KNN Classification Results DR (Labeled))

Testing Datasets (Labelled)	Class Size	Detected Size	Attack Detection Rate
Normal	3975	3705	93.21%
DoS	5401	3896	72.13%
Prob	363	215	59.23%
R2L	97	78	80.41%
U2R	52	34	65.38%
Total	**9888**	**7928**	**80.18%**

According to table 6.7, we used (9888) instances. The correctly classified instances were (7928), which represents (80.18%) of the total testing set. On the other hand, there were (1960) incorrectly classifies instances that represent (19.82%) of the total number instances in the testing set. KNN was able to classify normal class with good detection rate, but Probe and U2R attack, which is considered one of the hardest class to be classified which have an attack detection rate of about 59.23%, 65.38% simultaneously.

DoS attacks have a reasonably good detection rate of about 72.13%, but on the other hand, KNN had a good result in R2L detection rate of about 80.41%. Classifiers are best judged by the classification rate distribution in the confusion matrix which is

considered as the best way to determine whether KNN will produce an acceptable result or not. Table 6.8 demonstrates the confusion matrix of the KNN classifier:

Table 6.8 (K-Nearest Neighbor classifier Confusion Matrix)

K-Nearest Neighbor method CONFUSION MATRIX						
CLASS	**Normal**	**DoS**	**Prob**	**R2L**	**U2R**	**TOTAL**
Normal	**3705**	138	69	50	13	**3975**
DoS	247	**3896**	300	520	438	**5401**
Prob.	69	0	**215**	53	26	**363**
R2L	14	4	0	**78**	1	**97**
U2R	0	0	17	1	**34**	**52**
TOTAL	**4035**	**4038**	**601**	**702**	**511**	**9888**

In the following tables 6.9 and 6.10 there a demonstration of the TP, FP, FN, Precision, Recall, FPR and FNR for each attack:

Table 6.9 (K-Nearest Neighbor classifier TP,FP,FN)

Class name	Real Class RC	Detected class EC	TP	FP	FN
DoS	5401	3896	5154	138	247
Prob.	363	215	294	69	69
R2L	97	78	83	50	14
U2R	52	34	52	13	0
TOTAL RECORD =			**5583**	**270**	**330**

Table 6.10 (K-Nearest Neighbor classifier Precision, Recall, FPR and FNR)

Class name	Real Class RC	Expected class EC	Recall(NPV)	Precision= TP/(TP+FP)	FNR
DoS	5401	3896	95.43%	97.39%	4.57%
Prob.	363	215	80.99%	80.99%	19.01%
R2L	97	78	85.57%	62.41%	14.43%
U2R	52	34	100%	80%	00.00%

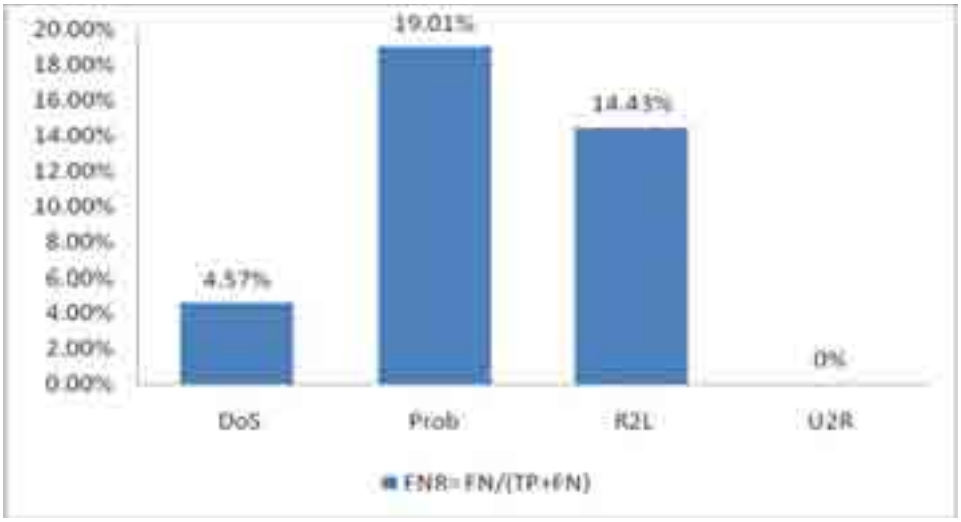


Figure 6.1 (False Negative Rate for Each Class (KNN Classifier))

As noticed for training labeled data that KNN algorithm results showed in figure 6.1, FNR result with U2R attack has a False Negative Rate 0%. DoS attack has about 4.57% False Negative Rate. R2L attack has False Negative Rate of about 14.43%, and Probe attack has about 19.01% False Negative Rate.

Table 6.11 shows the result of applying KNN for testing unlabeled data set and performance in detecting novel attacks with a detection rate of approximately 54%, and this percentage can be described as a bad performance for KNN.

Table 6.11 (KNN Classification Results (Unlabeled))

Testing (Unlabeled) Datasets	Class Size	Detected Size	Detection Rate
Unknown attacks	4500	2418	54%

The evaluation formulas for the KNN classifier are mentioned in the following table 6.12:

Table 6.12 (KNN Classifier Evaluation Formulas)

True Positive (TP) = 5583	False Positive (FP) = 270
False Negative (FN) =330	True Negative (TN) = 3705
Recall (NPV) = TP/(TP+FN) =94.42 %	Precision (PPV) = TP/(TP+FP) = 95.4%
False Positive Rate (FPR) = FP/(FP+TN) = %6.59%	False Negative Rate (FNR) = FN/(FN+TP) = 5.58%
Classification Rate = 7928/9888= 80.18 %	Accuracy Rate = (TP+TN)/(TP+FP+FN+TN) =94 %

The results in the above table 6.12 can be used to judge the KNN classifier. It was able to detect records with a little more Classification rate of more than 80%, but KNN had a low false positive, which means only 270 normal records were detected and classified as intrusion. Therefore, the precision rate was 95.4%, which is considered a high rate. KNN had a very bad performance in terms of false negative rate, where 330 records are attacks and detected as normal. Therefore, using KNN separately, as a classifier, is a bad idea because it can not produce an accurate result for detecting intrusions. KNN results will be compared later with the results of the K-means in order to choose the classifier with a proper result.

6.5.2 K- means Algorithm Results

K-means classifier algorithm will be used to classify the testing dataset into five classes. These five classes are: Normal, DoS, R2L, R2U and Probe.

K-means results for labeled and unlabeled testing dataset are given in the following table 6.13:

Table 6.13 (K- means Classification Results)

Testing (labelled) Datasets	Class Size	Detected or Expected size	Attack detection rate
Normal	3975	3919	98.57%
DoS	5401	4387	81.23%
Prob	363	1226	337.7%
R2L	97	253	260.8%
U2R	52	47	90.38%
Total	9888	9831	99.42%
Testing (Unlabeled) Datasets	Class Size	Detected Size	Detection Rate
Unknown attacks	4500	3865	85.59%

Classifiers are best judged by the classification rate distribution in the confusion matrix. Table 6.14 illustrates the Detection Rate of the K-means classifier:

Table 6.14 (K-means Classifier DR)

Class name	Real Class RC	Detected or Expected class EC
Normal	3975	3919
DoS	5401	4387
Prob	363	1226
R2L	97	253
U2R	52	47

Table 6.15 below illustrates the confusion matrix of the K-means classifier

Table 6.15 (K-means Classifier Confusion Matrix)

	K MEANS Method CONFUSION MATRIX					
CLASS	**Normal**	**DoS**	**Prob**	**R2L**	**U2R**	**Accuracy Rate DR**
Normal	3919	24	13	8	7	90.9
DoS	9	4387	-495	-495	0	1.00
Prob.	18	495	1226	0	381	0.98
R2L	26	495	0	253	-383	0.91
U2R	3	0	-381	331	47	0.91
TOTAL	3975	5401	363	97	52	

Table 6.16 below demonstrates the TP, FP, FN, for each attack type:

Table 6.16 (K-means TP, FP, FN)

Class name	Real Class RC	Detected class DR	TP	FP	FN
DoS	5401	4387	5392	24	9
Prob	363	1226	345	13	18
R2L	97	253	71	8	26
U2R	52	47	49	7	3
TOTAL RECORD			**5857**	**52**	**56**

Table 6.17 below demonstrates the Precision, Recall, FPR and FNR for each attack type:

Table 6.17 (K-means Recall, Precision, FPR, FNR)

Class name	Real Class RC	Detected class DR	Recall= TP/(TP+FN)	FNR= FN/(TP+FN)	FPR= FP/(TN+FP)	Precision= TP/(TP+FP)
DoS	5401	4387	99.83%	0.17%	0.44%	99.56%
Prob	363	1226	95.04%	4.96%	3.46%	96.37%
R2L	97	253	73.20%	26.80%	7.62%	89.87%
U2R	52	47	94.23%	5.77%	11.86%	87.50%

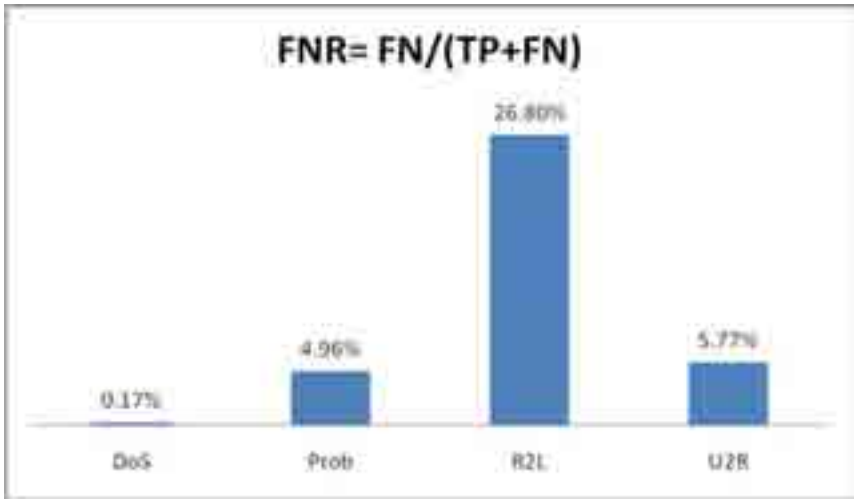


Figure 6.2 (False Negative Rate for Each Class (k means Classifier))

From the above tables 6.17 and figure 6.2, training labeled data, K-means algorithm showed that the FNR result with U2R attack has a False Negative Rate 5.77% ,DoS attack has about 0.17% False Negative Rate, R2L attacks has False Negative Rate of about 26.80% which is the highest rate of all classes and Probe attack has about 4.96% False Negative Rate .Thus, the evaluation formulas for the K-means classifier are given in table 6.18.

Table 6.18 (K-means Classifier Evaluation Formulas)

True Positive (TP) = 5857	**False Positive (FP) = 52**
False Negative (FN) = 56	**True Negative (TN) = 3919**
Recall (NPV) = TP/(TP+FN) =99.05 %	**Precision (PPV) =TP/(TP+FP) =99.12 %**
False Positive Rate (FPR) = FP/(FP+TN) = 1.31 %	**False Negative Rate (FNR) = FN/(FN+TP) = 0.95%**
Classification Rate = 9831/9888= 99.4%	**Accuracy Rate = (TP+TN)/(TP+FP+FN+TN) = 98.91%**

From table 6.18 we conclude that K-means classifier algorithm was able to detect records with a classification rate of approximately 99.4% which is considered a high rate, but K-means algorithm had a low false positive, which means that only 52 normal records were detected as intrusion. Therefore, the precision rate was 99.12% which is considered a high rate. K-means algorithm had a very good performance in terms of false negative rate, where 56 records, which are attacks, were detected and misclassified as normal. Therefore, using K-means classifier is not an appropriate choice because it does not produce an accurate result in detecting intrusions. K-means results will be combined later with the enhanced Hopfield in order to improve the overall performance of the hybrid system (HNKMIDS) and to reach accurate results. Now, we will consider the comparison between the two classifiers according to the following parameters represented in table 6.19.

Tables 6.19 (Comparison between K Mean and KNN Classifier)

MEHOD	True Positive	False Positive	False Negative	True Negative	RECALL	Precision	FPR	FNR	TOTAL DETECTION	Classification Rate	Accuracy Rate
KNN	5583	270	330	3705	94.42%	95.39%	6.79%	5.58%	7928	80.18%	93.93%
KMEANS	5857	**52**	56	3919	99.05%	99.12%	1.30%	0.95%	9831	**99.42%**	98.91%

From Table 6.19, we conclude that K-means produce the lowest false positive with 52 records (instance), KNN classifier has 270 records, K-means produce the better accuracy rate with 98.91% than KNN classifier which has 93.93%, and K-means produces also the better classification rate with 99.42% than KNN classifier which has 80.18% as shown in figure 6.3.

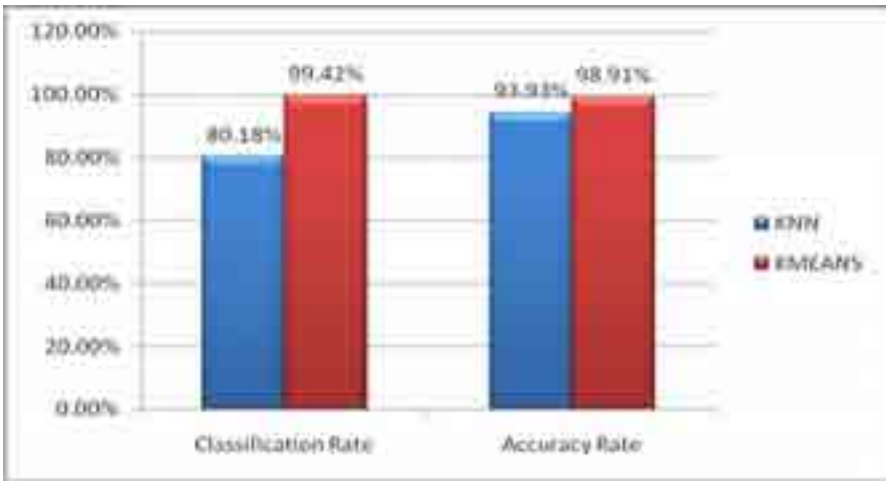


Figure 6.3 (K-mean, KNN Classification rate and Accuracy Rate)

Figure 6.4 (FNR for K-means and KNN)

Looking at FNR in figures 6.4, we conclude that K-means produces the lowest False Negative Rate with 0.95%, followed by KNN with 5.58%. This result reflects that K-means do well and produces more accurate results than KNN.

6.5.3 Enhancement Hopfield Artificial Neural Network with K-means algorithms (HNKMIDS)

At the final stage in classification, the enhanced Hopfield neural network will be used to classify the testing dataset into five classes. Each classifier has its own advantages and disadvantages; therefore, combining both classifiers separately with Enhanced Hopfield NN, we can increase the performance of detection rate and

decrease misclassified attacks. That is, the performance of the hybrid system is improved. The results of the hybrid systems are shown in the following tables, which represent the confusion matrix of the 5 classes (labeled) and the detection rate for the unlabeled testing dataset:

As mentioned in the training phase section, we chose our classifier carefully and precisely according to the element of confusion matrix results. The neural network system was trained using the enhanced Hopfield to improve the performance of the system in terms of classification rate.

Thus the enhanced Hopfield neural network (HNKMIDS) was able to classify the labeled testing dataset as in the following table 6.20:

Table 6.20 (Enhanced Hopfield (HNKMIDS) Results (Labelled))

Testing (labelled) Datasets	Class Size	Detected size	Attack detection rate
Normal	3975	3967	99.80%
Denial of Service (DoS)	5401	5394	99.87%
Prob.	363	352	96.97%
Root to Local (R2L)	97	84	86.60%
User to Root (U2R)	52	30	57.69%
Total	**9888**	9827	99.38%

In table 6.21, Enhanced Hopfield had a reasonable good detection rate for detection unlabeled data set when detecting novel attacks:

Table 6.21 (Enhanced Hopfield (HNKMIDS) Results (Unlabeled))

Testing (Unlabeled) Datasets	Class Size	Detected Size	Detection Rate
Unknown attacks	4500	4379	97.31%

The evaluation formulas for each intrusion attack are best judged by the classification rate distribution in the confusion matrix.

Table 6.22 below illustrates the Detection Rate and Error Rate of the neural network as a classifier:

Table 6.22 (HNKMIDS Algorithm , DR and ER)

Class name	Real Class RC	Detected class	Error Rate(ER)
Normal	3975	3967	**0.002013**
DoS	5401	5394	**0.001296**
Prob	363	352	**0.303030**
R2L	97	84	**0.134021**
U2R	52	30	**0.423077**
TOTAL	**9888**	**9827**	

Table 6.23 below illustrates the confusion matrix of the neural network as a classifier:

Table 6.23 (HNKMIDS Algorithm Confusion Matrix)

	HNKMIDS Algorithm CONFUSION MATRIX					
CLASS	**Normal**	**DoS**	**Prob**	**R2L**	**U2R**	**Accuracy Rate (AR)**
Normal	**3967**	4	8	10	31	**0.998**
DoS	0	**5394**	0	-3	0	**0.999**
Prob.	2	0	**352**	-2	-1	**0.970**
R2L	3	3	2	**84**	-8	**0.866**
U2R	3	0	1	8	**30**	**0.577**
TOTAL	3975	5401	363	97	**52**	
Hopfield and k means SVMs Accuracy Rate=(∑Classes Accuracy Rate)/5						**0.882**

Table 6.24, demonstrates the TP, FP, FN for each attack type:

Table 6.24 (HNKMIDS Algorithm TP, FP, FN)

Class name	Real Class RC	Detected class	TP	FP	FN
DoS	5401	5394	**5401**	**4**	**0**
Prob	363	352	**361**	**8**	**2**
R2L	97	84	**94**	**10**	**3**
U2R	52	30	**49**	**31**	**3**
TOTAL	**9888**	**9827**	**5905**	**53**	**8**

Table 6.25, demonstrate the Precision, Recall, FPR and FNR for each attack type:

Table 6.25(HNKMIDS Algorithm TPR, TNR, FPR, FNR)

Class name	Real Class RC	Detected class	TPR (Recall) =TP/(TP+FN)	Precision =TP/(TP+FP)	FPR= FP/(TN+FP)	FNR= FN/(FN+TP)
DoS	5401	5394	**100.00%**	**99.93%**	**0.07%**	**0.00%**
Prob	363	352	**99.44%**	**97.83%**	**2.20%**	**0. 56%**
R2L	97	84	**96.91%**	**90.38%**	**10.31%**	**3.09%**
U2R	52	30	**94.23%**	**61.25%**	**59.62%**	**5.77%**
TOTAL	**9888**	**9827**				

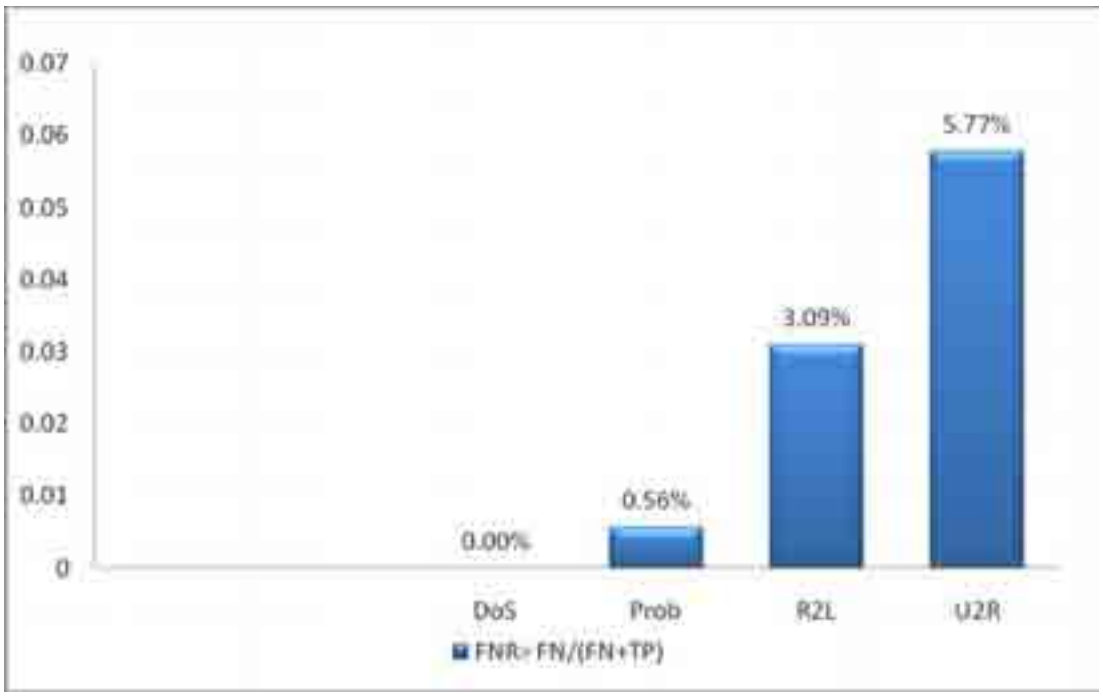


Figure 6.5 (False Negative Rate for Each Class (HNKMIDS Algorithm))

From figure 6.5, we conclude that for training labeled data ,the HNKMIDS, FNR, with Probe attack; has a False Negative Rate of 0.56% ,DoS attack has about 0.00% False Negative Rate, R2L attack has False Negative Rate of about 3.09% and U2R attack has about 5.77% False Negative Rate.

The evaluation formulas for the HNKMIDS model are represented in table 6.26:

Table 6.26 (HNKMIDS Evaluation Formulas)

True Positive (TP) = 5905	**False Positive (FP) =53**
False Negative (FN) = 8	**True Negative (TN) =3967**
Recall (NPV) = TP/(TP+FN) =99.86%	**Precision (PPV) =TP/(TP+FP) =99.11%**
False Positive Rate (FPR) = FP/(FP+TN) =1.32%	**False Negative Rate (FNR) = FN/(FN+TP) = 0.135%**
Classification Rate = 9827/9888=99.38%	**Accuracy Rate = (TP+TN)/(TP+FP+FN+TN) =99.39 %**

From the above table 6.26 we are able to say that HNKMIDS system was able to detect records with a classification rate of approximately 99.38%, which is considered a very good performance and with a False Negative Rate of 0.135%. The recall and precisions for HNKMIDS system has a very good result. The system truthful is guaranteed to detect intrusions; especially, when the false negative records were only 8 (intrusions detected as normal). But, the drawback of the HNKMIDS was the false positive rate. The main advantage of the HNKMIDS system is that it was able to detect normal class with a reasonable good Detection Rate, the HNKMIDS system was also able to detect Denial Of Service (DoS), and Probe. With very high attack detection rate, but unfortunately it was still unable to, correctly, classify the user to root (U2R) root to local (R2L) attack.

6.5.4 Enhancement Hopfield Artificial Neural Network with K-nearest neighbor algorithms (HNKNNIDS)

Now, we will consider Enhanced Hopfield Neural network combined with K-Nearest Neighbor algorithm, and we call it HNKNNIDS. We will demonstrate how to use the HNKNNIDS as a classification to the labeled data, use for train. The HNKNNIDS algorithm will be used to classify the testing dataset into five classes. These five classes are: Normal, DoS, R2L, R2U and Probe and the results for labeled testing dataset are given in the table 6.27 below:

Table 6.27 (HNKNNIDS Classification Results DR (Labeled))

Testing Datasets (Labelled)	Class Size	Detected Size	Attack Detection Rate
Normal	3975	3746	94.24%
DoS	5401	3921	72.60%
Prob	363	230	63.36%
R2L	97	82	84.54%
U2R	52	39	75%
Total	**9888**	**8018**	**81.09%**

According to the previous table 6.27, we used (9888) instances. The correctly classified instances were (8018), which represent (81.09%) of the total testing set. On the other hand, there were (1870) incorrectly classifies instances that represent (18.91%) of the total number instances in the testing set.

Thus HNKNNIDS was able to classify normal class with good Detection Rate of about 94.24%, but Probe and U2R attack which was considered one of the hardest classes to classify; have an attack detection rate of about 63.36% and 75% simultaneously.

DoS attack has an attack Detection Rate of about 72.60%, but on the other hand, HNKNNIDS produced an accurate result for R2L detection, which has an attack detection rate of about 84.54%. Classifiers are best judged by the classification rate distribution in the confusion matrix which is considered the best way to determine whether the model that HNKNNIDS produce an acceptable result. In table 6.28, we demonstrate the confusion matrix of the HNKNNIDS classifier:

Table 6.28 (HNKNNIDS Confusion Matrix)

HNKNNIDS method CONFUSION MATRIX						
CLASS	**Normal**	**DoS**	**Prob**	**R2L**	**U2R**	**TOTAL**
Normal	**3746**	118	58	42	11	3975
DoS	156	**3921**	310	507	507	5401
Prob.	58	-310	**230**	264	121	363
R2L	42	-507	264	**82**	216	97
U2R	40	-507	264	216	**39**	52
TOTAL	**4185**	**2715**	**983**	**1111**	**894**	9888

Table 6.29 is a demonstration of the TP, FP, FN for each attack:

Table 6.29 (HNKNNIDS TP, FP, FN)

Class name	Real Class RC	Detected class	TP	FP	FN
DoS	5401	3921	**5245**	**118**	**156**
Prob	363	230	**305**	**58**	**58**
R2L	97	82	**55**	**42**	**42**
U2R	52	39	**12**	**11**	**40**
TOTAL			**5617**	**229**	**296**

Table 6.30 is a demonstration of the Precision, Recall, FPR and FNR for each attack:

Table 6.30 (HNKNNIDS RECALL, PRECISION, FPR, FNR)

Class name	Real Class RC	Detected class	TPR (Recall) =TP/(TP+FN)	Precision =TP/(TP+FP)	FPR= FP/(TN+FP)	FNR= FN/(FN+TP)
DoS	5401	5394	**97.11%**	**97.80%**	**2.92%**	**2.89%**
Prob	363	352	**84.02%**	**84.02%**	**20.14%**	**15.98%**
R2L	97	84	**56.70%**	**56.70%**	**33.87%**	**43.30%**
U2R	52	30	**23.08%**	**52.17%**	**22%**	**76.92%**

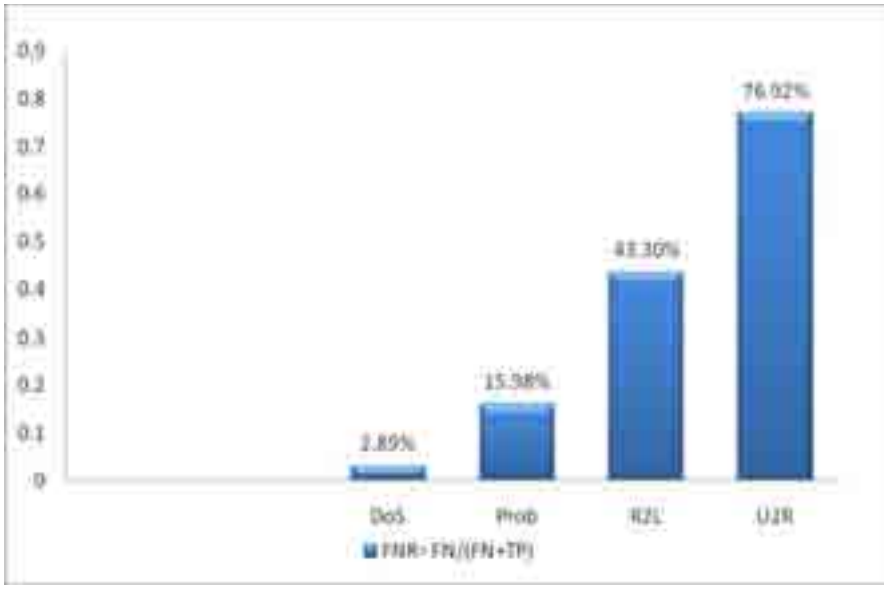


Figure 6.6 (False Negative Rate for Each Class (HNKNNIDS))

From figure 6.6 we conclude that for training labeled data ,the HNKNNIDS, FNR, with U2R attack; has a False Negative Rate 76.92% ,DoS attack has about 2.89%

False Negative Rate, R2L attack has False Negative Rate of about 43.30%, and Probe attack has about 15.98% False Negative Rate .

The next step is applying HNKNNIDS for testing unlabeled data set. The use of HNKNNIDS showed a good performance in detecting novel attacks with a detection rate of approximately 60.67%, as we show in table 6.31.

Table 6.31 (HNKNNIDS Classification Results (Unlabeled))

Testing (Unlabeled) Datasets	Class Size	Detected Size	Detection Rate
Unknown attacks	4500	2730	60.67%

The evaluation formulas, (FN, NPV, FP, TN, FPR, AR, and FNR), for the HNKNNIDS are computed in the following table 6.32:

Table 6.32 (HNKNNIDS Evaluation Formulas)

True Positive (TP) = 5617	**False Positive (FP) = 229**
False Negative (FN) =296	**True Negative (TN) = 3746**
Recall (NPV) = TP/(TP+FN) =94.99%	**Precision (PPV) = TP/(TP+FP) = 96.08%**
False Positive Rate (FPR) = FP/(FP+TN) = 5.76%	**False Negative Rate (FNR) = FN/(FN+TP) = 5.00%**
Classification Rate = 8018/9888= 81.09 %	**Accuracy Rate = (TP+TN)/(TP+FP+FN+TN) =94.69%**

The results from table 6.32 demonstrated that the HNKNNIDS can detect records with classification rate of 81.09%, but HNKNNIDS had a low false positive, which means only 229 normal records, were detected and classified as intrusion. The precision rate was 96%, which is considered a high rate. HNKNNIDS had a very bad performance in terms of false negative rate, where 313 records, which are attacks, were detected as normal. The result of application of HNKNNIDS will be compared later with the results of the HNKMIDS in order to choose the best performance.

6.6 Comparison between Experimental models

Now, we will consider the comparison between the two models according to the following parameters represented in table 6.33.

Tables 6.33 (Comparison between HNKMIDS and HNKNNIDS)

METHOD	HNKMIDS	HNKNNIDS
True Positive	**5905**	5617
False Positive	53	229
False Negative	**8**	296
True Negative	3967	3746
RECALL	99.86%	94.99%
Precision	99.11%	96.08%
FPR	1.32%	5.76%
FNR	0.135%	5.00%
Total Detection	9827	8018
Classification Rate	99.38%	81.08%
Accuracy Rate	**99.39%**	94.69%

From Table 6.33, we conclude that HNKMIDS produced false positive with 53 records (instance), where as HNKNNIDS produced false positive with 229 records (instance). Thus HNKNNIDS had an advantage over HNKMIDS. HNKMIDS performance is the best, in terms of accuracy rate with 99.39% than HNKNNIDS model which has 94.69%, and HNKMIDS performance is better in terms of Classification Rate with 99.38% than HNKNNIDS model, which has 81.08%.

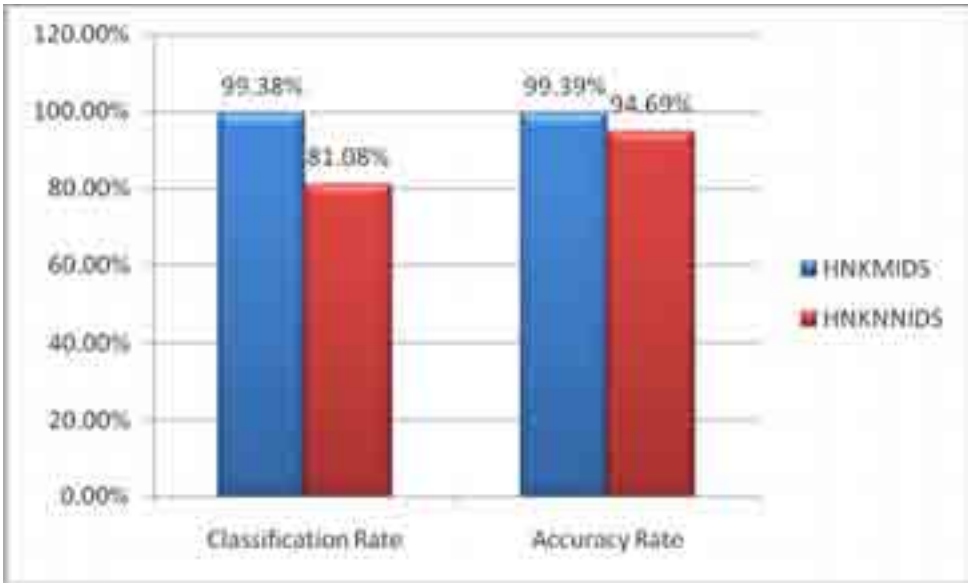


Figure 6.7 (Accuracy Rate ,Classification Rate for Each model)

From figure 6.7, we conclude that HNKMIDS produced an Accuracy Rate of 99.39%. HNKNNIDS produced Accuracy Rate of 94.69%. We also conclude that HNKMIDS produced Classification Rate of 99.38%. HNKNNIDS produced Classification Rate of 81.08%.

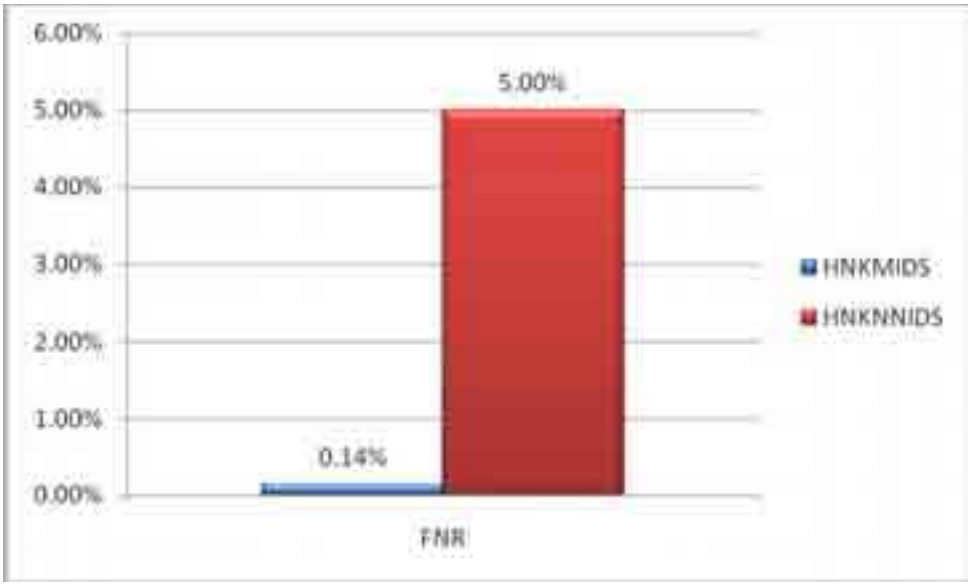


Figure 6.8 (False Negative Rates for Each model)

Looking at FNR results in figures 6.8 we conclude that HNKMIDS produced the lowest false negative rate with 0.14%, followed by HNKNNIDS with 5.00 %. This result reflects that HNKMIDS produced more accurate results than HNKNNIDS.

6.7 Comparing With Other Research Result

Finally, the Enhanced two systems (HNKMIDS and HNKNNIDS) performance is compared to other intrusion detection systems that use either neural network (supervised, unsupervised), K-means machine learning algorithm and Iterative Dichotomiser3 (ID3) which is a decision tree method. In table 6.34, we compare the False Positive Rate for each algorithm:

Table 6.34 (Intrusion Detection System Evaluation Rates vs. Other Systems)

Algorithms	DR	AR	NPV(Recall)	PPV Precision (PPV)	FPR	FNR
***HNKMIDS**	**99.38%**	**99.39%**	**99.86%**	**99.11%**	**1.32%**	**0.14%**
Hybrid (K-means & Naïve Bayes Islim)	97.90%	-	97.90%	98.60%	0.30%	-
SOM -Consience - Al-Rashdan	92.50%	-	94.70%	96.40%	3.50%	5.20%
BP Brifcani and Issa	91.80%	-	-	-	-	-
K-means Nieves	89%	-	-	-	4.80%	-
***HNKNNIDS**	81.08%	94.69%	94.99%	96.08%	5.76%	5.00%

From the above table 6.34, we conclude that the enhanced (HNKMIDS) system has improved the performance over the other algorithms, where the detection rate for the five classes is more than 99.38%. The Recall was improved with 99.86%. The main advantage of the enhanced system (HNKMIDS) is that the false negative rate, which is considered the most critical evaluation, is about 0.14%. On the other hand, the enhanced system (HNKMIDS) has a false positive rate (false alarm) of 1.32%.

Finally, the enhanced system performance is compared to other intrusion detection systems that use either neural network (supervised, unsupervised), K-means machine learning algorithm and naïve bayes classifier.

The enhanced system (HNKMIDS) is compared to K-means intrusion detection system proposed by Nieves (Nieves, 2009). The results demonstrate that the proposed system produced better results in terms of the Accuracy Rate with 99.39% as follows:

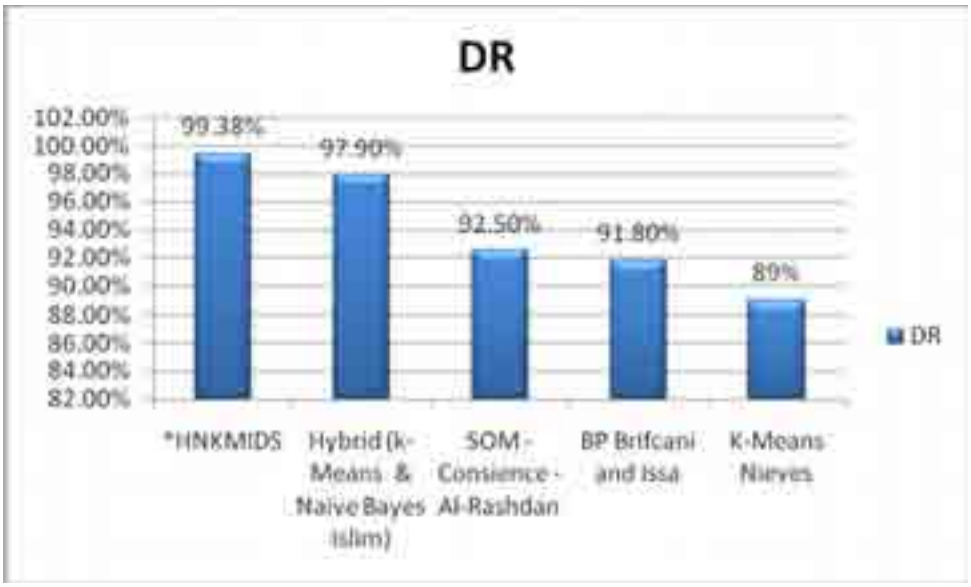


Figure 6.9 (comparison between HNKMIDS models according to detection rates)

By exploring the detection rates, as shown in Figure 6.9, we can conclude that the (HNKMIDS) gains better detection result than others. We can also define the rank of our model amongst other models, as shown bellow in table 6.35. It is very clear that (HNKMIDS) achieved the first rank with 99.38% detection rate.

According to experimental results, (HNKMIDS) achieved the first rank in Detection Rate compared with the other four studies that were mentioned above in Table 6.34. Our model rank can be summarized as seen bellow in table 6.35.

Table 6.35 (The (HNKMIDS) Rank.)

Model	DR	Rank
HNKMIDS	**99.38%**	**1**
Hybrid K-means & Naive Bayes (2012, Islim)	97.90%	2
SOM – Consience (2011, Al-Rashdan)	92.50%	3
BP (2010,Brifcani and Issa)	91.80%	4
K-Means (2009, Nieves)	89%	5

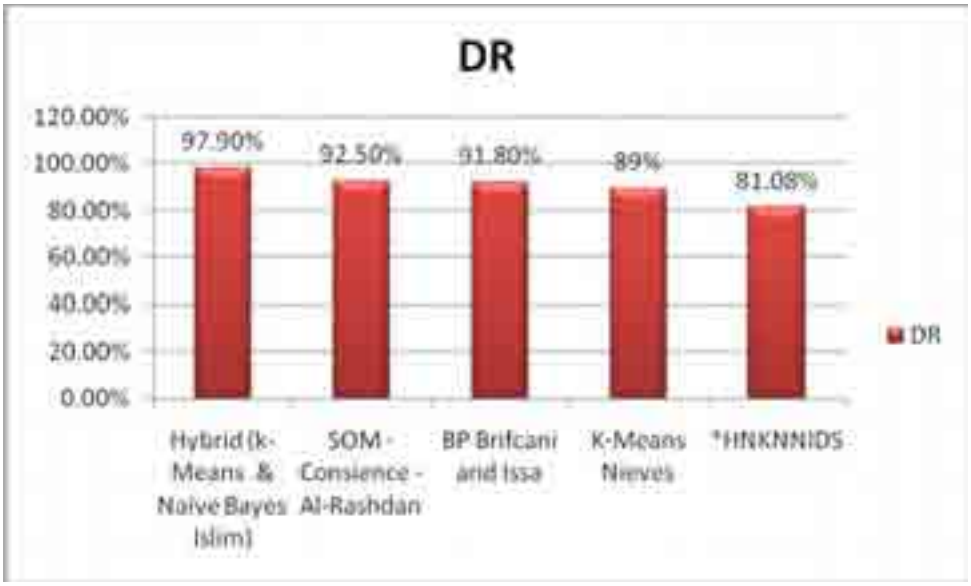


Figure 6.10 (comparison between HNKNNIDS models according to detection rates)

From the above Figure 6.10, we can note that (HNKNNIDS) method achieved the fifth rank with 81.08% as a Detection Rate while the first rank was achieved by (Islim, 2012) Model that was based on combining K-means and Naïve Bayes.

According to experimental results, (HNKNNIDS) achieved the fifth rank in Detection Rate compared with the other four studies that mentioned above in table 6.34. Our model rank can be summarized as seen bellow in table 6.36.

Table 6.36 (The (HNKNNIDS) Rank.)

Model	DR	Rank
Hybrid K-means & Naive Bayes (2012, Islim)	97.90%	1
SOM – Consience (2011, Al-Rashdan)	92.50%	2
BP (2010, Brifcani and Issa)	91.80%	3
K-Means (2009, Nieves)	89%	4
HNKNNIDS	81.08%	5

6.8 Conclusion

Since the early 1980's, research and development of intrusion detection systems has been unending. The challenges and troubles faced by designers have increased as the targeted systems became more distinct and complex. Misuse detection is a particularly difficult problem because of the extensive number of vulnerabilities in computer systems and the creative ideas of the attackers. Artificial Neural networks present a great number of advantages in the detection of these attacks. The early results of our tests of these technologies show a significant promise.

- Using KNN alone as a classifier is not a good choice because it cannot produce an accurate result for detecting intrusions.
- Using K-means as a classifier is not an appropriate choice because it does not produce accurate results in detecting intrusions.

- The main advantage of the HNKMIDS system is that it was able to detect normal class with a reasonable good detection rate. The HNKMIDS system was able to detect Denial of Service (DoS), and probe. With very high attack Detection Rate, it is still unfortunately unable to correctly classify the user to root (U2R) and root to local (R2L) attacks.

- The HNKNNIDS, FNR ,with U2R attack has a False Negative Rate 0% ,DoS attack has about 5.57% False Negative Rate, R2L attack has False Negative Rate about 14.43% and Probe attack has about 19.01% False Negative Rate .

- The use of HNKNNIDS showed a good performance in detecting novel attacks with a good detection rate.

- The HNKMIDS was able to detect novel (unknown) attacks with a good detection rate.

- Adding a unit /codification unit/ to SVM as a support unit to achieve the homogeneity of data format among all phases of our systems has an impact on our systems, which enhanced our models' result.

6.9 Future Work

1. KDD'99 dataset consists of instances with 41 features for each instance. More deep studies are needed to reduce the number of these features in order to increase the accuracy of IDS.
2. Other methods like K-Medoid can be used with supervised and unsupervised learning for intrusion detection field.
3. Using reinforcement learning with our system mechanisms can produce higher TP rate.
4. Combining firewall with IDS system mechanisms and its properties can produce higher TP rate.
5. Reducing the dataset features dimensionality.
6. Using parallel computing.
7. Developing a new Artificial Neural Network to remove local minima.

References

[1] Al-Rashdan, W. Naoum (2011). **"A Hybrid Artificial Neural Network Model (Hopfield-SOM with Conscience) for Effective Network Intrusion Detection System"**. (Doctoral dissertation, The Arab Academy for Banking and Financial Sciences, 2011).

[2] AL-Rashdan, W, Naoum, R, Al_Sharafat, W & Al-Khazaaleh, M. (2010). **"Novel network intrusion detection system using hybrid neural network (Hopfield and Kohonen SOM with conscience function)".** IJCSNS International Journal of Computer Science and Network Security, 10(11). Retrieved January 26 2012, **http://paper.ijcsns.org/07_book/201011/20101103.pdf**

[3] Anderson J. P. (1980).**Computer Security Threat Monitoring and Surveillance**, tech. report, James P. Anderson Co., fort Washington, Pa.

[4] Bishop, M. (2005). "**Introduction to Computer Security"**. Boston: Pearson Education, Inc., .457-459,461-465,469,473,484.

[5] Bridges S.M.,Vaughn R.B.(2000)**,"Intrusion Detection Via Fuzzy Data Mining "**,accepted for 12th annual Canadian information technology security symposium ,june 19-23 2000.

[**6**] Brifcani A.& Issa A. (2011), Intrusion detection and attack classifier based on three techniques: acomparative study, Eng. & Tech. Journal, vol 29, no 2.

[7]Chen, R.C.; Cheng, K.F.; and Hsieh, C.F. (2009). "**Using rough set and support vector machine for network intrusion detection"**. International Journal of Network Security & Its Applications (IJNSA), 1(1), 1-13.

[8] Das, K., J., (2000). **Attack Development For Intrusion Detection Evaluation.** Master thesis, Massachusetts Institute of Technology (MIT),USA.

[9] Dorosz, P., Kazienko P. (2004). "**Intrusion Detection System (IDS) Part 2"**. WindowSecurity.com. Tech Genix Ltd. , [On-Line],Available: http://www.windowsecurity.com/articles/IDS-Part2-Classification-methods-techniques.html]

[10] Elkan , C. (2011).”**Nearest Neighbor Classification”**, University of California, San Diego (UCSD CSE), [On-Line], Available: http://cseweb.ucsd.edu/~elkan/250B/nearestn.pdf

[11] Faraoun K.M. , Boukelif A.(2006).”**Neural Networks Learning Improvement using the K-means Clustering Algorithm to Detect Network Intrusions”.**

[12] Fox, Kevin L., Henning, Rhonda R., and Reed, Jonathan H. (1990). **“A Neural Network Approach Towards Intrusion Detection”**. In *Proceedings of the 13th National Computer Security Conference.*

[13] Gupta C. (2006)**,”Implementation of Back Propagation Algorithm (of neural networks) in VHDL”**. Master Thesis (Deemed University)Patilal-147004,India ,June,2006.

[14] Hammerstrom, Dan. (June, 1993). **“Neural Networks at Work”**. *IEEE Spectrum*. pp. 26-53.

[15] Hebb, (1949).”**The organization of behavior”,** John wiley, pub. ,usa

[16] Hopfield, J. J. (1982). “**Neural Networks and Physical Systems with Emergent Collective Computational Abilities”.** Proceeding of the National Academy of Scientists, 79:25542558.

[17] Hopfield, J. J. (1988).)**,”Artificial neural networks”** ,IEEE 1988]

[18] Horeis T. (2003).”**Intrusion Detection with Neural Networks - Combination of Self-Organizing Maps and Radial Basis Function Networks .for Human Expert Integration”**.

[19] Islim, E. (2012). “**A Hybrid Intrusion Detection Model Based on Human Immune System”**, (Master dissertation), Middle East University, Jordan.

[20] Jaeger R. (2006).”**HIDS / NIDS”**, [On-Line], Available : http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci1211526,00.html#

[21] Katos V. (2007).”**Network intrusion detection: Evaluating cluster, discriminant, and logit analysis”**. Science Direct, Information Science177, PP.3060-3073.]

[22] Kaxienko P.,Dorosz P.(2004). "**Intrusion Detection Systems (IDS) Part I - (network intrusions; attack symptoms; IDS tasks; and IDS architecture)"** .

[23]KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html . Ocotber 2007.)

[24]KDD, the International Knowledge Discovery and Data Mining Tools Competition, (1999). [On-Line], Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[25] kukielka P.,kotulski Z.(2008) **"Analysis of different Architectures of Neural Network For Applications in Intrusion Detection Systems"**International Multi conference on Computer Science and Information TECHNOLOGY,PP.807-811.

[26] Kurdi , W.(2011). **"A Hybrid Artificial Neural Network Model (Hopfield-SOM with Conscience) For Effective Network Intrusion Detection System".** PhD thesis**,** Amman, Jordan.

[27] Kurose, J & Ross, K. (2010). **"Computer Networking A Top-Down Approach (5th ed.)**". Boston: Pearson Education.

[28] LEE. B.J. (1991). **"Modified Hopfield ANN for Retrieving the Optimal Solution".** IEEE Transaction On Neural Networks, vol.2, no.1.

[29] Lee, B.W.,Sheu B.J.(1991**)"Modified Hopfield Neural Network for Retrieving the Optimal Solution",** IEEE Transaction on Neural Network,Vol2,No.1,January 1991.

[30] Lee, W., Stolfo, S.J.: **"Data Mining Approaches for Intrusion Detection. In: Seventh USENIX Security Symposium (SECURITY '98)",** San Antonio, TX (1998).

[31] Lippmann R. P. (1987). "**An Introduction to Computing With Neural Nets", IEEEASSP** MAGAZINE, PP. 4-16.

[32] Liu Z., Florez g. and bridges S.M. (2002)."**A Comparison Of Input Representations In Neural Networks: A Case Study In Intrusion Detection**. ", [On-Line], Available: http://www.Cse.Msstate.Edu/~Bridges/Paper/Ijcnn2002.Pdf]

[33] MacQueen J. B., **"Some methods for classification and analysis of multivariate observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability",** Berkeley: University of California Press, Vol. 1, (1967), pp. 281-297.

[34] Magalhaes R. M. (2003). "**Host-Based IDS vs Network-Based IDS (Part1)"**,[On-Line],Available: http://www.windowsecurity.com/articles/Hids_vs_Nids_Part1.html

[35] Marilyn McCord Nelson and W.T. Illingworth (1991)."**A Practical Guide To Neural Nets"** .

[36] McCulloch, W.H., and Pitts, W.S., **"A Logical Calculus of the Ideas Immanent in Neural Nets"**, Bulletin of Mathematical Biophysics, Vol. 5, 1943, pp. 115–133.

[37] MIT Lincoln Labs, 1998 DARPA Intrusion Detection Evaluation. Available on: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/index.html , February 2008.

[38] Mukherjee B., L. Heberlein, and K. Levitt, **"Network Intrusion Detection,"** IEEE Network, vol. 8, no. 3, May/June 1994, pp. 26-41.

[39] Mukkamala S.,Sung A.H.(2003**)"Feature Selection For Intrusion Detection Using Neural Networks And Support Vector Machines"**.

[40] Muthukkuumarasamy v.,Birkely R**."An Intelligent Intrusion Detection System Based On Neural Network"** www.iadis.netldl/final_uploads/200401l028.pdf(2004)

[41] Naoum, R. (2011). **Artificial Neural Network** [Acrobat Reader], Middle East University (MEU), Jordan.

[42] Nieves, J. (2009). "**Data Clustering For Anomaly Detection In Network Intrusion Detection"**. Oak Ridge National Laboratory. Retrieved November 20, 2011, from http://info.ornl.gov/sites/rams09/j_nieves_rodrigues/Documents/report.pdf

[43] Paul I. (2001)."**An Introduction to IDS"**, [On-Line],Available: http://www.securityfocus.com/infocus/1520

[44] Paul,I. (2001).”**The Evolution of Intrusion Detection Systems.”** , [On-Line], Available:http://www.securityfocus.com/infocus/1514,]

[45] Pfleeger, C. P.,Shari L. Pfleeger (2003). “**Security in Computing”**. 3rd ed. Upper Saddle River: Pearson Education, Inc., 2003. 259, 468-468, 472-473.

[46] Reingold, E & Nightingale, J. (1999). “**Artificial Intelligence tutorial review for psychology students”**, PSY371. Retrieved March 3, 2012, from http://www.psych.utoronto.ca/users/reingold/courses/ai/ai.html.

[47] Rung-Ching C. , Kai-Fan C. ,Chia-Fen H. (2009).”**Using Rough Set And Support Vector Machine For Network Intrusion Detection”**. International Journal of Network Security & Its Applications (IJNSA),Vol 1, No 1.

[48] Scarfone K., Mell P. (2007).”**Guide To Intrusion Detection and prevention Systems (IDPS)”.** Recommendations of the National Institute Of Standards And Technology Computer Security Division Information Technology Laboratory National Institute Of Standards And Technology.

[49] Tanenbaum S. Andrew(2003).**“Computer Networks, Fourth Edition “.**

[50] The MathWorks. MATLAB Help. (2011): The Language of Technical Computing [Online], Available: http://www.mathworks.com.

[51] Wikipedia: The Free Encyclopedia (2005).”**Intrusion-Detection System”.** [On-Line], Available: http://en.wikipedia.org/wiki/Intrusion_detection_system.

[52] Xu X.(2006) **.”Adaptive intrusion detection based on machine learning: feature extraction, classifier construction and sequential pattern prediction”**, international journal of web services practices, vol.2, no.1-2, pp. 49-58.