A HYBRID APPROACH FOR NETWORK INTRUSION DETECTION SYSTEM

By

Samuel Lalmuanawma

(MZU/Ph.D./509 of 29.10.2012)

Thesis submitted in fulfillment for the requirement of the Degree of Doctor of Philosophy in Computer Science

To

Department of Mathematics & Computer Science
School of Physical Sciences
Mizoram University
Aizawl - 796 004
Mizoram, India
December, 2015

DEPARTMENT OF MATHEMATICS & COMPUTER SCIENCE
MIZORAM UNIVERSITY

Dr. Jamal Hussain
Professor

Aizawl – 796 004, Mizoram: INDIA
Phone: +91 9436352389(m)
e-mail: jamal.mzu@gmail.com

CERTIFICATE

This is to certify that the thesis titled "A hybrid approach for network intrusion detection system" submitted by Mr. Samuel Lalmuanawma (Registration No: MZU/ Ph. D./509 of 29.10.2012) for the degree of Doctor of Philosophy (Ph. D.) of the Mizoram University, embodies the record of original investigation carried out by him under my supervision. He has been duly registered and the thesis presented is worthy of being considered for the award of the Ph.D degree. This work has not been submitted for any degree of any other universities.

Prof. Jamal Hussain

(Supervisor)

MIZORAM UNIVERSITY

TANHRIL

Month: December **Year: 2015**

CANDIDATE'S DECLARATION

I, Samuel Lalmuanawma, hereby declare that the subject matter of this thesis titled with "A hybrid approach for network intrusion detection system" is the record of work done by me, that the contents of this thesis do not form basis of the award of any previous degree to me or the best of my knowledge to anybody else, and that the thesis has not been submitted by me for any research degree in other University/Institute.

This is being submitted to the Mizoram University for the degree of Doctor of Philosophy (Ph. D.) in Computer Science.

Samuel Lalmuanawma
(MZU/Ph.D/509 of 29.10.2012)
(Candidate)

Prof. Jamal Hussain
(Supervisor)
Dept. Maths. & Comp. Sc.
Mizoram University

Prof. Jamal Hussain
(Head of Department)
Dept. Maths. & Comp. Sc.
Mizoram University

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisor, Prof. Jamal Hussain also Head of the Department, Mathematics & Computer Science, Mizoram University, for his excellent supervision, encouragement, patience, and providing me an excellent environment through the research work. I also thank the committee members of Departmental Research Committee (DRC) and Board of Studies (BOS), in Mathemetics & Computer Science for providing valuable suggestions from time to time.

I thank Prof. R. C. Tiwari, Dean, School of Physical Sciences, Mizoram University for his help and co-operation during my research work. I also thank my teachers, Dr. J. P. Singh, Mr. Laltanpuia, Dr. S. S. Singh and Mrs. M. Saroja Devi for their valuable suggestions and support. I also thank all the non-teaching staff for their support and kind assistance. I would also like to thank all the research scholars, especially, Mr. C. Zorammuana and Mr. David Rosangliana of the Department for their cooperation and wonderful time we enjoyed together during my research work.

I acknowledge the University Grants Commission (UGC), New Delhi for providing financial assistance as Junior Research Fellow (JRF) and Senior Research Fellow (SRF) under the Rajiv Gandhi National Fellowship for SC/ST to support my research work.

Last but not the least, I would like to give special thanks to my family. Words cannot express how grateful I am to my father, my mother, brother and sister for all of the sacrifices that you've made on my behalf. Your prayer for me has what sustained me thus far. I also thank all of my friends who supported me in writing and incented me to strive towards my goal. In the end, I would like express appreciation to my beloved wife who spent sleepless nights with and was always my support in the moments when there was no one to answer my queries.

Dated: 30th Dec., 2015

Place: Aizawl Samuel Lalmuanawma

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Introduction

The modern communication systems have converted connectivity applications into digital systems. Industries, institutions and organizations associated with a complex computer networks that result in huge services to society in a right approach with accurate high-speed connectivity. These advancements lead to increased risk of intrusion attempts over the network system. Due to these rapid changes, Network Intrusion Detection Systems (NIDSs) are becoming challenging areas of research in computer network security. Therefore, to secure valuable resources over the public network, it is essential to implement an Intrusion Detection System (IDS).

An IDS has been developed for a second line of defense in the security environment. The conventional prevention system such as data encryption, user validation, and firewall, etc. are implemented as the first line of defense for the computer security (Lazarevic *et al.*, 2003). However, intruders know how to detour these defense tools; as a result, a second line of defense is required, which is constituted by tools such as intrusion detection system and antivirus software (Lee *et al.* (2009).

As shown by Landwehr *et al.* (1994), our network system suffers from various security vulnerabilities, which activate to deny, disrupt, degrade and destroy services

and information resident in the network system. The primary aim of the network attack was to compromise the integrity, availability or confidentiality of the network system that is done through the data stream on a computer network by an intruder. Therefore, Intrusion detection system is intended to detect malicious or unauthorized activities on the network and inform the administrator to take appropriate action to prevent the system from further damages. IDS first analyzes all the network traffic and raise an alarm to assist the network administrator if malicious attempts are found.

1.1.1 Intrusion methods

Various intrusion methods may be a system, physical or remote intrusion. An unauthorized intrusion attempt may get into our system based on the flaws of Operating System design, TCP/IP protocol, and unsecure network. Intruders may also utilize the bugs on software implementation, buffer overflow, combinations of unexpected commands, lack of system configuration or administrator attention and unhandled key in sequences. Honeypots and establishment of holes, flaw in design, cracking of password, sniffing or probing over unsecure network traffic, etc., may assist intruder to find vulnerability of our expensive system.

1.1.2 Intrusion Detection System

Amorso (1999) discussed that intrusion detection system is the process of identifying and responding to malicious activity targeted at computing and networking resources. An IDS is designed to monitors network activity to identify malicious events. According to Schneier (2000), IDS functions in three stages namely, prevention, detection, and reaction. So, numerous techniques and controls are frequently adopted to prevent the network system from unauthorized and malicious attacks by implementing a firewall, antivirus, etc. If the intrusion penetrates the network systems even after installing preventive software, IDS acts as a next line of protection for the system.

Bishop (2004) stated that IDS aim to help filter out various potential unauthorized accesses to the computer network system based on the three essential pillars of information security, i.e., confidentiality, integrity and availability of resources. According to Beauquier *et al.* (2008), IDS gathers and analyze information data from various sources within the computer network, trigger alarm to a system administrator and blocks unauthorized access if an attack attempt is found.

1.1.3 Types of IDS

IDSs are broadly classified into two broad categories based on deployment.

1.1.3.1 Host-based IDS

A Host-based IDS (HIDS) is designed for a particular computer system and resides in an individual computer system. It is mainly deployed on the main server system called host, which examines the server activities only. Therefore, the HIDS is in use to monitor the system log files, operating system audit trails, stored configuration files and detect the creation, process log, alteration and deletion of system files attempted by the intruders. Host-based intrusion detection system can identify local events as well as an intrusion that have not been identified by the NIDS. The pattern of HIDS resides on the single host only and need more supervision effort to configure and install in multiple systems. HIDS are more exposed to direct intrusion and vulnerable to various Denial of Service (DoS) intrusion.

An upgraded version of HIDS called Application based IDS (AIDS) identifies an application for intrusive activity by analyzing the files created in implementation and anomaly activities such as beyond the users' permission, execution of the free file, etc. Therefore, an AIDS notice the interface between the application and the user so that they can analyze the encrypted network packets. However, AIDS is more powerless to intrusion and does not have power over to identify the tampering of application.

Table 1.1 demonstrates detail comparison of HIDS and NIDS.

Table 1.1: Detail comparison of HIDS and NIDS (Magalhaes, 2003).

Function	HIDS	NIDS
Protection on LAN	protect your Host	protect your LAN
Protection off LAN	HIDS protects off the LAN only	In LAN only
Machine registry scans	Yes	No
Versatility	more versatile systems	Less versatile systems
Ease of Implementation	Easy to Implement	Easy to Implement
Training required	requires less training	More training required
Total cost of ownership	Less in long run	High
Bandwidth requirements on LAN	No	NIDS require LAN bandwidth.
Network overhead	No Overhead	Double the total network bandwidth require for LAN
Spanning port switching requirements	Not required	Port spanning to be enabled for LAN traffic is scanned.
Update frequency to clients	Can update all clients from a central file.	No
Cross platform compatibility	Specific to OS, application	Adaptable to cross-platform environments.
Logging	Log the data	Log the data
Alarm functions	Alarm the individual or the administrator.	Alarm the individual or the administrator.
Packet rejection	No	Reject or drop packets.
Specialist knowledge	Only the application specific Knowledge	Knowledge is required for installing & understanding of a network security
Central management	Specific to the Host, with less central management.	Centrally managed.
Disable risk factor	Failure rate is very low	The failure rate is higher as one point of failure.

1.1.3.2 Network-based IDS

A Network based IDS (NIDS) is an inactive tool that resides in the system or the network system of an organization and observes the in and out network traffic for the

sign of intrusive activity. NIDS detect attacks by capturing and analyzing incoming and outgoing network traffic. If NIDS identifies any intrusive attempt, it raises an alarm immediately and notifies such malicious attempt to system administrators to take appropriate actions. To observe the traffic going into and out of the network system, an NIDS can be deployed in the router boundary. Without disturbing any of the normal operations of the network system, the minimum quantity of monitoring units for a huge network can be deployed. Network intrusion detection system is also not susceptible to direct intrusion attempt. However, it can turn out to be overwhelmed by network traffic, powerless to identify encrypted packets and fail to distinguish various intrusion activities.

1.1.4 Detection approaches

Based on the detection method, intrusion detection system are broadly categorized into three categories.

1.1.4.1 Misuse based detection

Misuse based detection relies on pattern matching techniques, containing a database of signatures of known attacks and tried to match these signatures against the analyzed data. In misuse method the observed behaviors were compared against the predefined attack signature, an alarm is raised if a match is found.

While misuse detection is fully effective in uncovering known attacks, it is powerless when faced with an unknown or new form of attack until and unless the signature for that novel attack are available. This issue is generally due to obsolete signature or absence of attack signature, as results, those attack activities will be undetected and classified as a false negative. Any mistakes in the definition of the signatures will increase the false alarm rate that will also decrease the effectiveness of the detection technique.

According to Lazarevic *et al.* (2003), most of the detection techniques employed by IDS are Signature Based, which try to search for patterns or signatures of the already known attacks. The advantage of such kind of system is that signatures can be developed for known attacks and that are faster compared to anomaly based detection. However, due to its limitation over network packet overload, expensive computational power on signature matching and massive false negative rate, research has been carried towards anomaly detection technique. Table 1.2 illustrate a comparison between misuse and anomaly detection.

Table 1.2: Comparison between misuse and anomaly detection based on the strength and weakness (Zanero, 2007).

Misuse based detection	Anomaly based detection
Continuous updates required	Not required
Initial training not required	Requires extended and complex training
Requires tuning and alteration	Tuning incorporated in training process
Unable to identify new or novel attacks	Able to identify new or novel attacks
Generate accurate alarm	Generate indistinct alarm
More or less no false positives rate	Massive number of false positives rate
Simple design	More complicated design

1.1.4.2 Anomaly based detection

Anomaly based detection first built a statistical model describing the normal network traffic that defines the normal baseline profile model and then flags any behavior that significantly deviates from the model. Anomaly detection approaches the problem by attempting to find deviations from the established baseline normal profile model against the analyzed data, which gave the anomaly detection ability to detect new types of attacks. However, as discussed in McHugh (2000), it may also cause a significant number of false alarms because the normal behavior varies widely and obtaining a complete description of normal behavior is often difficult.

The anomaly method tried to detect abnormal activity from the normal user behavior, if the observed behavior deviate too much from the normal baseline profile,

an alarm was triggered to the system administrator (Beauquier *et al.*, 2008). However, these detection systems still suffered from some limitations. Such as, ineffective detection rate towards known attack and massive false positive rate, which misclassify legitimate network traffic as an abnormal activity and raise some false positive to the administrator with an annoying alarm and remain ignorant of the real intrusion attempt (Anderson, 2001).

1.1.4.3 Hybrid detection

To resolve the disadvantages of these two conventional IDS techniques, a hybrid intrusion detection system combining both misuse and anomaly technique have also been proposed by recent research (Depren *et al.*, 2005; Kim *et al.*, 2014). In a hybrid technique, both misuse and anomaly methods are combined in such a way to ameliorate the performance of detection accuracy along with a lower degree of false alarm. The performance of the detection depends on the combination method of these two conventional detection techniques. Most hybrid IDS train both an anomaly and misuse detection technique independently based on parallel/serial and then calculates the weighted average results of the detection technique (Depren *et al.*, 2005). These techniques obviously increase the detection rate but still have a high degree of false alarm. To overcome this situation Kim *et al.* (2014) proposed a new method that integrate misuse and anomaly technique but still have high time complexity due to the absence of features selection.

1.2 Analysis of intrusion dataset

A systematical classification of patterns among the collected data is called data analysis which describes them to a distinct problem. Examining, transformation and modeling of information and deciding how to categorize, organize, interrelate, compare and display are the important process of data analysis. In every dataset, the reliability and correctness of data gathered and utilization in an evaluation determine the quality of

data. Data quantity deals with the quantity of information gathered for the evaluation.

The task of research requires a range of specific databases in its area, and the experimentation must be accomplished efficiently if the quality and features of data for the particular field are excellent. In the following section, detail analysis of the benchmark KDD-Cup'99 NIDS dataset and its features for the classification analysis of network traffic is shown.

1.2.1 KDD Cup'99 dataset

KDD Cup is the leading Data Mining and Knowledge Discovery competition in the world, organized by Association of Computing Machinery (ACM) Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) (http://www.sigkdd.org/kddcup) and is the most leading professional organization of data miners. The ACM SIGKDD organized the annual Data Mining and Knowledge Discovery challenges called KDD Cup which provides a competitive platform for various researchers around the world. Each year this annual competition focused on various areas. Table 1.3 demonstrates various KDD Cup center and their focuses area from 1997-2015.

KDD'99 is widely used as one of the few openly accessible intrusion datasets for NIDS. Stolfo *et al.* (2000) prepares this dataset and is based on the evaluation IDS evaluation program data captured by the Defense Advanced Research Projects Agency 1998 (DARPA) (Lippmann *et al.*, 2000). The Lincoln Lab. at MIT generated the standard network traffic data for NIDS evaluation program called DARPA'98, which was jointly sponsored by DARPA and the US Air Force Research Laboratory. They operated the LAN as if it were a true Air Force environment, but subjected it with multiple attacks. DARPA98 is about 4 GB of TCP dump raw data of about 5 million connections collected from 7 weeks of network traffic records of training sets and 2 weeks records of the testset data having around 2 million network traffic records. For each TCP/IP connection, 41 quantitative and qualitative features were extracted and

Table 1.3: Annual KDD Cup center with the focused area.

Year	Focused Area
1997	Direct marketing for lift curve optimization
1998	Direct marketing for profit optimization
1999	Computer network intrusion detection
2000	Online retailer website clickstream analysis
2001	Molecular Bioactivity and Protein locale prediction
2002	Bio Medical document and Gene role classification
2003	Network mining and usage log analysis
2004	Particle physics; plus Protein homology prediction
2005	Internet user searches query categorization
2006	Pulmonary embolisms detection from image data
2007	Consumer recommendations
2008	Breast cancer
2009	Fast scoring on a large database
2010	Educational Data Mining
2011	Music Recommendation
2012	Predict the click-through rate of ads given the query and user information
2013	Author-Paper identification
2014	Predicting Excitement at DonorsChoose.org
2015	Predicting dropouts in Massively-Online Open Courses (MOOC)

labeled as either normal or attack, with a specific type of attack. Data distribution of KDD'99 is demonstrated in Figure 1.1. Table 1.4 describes the types of attack found in KDD'99 dataset. The simulated attacks are classified into four major categories as below:

- Denial of Service (DoS): A DoS attack is a type of attack in which the intruder objectives is to block normally authorized access to services offered by a host or a network. The main aim is to exploit memory resources exhaustively and prevent serving legitimate network requests, and hence denying users access to a machine or a network. e.g., smurf, neptune, back, etc.

- Remote to Local (R2L): A remote to local attack is an attack aiming at gaining access to a local account from another host or network. In this type of attack, the user sends packets to a machine over the internet, and the user does not have access to expose the machine vulnerabilities and exploit privileges that a local

user would have on the computer, e.g., ftp_write, phf, multihop, etc.

- User to Root (U2R): These attacks are exploitations in which the intruder starts off on the system with a limited user account or normal user privileges and attempts to abuse vulnerabilities in the system to gain root access (system administrator privilege), e.g., perl, rootkit, etc.

- Probe: Probe is an attack in which the hacker scans a machine or a network to gather information or to find known vulnerabilities. The goal of this information gathering is to find out about computer and services that are present in a network with known vulnerabilities that may later be exploited so as to compromise the system in future, e.g., satan, portsweep, nmap, etc.

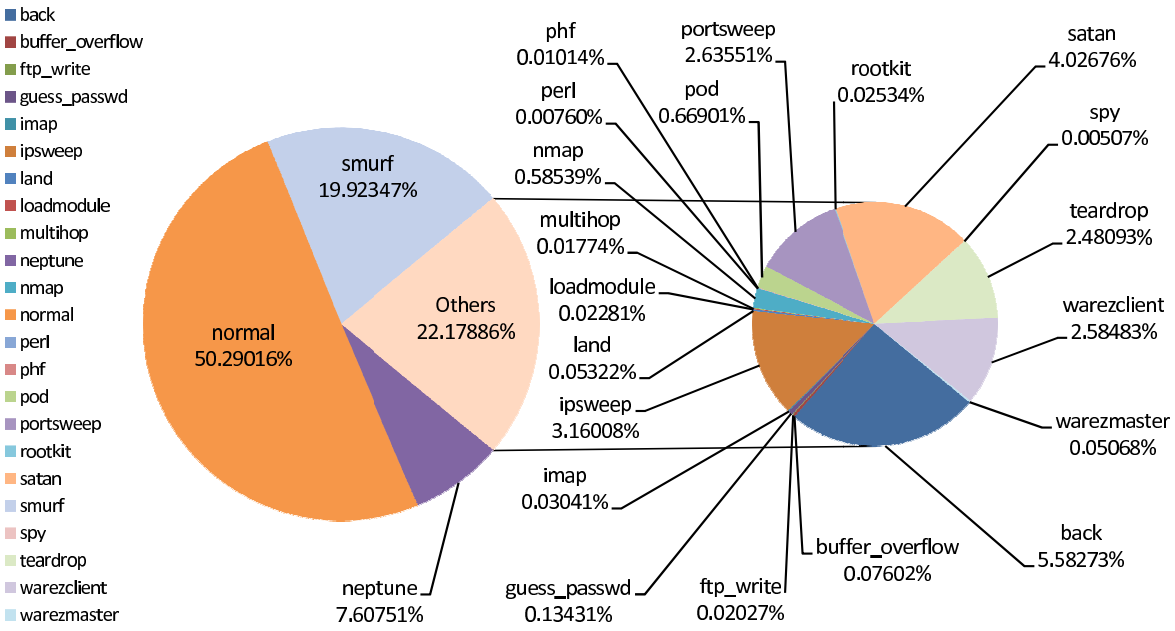

Figure 1.1: Data distribution in KDD'99.

The KDD'99 contains a huge number of repeated records of 78% and 75% redundant data on training and test dataset. The redundant dataset can harm the result of the evaluation to a much higher degree of detection accuracy. The necessary adjustment made on KDD'99 dataset by Tavallaee *et al.* (2009) results in a new dataset called

NSL-KDD. Tables 1.5, 1.6 & 1.7 illustrate the detail modifications made between KDD'99 with attack name and types of attack found in NSL-KDD. NSL-KDD dataset is not perfect and still suffers from the issue criticized by McHugh (2000) due to the use of synthetic simulation of normal with scripted anomaly data that can hamper the evaluation results. However, study and evaluation results by Thomas *et al.* (2008) sup-

Table 1.4: Descriptions of attack found in KDD'99 dataset.

Attack name	Attack type	Method	Effect of the attack
back	DoS	Abuse/Bug	Slows down server response
land	DoS	Bug	Slows down server response
neptune	DoS	Abuse	Slows down server response
smurf	DoS	Abuse	Slows down the network
pod	DoS	Abuse	Slows down server response
teardrop	DoS	Bug	Reboots the machine
loadmodule	U2R	Poor environment sanitation	Gains root shell
buffer_overflow	U2R	Abuse	Gains root shell
rootkit	U2R	Abuse	Gains root shell
perl	U2R	Poor environment sanitation	Gains root shell
phf	R2L	Bug	Executes commands as root
guess_passwd	R2L	Login misconfiguration	Gains user access
warezmaster	R2L	Abuse	Gains user access
imap	R2L	Bug	Gains root access
multihop	R2L	Abuse	Gains root access
ftp_write	R2L	Misconfiguration	Gains user access
spy	R2L	Abuse	Gains user access
warezclient	R2L	Abuse	Gains user access
satan	Probe	Abuse of feature	Looks for known vulnerabilities
nmap	Probe	Abuse of feature	Identifies active ports on a machine
portsweep	Probe	Abuse of feature	Identifies active ports on a machine
ipsweep	Probe	Abuse of feature	Identifies active machines

port KDD'99 as a benchmark dataset for research in NIDS environment. However, as our main effort is on a hybrid approach of NIDS based on the combinations of misuse and anomaly techniques, KDD'99 can still be used as a test bed dataset for carrying out various experiments on NIDS.

Table 1.5: Redundant records found in KDD'99 training dataset.

	Normal	Anomaly	Total
Original Records	972,781	3,925,650	4,898,431
Distinct Records	812,814	262,178	1,074,992
Reduction Rate	16.44%	93.32%	78.05%

Table 1.6: Redundant records found in KDD'99 test data.

	Normal	Anomaly	Total
Original Records	60,591	250,436	311,027
Distinct Records	47,911	29,378	77,289
Reduction Rate	20.92%	88.26%	75.15%

Table 1.7: Four attack types with corresponding attack name in NSL-KDD datasets.

Attack Type	Attack Name
Denial of Service (DoS)	back, land, neptune, pod, smurf,teardrop.
Remote to Local (R2L)	guess_password, ftp_write, imap, phf, multihop, warezmaster, warezclient, spy.
User to Root (U2R)	buffer_overflow, loadmodule, perl, rootkit.
Probing	satan, ipsweep, nmap, portsweep.

1.2.2 KDD Cup'99 features

The KDD Cup'99 dataset consists of 41 features, according to Lee *et al.* (1999) and Tavallaee *et al.* (2009), the features are group into three broad categories:

- Basic features: This category includes features 1-9 and is used to represent the basic characteristics of the network packet, they encapsulate all the features that can be extracted from a TCP/IP connection stream.

- Content features: This category employs the features 10-22 that contain information to be able to look for suspicious behavior in the data portion. The R2L and U2R intrusions do not have any consequence on attack frequent sequential patterns, and this is because the DoS or Probing attack engage a number of relations to various host(s) in a very short time. However, the U2R and R2L intrusion are implanted in the data portions of the network packets and usually involve single connection only.

- Traffic features: This category includes features 23-41 that are computed based on a two seconds time window. Features 23-31 are named as used for traffic features with two seconds of the time window that have the same service as the current connection and, features 32-41 examine for same destination host features as the current connection.

Detail description of all the 41 features is shown in the following Table 1.8.

Table 1.8: Detail descriptions of KDD'99 dataset features.

Label	Feature Name	Category	Type	Description
f1	duration	1	C	Number of seconds of the connection
f2	protocol_type	1	N	Type of the protocol, e.g., TCP, UDP, etc.
f3	service	1	N	Network service on the destination, e.g., HTTP, telnet, etc.
f4	flag	1	N	Normal or error status of the connection
f5	src_bytes	1	C	Number of data bytes from source to destination
f6	dst_bytes	1	C	Number of data bytes from destination to source

– Continued on next page

Table 1.8 – continued from previous page

Label	Feature Name	Category	Type	Description
f7	land	1	N	1-connection is from/to the same host/port; 0-otherwise
f8	wrong_fragment	1	C	Number of 'wrong' fragments
f9	urgent	1	C	Number of urgent packets
f10	hot	2	C	The count of access to system directories, creation and execution of programs
f11	num_failed_logins	2	C	Number of failed login attempts
f12	logged_in	2	N	1 - successfully logged in; 0 -otherwise
f13	num_compromised	2	C	Number of "compromised" conditions
f14	root_shell	2	C	1 - root shell is obtained; 0 -otherwise
f15	su_attempted	2	C	1 - 'su root' command attempted; 0 - otherwise
f16	num_root	2	C	number of 'root' accesses
f17	num_file_creations	2	C	Number of file creation operations
f18	num_shells	2	C	Number of shell prompts
f19	num_access_files	2	C	Number of writes, delete, and create operations on access control files
f20	num_outbound_cmds	2	C	Number of outbound Commands in an FTP session
f21	is_hot_login	2	N	1 - the login belongs to the 'hot' list (e.g., root, adm, etc.) ; 0 -otherwise
f22	is_guest_login	2	N	1 - the login is a 'guest' login (e.g., guest, anonymous, etc.) ; 0 - otherwise
f23	count	3	C	Number of connections to the same host as the current connection in the past 2 seconds
f24	srv_count	3	C	Number of connections to the same service as the current connection in the past 2 seconds

– Continued on next page

Table 1.8 – continued from previous page

Label	Feature Name	Category	Type	Description
f25	serror_rate	3	C	% of connections that have 'SYN' errors to the same host
f26	srv_serror_rate	3	C	% of connections that have 'SYN' errors to the same service
f27	rerror_rate	3	C	% of connections that have 'REJ' errors to the same host
f28	srv_rerror_rate	3	C	% of connections that have 'REJ' errors to the same service
f29	same_srv_rate	3	C	% of connections to the same service and the same host
f30	diff_srv_rate	3	C	% of connections to different services and the same host
f31	srv_diff_host_rate	3	C	% of connections to the same service and different hosts
f32	dst_host_count	3	C	Number of connections to the same host to the destination host as the current connection in the past 2 seconds
f33	dst_host_srv_count	3	C	Number of connections from the same service to the destination host as the current connection in the past 2 seconds
f34	dst_host_same_srv _rate	3	C	% of connections from the same service to the destination host
f35	dst_host_diff_srv _rate	3	C	% of connections from the different services to the destination host
f36	dst_host_same_src _port_rate	3	C	% of connections from the port services to the destination host
f37	dst_host_srv_diff_ host_rate	3	C	% of connections from the different hosts from the same service to destination host

– Continued on next page

Table 1.8 – continued from previous page

Label	Feature Name	Category	Type	Description
f38	dst_host_serror _rate	3	C	% of connections that have 'SYN' errors to the same host to the destination host
f39	dst_host_srv_ serror_rate	3	C	% of connections that have 'SYN' errors from the same service to the destination host
f40	dst_host_rerror _rate	3	C	% of connections that have 'REJ' errors from the same host to the destination host
f41	dst_host_srv_ rerror_rate	3	C	% of connections that have 'REJ' errors from the same service to the destination host

The attributes/features are labeled as f1, f2,..., f41 and the data types of each feature is represented as Continuous 'C' or Nominal 'N' (which are Discrete values) respectively.

The label f2 (protocol_type), f3 (service), f4 (flag), f7 (land), f12 (logged_in), f21 (is_hot_login) and f22 (is_guest_login) are labeled as nominal or discrete features. The other 34 feature out of 41 i.e, label f1 (duration), f5 (src_bytes), f6 (dst_bytes), f8 (wrong_fragment), f9 (urgent), f10 (hot), f11 (num_failed_logins), f13 (num_ compromised), f14 (root_shell), f15 su_attempted), f16 (num_root), f17 (num_file_creations), f18 (num_shells), f19 (num_access_files), f20 (num_outbound_cmds), f23 (count), f24 (srv_ count), f25 (serror_rate), f26 (srv_serror_rate), f27 (rerror_rate), f28 (srv_rerror_rate), f29 (same _srv_rate), f30 (diff_srv_rate), f31 (srv_diff_host_rate), f32 (dst_host_count), f33 (dst_host_srv_count), f34 (dst_host_same_srv_rate), f35 (dst_host_diff_srv_rate), f36 (dst_host_same_src_port_rate), f37 (dst_ host_srv_diff_host_rate), f38 (dst_host_serror _rate), f39 (dst_host_srv_serror_rate), f40 (dst_host_rerror_ rate) and f41 (dst_host_srv_ rerror_rate) are labeled as continuous type features. The features with nominal values, lebeled with f2 (protocol_type), f3 (service) and f4 (flag) called categorical features, and contain special values that are listed in Table 1.9.

The feature f2 (protocol_type) has 3 unique values of 'tcp', 'udp' and 'icmp' respectively. Similarly, the feature label f3 (service) have 70 unique values starting from 'aol' up to 'Z39_50'and label f4 (flag) feature gives distinct values of 11. Detail descriptions of the special feature 'f4 (flag)' values are shown in Table 1.10. This 3 characteristic of features and their unique values obtain an important position to build grammars in the proposed system.

Table 1.9: Different special values of protocol, service and flag.

Protocol(f2)	Label	Service(f3)	Label	Service(f3)	Label	Service(f3)	Label
tcp	1	aol	1	http_8001	25	red_i	49
udp	2	auth	2	imap4	26	remote_job	50
icmp	3	bgp	3	IRC	27	rje	51
Flag	*Label*	courier	4	iso_tsap	28	shell	52
(f4)		csnet_ns	5	klogin	29	smtp	53
OTH	1	ctf	6	kshell	30	sql_net	54
REJ	2	daytime	7	ldap	31	ssh	55
RSTO	3	discard	8	link	32	sunrpc	56
RSTOS0	4	domain	9	login	33	supdup	57
RSTR	5	domain_u	10	mtp	34	systat	58
S0	6	echo	11	name	35	telnet	59
S1	7	eco_i	12	netbios_dgm	36	tftp_u	60
S2	8	ecr_i	13	netbios_ns	37	tim_i	61
S3	9	efs	14	netbios_ssn	38	time	62
SF	10	exec	15	netstat	39	urh_i	63
SH	11	finger	16	nnsp	40	urp_i	64
		ftp	17	nntp	41	uucp	65
		ftp_data	18	ntp_u	42	uucp_path	66
		gopher	19	other	43	vmnet	67
		harvest	20	pm_dump	44	whois	68
		hostnames	21	pop_2	45	X11	69
		http	22	pop_3	46	Z39_50	70
		http_2784	23	printer	47		
		http_443	24	private	48		

Table 1.10: Detail descriptions of feature f4 (flag) value.

Flag	Description
RSTOS0	Originator sent a SYN followed by a RST, never see a SYN ACK from the responder
RSTR	Established, responder aborted
RSTO	Connection established, originator aborted (sent a RST)
OTH	No SYN seen, just midstream traffic (a "partial connection" that was not later closed)
REJ	Connection attempt rejected
S0	Connection attempt seen, no reply
S1	Connection established, not terminated
S2	Connection established and close attempt by originator seen (but no reply from responder)
S3	Connection established and close attempt by responder seen (but no reply from originator)
SF	Normal establishment and termination
SH	Originator sent a SYN followed by a FIN (finish 'flag') , never saw a SYN ACK from the responder (hence the connection was "half open)

1.3 Review of Literatures

Over the past decades, researchers have studied and proposed various methods of both hybrid and individual classification techniques. Various studies used Data Mining and Machine Learning (ML) technique to decrease the high degree of human activity in NIDS. As stated in Depren *et al.* (2005), Peng *et al.* (2006) and Gorbani *et al.* (2010), the misuse detection technique also known as signature detection technique detects the known attack and the anomaly detection technique detects the unknown attack.

In Depren *et al.* (2005), the author applied two level detection technique based on anomaly module, misuse detection technique and decision support module. The anomaly detection module is constructed based on Self-Organizing Map (SOM) on the normal profile and any deviation from this normal baseline profile is treated as attack. The misuse module was created based on the Decision Tree algorithm to classify various types of attack. The last module called decision support system is designed to simply

combine the results from each misuse and anomaly module, the three predefined rules in decision support module classify whether the encountered instance is an attack or normal activity. The proposed system was evaluated on 1999 KDDCUP intrusion dataset using first basic six features, the experimental results demonstrated that the proposed hybrid technique gave better performance over individual classification technique.

Two-stage manner hybrid intrusion detection is proposed in Peng *et al.* (2006), the proposed hybrid detection and visualization system leveraged the advantages of both signature and anomaly based detection technique and claimed that the proposed hybrid technique could identify both known and unknown attack on standalone host system calls. The first module called misuse was designed to handle the known attack. The second stage, anomaly detection stage was used to overcome the shortcoming of the first phase and can detect a novel attack. However, experimental environment and evaluation results for the proposed system were missed in the paper. The hybrid system framework gives an introduction on how to apply multiple classifications to improve the detection accuracy with an acceptable degree of false alarm rate of IDS.

There is some hybrid detection technique that combined the advantages of both misuse and anomaly detection technologies. Misuse based is well-known for its low false positive rate while anomaly based technique can detect unknown attack. The primary aim of such hybrid detection technique is to improve the anomaly detection technique in terms of decreasing the high false alarm rate with an acceptable detection rate, e.g., Kim *et al.* (2014) and Yousef *et al.* (2014). In Kim *et al.* (2014), the author proposed a new hybrid intrusion detection system that hierarchically integrates a misuse and anomaly detection model and claimed that it is the first attempt to use misuse detection model to enhance the performance of the anomaly detection model. First the misuse detection model was created based on DT to decompose the normal training data into smaller subsets using normal and attack traffic data. Then, anomaly detection model based on multiple one-class SVM was created in each decomposed

region. However, improvement is required as the misuse detection module degrades its performance due to subset decomposition problem and the absence of feature selection that results to higher computational time.

The author Yousef *et al.* (2014) proposed a Netflow based hybrid intrusion detection using two neural network (NN) stages classification system on a high-speed network. The first phase, called anomaly detection phase based on Levenberg-Marquardt NN classify whether flows are normal or abnormal. Then, the second module, called detection and classification phase based on radial basic function (RBF) NN detects the packets and their classification with a larger number of inputs or one of the four main attack types (Dos, port scan, land and other/unknown attack). The author claimed that the proposed system outperforms other existing techniques by comparing conventional model detection results. However, the selection of a different feature in both stages could harm compatibility for further classifications and time complexity.

Some other hybrid detection technique fuses multiple anomaly detection approach instead of fusing misuse and anomaly technique, e.g., Panda *et al.* (2012); Feng *et al.* (2014). The main goal is to reduce a large number of false positive generated by conventional anomaly technique. In (Panda *et al.*, 2012), the author proposed a novel hybrid detection technique based on END (Ensembles of balance nested dichotomies for multiclass problems) with nested dichotomies and random forest (RF). The evaluation results demonstrated that the proposed system outperforms various novel hybrid IDS regarding two class classification strategies. Feng *et al.* (2014) introduced a new ML-based classification algorithm for network intrusion detection. The fundamental idea is to classify network activities into normal or abnormal while minimizing the misclassification rate. The proposed new method combines the SVM classifier with Clustering based on Self-Organized Ant Colony Network (CSOACN) to take the advantages of both while avoiding their weaknesses. Evaluation results demonstrate that the proposed system outperforms individual classification of SVM regarding both clas-

sification rate and run-time efficiency.

The conventional anomaly detection technique has been criticized for its normal profile construction, development of comprehensive profile degrades detection rate while narrow profile results to a high degree of false alarm. To resolve this problem, Kim *et al.* (2014) firstly proposed a hybrid detection technique integrating misuse based with anomaly based. Their evaluation results produced an acceptable detection rate of 99.1% for known attack and 30.5% for unknown attack with 1.2% false positive based on misuse module. However, the evaluation results in terms of detection rate and false positive were not explicitly mentioned in the anomaly module, only the training and testing time were compared with conventional method. So, to resolve the limitation, section 5 concentrates on misuse detection improvement by creating more subset decomposition and feature extraction based on the effect of relevant features to improve the computational time.

Several studies proposed various methods for investigating the effectiveness of different ML algorithms to improve the performance of intrusion detection system. Most of the IDS studies are evaluated based on the KDD'99 CUP (1999) Intrusion dataset. However, as discussed in Tavallaee *et al.* (2009), the KDD'99 contains an enormous number of redundant instances. After removing unnecessary data, authors proposed refined versions of dataset called NSL-KDD dataset that characterized much more consistent environment for various algorithms than the original KDD'99 dataset.

As stated by Giray and Polat (2013), majority of the IDS studies used anomaly based detection technique. Most of the proposed anomaly based detection method in the past concentrates on the detection performance and the algorithm, instead of the effectiveness of that model over a noisy environment and have appeared in the literature, e.g., Manikopoulos and Papavassiliou (2002), Mukkamala *et al.* (2002), Mukkamala *et al.* (2004) ,Chang *et al.* (2010), Wang *et al.* (2010) and Panda *et al.* (2012) where various types of ML algorithm for anomaly detection technique which are

given with comparative evaluation results with noise free datasets. Therefor, section 2 of our thesis deals with evaluation and comparison of various ML algorithms based on noise-free and noisy datasets which is a challenging issue.

In recent decades, various studies have used Machine Learning and data mining technique to remove high degree of human interaction in IDS. Many studies have focused on improving the detection accuracy by proposing a new classifier for IDS; but improving the effectiveness of the existing classifier is a difficult task, and therefore, as stated in Adetunmbi *et al.* (2010), researchers have used feature selection to optimize the existing classifier. The elimination of unimportant features/attributes is the main task of feature selection method, which reduces computational complexity as well. Amiri *et al.* (2011) experiment on three different models based on hybrid IDS by applying feature selection method over KDD'99 datasets and propose two feature selection algorithms. Evaluation function is done by Modified Mutual Information-based Feature Selection (MMIFS) method, Linear Correlation-based Feature Selection (LCFS) and Forward Feature Selection (FFS). The study also introduces an IDS that uses machine learning-based Least Square Support Vector Machine (LSSVM). Hassan *et al.* (2006) propose a hybrid base IDS for protecting network intrusion based on honeypot approach which consists of three main components, i.e., honeywall (equipped with Snort ids which is signature-based), the honeyds and honeynet. The honeyds are designed to emulate direct network traffic and are hosts to real physical honeypots where honeywall is designed to monitor and log each network traffic passing the honeynet. Lin *et al.* (2012) combine Support Vector Machine (SVM), Decision Tree (DT) and Simulated Annealing (SA) for anomaly intrusion detection system, claiming that SVM and SA can find the best selected features to increase the accuracy of anomaly intrusion detection over KDD'99 dataset, and DT with SA can obtain decision rule for new attacks which improves the accuracy of the classification.

As relevant feature selection is one of an important method to reduce the complexity

of an algorithm; several studies, e.g., Guan *et al.* (2003), Sung and Mukkamala (2003), Kayacik *et al.* (2005), Olusola *et al.* (2010), Lin *et al.* (2012) and Louvieris *et al.* (2013) proposed various methods of feature selection and extraction. Two types of relevant feature ranking method; performance-based and concrete feature ranking based are tested on SVM and ANN is used in Sung and Mukkamala (2003), and the author concluded that both features ranking methods heavily overlaps each other with features relevance. The use of essential features of each class gave the most remarkable performance. Guan *et al.* (2003) used the Y-means clustering method for IDS. Relevance feature analysis is done in Kayacik *et al.* (2005) that employe Information Gain based on the KDD'99 dataset, the primary objective of this study is to analyze the performance of each feature involvement to machine learning algorithms that is trained on the KDD'99 dataset. Olusola *et al.* (2010) introduced relevance feature selection based on the application of Rough Set. Authors statistically analyzed to find dependency and dependency ratio of each class based on KDD'99 dataset, to determine the most distinctive feature of each category, and the author concluded that 7 features out of 41 were not relevant or does not have a contribution in any way. Removal of such features can significantly improve the performance of ML concerning detection accuracy and search speed (Lin *et al.*, 2012). In Lin *et al.* (2012), the authors combine the advantage of SVM and Simulated Annealing (SA) to find the best feature and therefore, the performance of the DT can be elevated. Effect-based features identification method that combine k-means clustering algorithm, Nave bases feature selection and DT is used in Louvieris *et al.* (2013) to pinpoint cyber attack. The authors claimed that the proposed model improved the performance of NIDS concerning detection accuracy.

Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) is used by Neumann and Porras (1999), it a technique that uses past historical records to build a model and then compare the distribution of new data with the model.

EMERALD is designed to targets both the internal and external threat agents who try to misuse the system resources, the design also combined signature and statistical based analysis components to produce analysis results. Comparisons of nine different ML algorithms in Sabhnani and Serpen (2003), concluded that no algorithm can detect all attacks, means that every algorithm has its drawback. The parallel hybrid classification method proposed in Depren *et al.* (2005) combined Self-Organization Map (SOM) with a C4.5 classifier, SOM module was designed to model normal behavior, any deviation from the baseline model is treated as an intrusion. The C4.5 module designed as misuse detection simply classify those intrusion data into corresponding attack type, the final decision was made by designed module called Decision Support System (DSC). DSC analyze results from each module by simply adding the output and claimed as a hybrid classification, the experiment results in 99.9% detection accuracy along with 0.1% false alarm on KDD'99 datasets that contain numbers of redundant data. Most hybrid IDS system trained the designed model independently and then simply aggregates the results of the individual model for final results. Naive Bayes algorithm is used in Panda and Patra (2007) for anomaly based detection, employing 41 standard features from the KDD'99 dataset and achieved a detection rate of 95%. After removing 90% instances of the original datasets, the simulation results demonstrate that Naive Bayes outperform ANN based approach by resulting higher detection rate, consuming less time with a low-cost factor. Artificial Neural Networks with K-mean clustering algorithm is used in Faroun and Boukelif (2007) which results in detection accuracy of 92%, the method applied K-means algorithm to the training set to select an optimal set of samples and a multi-layered network with a Backpropagation mechanism classification. "Enhanced Support Vector Decision Function" for feature selection was used in Zaman and Karry (2009), based on two important factors, the feature's rank and the correlation between the features, experimental results show that the proposed algorithms deliver an acceptable outcome in classification accuracy, training and testing time. Combinations

of Support Vector Machine (SVM), Decision Tree (DT) and simulated annealing (SA) are introduced in Lin *et al.* (2012) for anomaly intrusion detection system. The author claimed that SVM and SA can find the best-selected features to increase the accuracy of anomaly intrusion detection over the KDD'99 dataset and DT with SA can obtain decision rule for new attacks that improve an accuracy of the classification. Recently, Chung and Wahid (2012) proposed intelligent dynamic swarm-based rough set feature selection with simplified swarm optimization showing 93.3% detection rate. Panda *et al.* (2012) compares six different ensemble methods experimented for NIDS on NSL-KDD datasets, and conclude that, the combinations of Principal Component Analysis (PCA) feature selection, in hybridization of Random Forest with nested dichotomies and Ensembles of Balance Nested Dichotomies (END) outperform other tested model with detection rate of 99.5% and 0.1% false positive. Artificial Bee Colony (ABC) was used for the first time in Bae *et al.* (2012) to solve the intrusion detection problems, a new network intrusion detection system based on ABC searching algorithm has been proposed and compared with five popular benchmarks classifier (Naives Bayes, SVM, Classification tree, K-NN and C4.5 classifier). The evaluation results are quite encouraging, but the individual anomaly classification technique still suffered from anomaly detection drawback, which is high false positive. So this study applied a hybrid two stages classification using Anomaly-Misuse technique to overcome the situation faced by the individual classification techniques.

Another hybrid design stated by Wozniak *et al.* (2014) is discussed by various studies, hybrid design using system topology is discussed in Rivest (1987), Clark and Niblett (1989), Freund and Schapire (1997), Fumera *et al.* (2004), Bartlett and Wegkamp (2008) and Termenon and Grana (2012). In particular, Fumera *et al.* (2004) investigate the usefulness of the reject option in text categorisation systems, to automatically handle rejections, a two-stage classifier architecture is used to improve reject option on a real text categorisation task. Termenon and Grana (2012) present a two-stage

sequential ensemble where a second stage classifier processes data samples whose outcome from the first stage classifier fall in a low confidence output interval (LCOI). Authors tested the model based on a database of feature vectors of Alzheimers disease (AD) and control subjects extracted from sMRI data and reveal improved result over previous results for this database.

Hybrid design based on ensemble method is discussed in Fleiss and Cuzick (1979), Krogh and Vedelsby (1995), Ueda and Nakano (1996), Partridge and Krzanowski (1997), Cunningham and Carney (2000), Tax and Duin (2001), Giacinto and Roli (2001a, 2001b), Shipp and Kuncheva (2002), Tax and Duin (2002), Kuncheva *et al.* (2003), Wang *et al.* (2003), Wu *et al.* (2004), Didaci *et al.* (2005), Zhang and Jin (2006), Giacinto *et al.* (2008), Brown and Kuncheva (2010), Smetek and Trawinski (2011), Galar *et al.* (2011), Wilk and Wozniak (2012), Bi (2012) and Woloszynski *et al.* (2012). In particular, Smetek and Trawinski (2011) discussed the problem of model selection to compose a heterogeneous bagging ensemble, a three self-adapting genetic algorithms were proposed with different control parameters of mutation, crossover, and selection adjusted during the execution. The evaluation results revealed that the self-adaptive algorithms converged faster than the classic genetic algorithms. The heterogeneous ensembles created by self-adapting methods showed an excellent predictive accuracy when compared with the homogeneous ensembles obtained in earlier research. Wilk and Wozniak (2012) displays the possibilities of generalizing the two-class classification into multi-class classification using a fuzzy inference system, evaluated via computer experiments carried out on benchmark datasets revealed the effectiveness of the proposed method based on the fuzzy logic theory. Woloszynski *et al.* (2012) presented a measure of competence based on random classification (MCR) for classifier ensembles, two MCR based systems were developed and their performance was compared against six multiple classifier systems using data sets taken from the UCI Machine Learning Repository and Ludmila Kuncheva Collection and superiority of the

proposed model is revealed.

A hybrid classification design of IDS based on fusion method is discussed in Cheeseman *et al.* (1988), Shlien (1990), Jacobs *et al.* (1991), Jacobs (1995), Opitz and Shavlik (1996), Woods *et al.* (1997), Alexandre *et al.* (2000), VanErp *et al.* (2002), Lin *et al.* (2002), Kittler and Alkoot (2003), Rao (2004), Rokach and Maimon (2005), Zheng and Padmanabhan (2007), Biggio *et al.* (2007) and Wozniak and Zmyslony (2010). In particular, VanErp *et al.* (2002) discussed and tested several well-known voting methods from politics and economics on classifier combination to see if an alternative to the simple plurality vote exists. Tha author found that, assuming a number of prerequisites, better methods are available, that are comparatively simple and fast. Biggio *et al.* (2007) presented a new theoretical framework for the analysis of linear combiners that extends the scope of previous analytical models, and provides some new theoretical results which improve the understanding of linear combiners operation. Wozniak and Zmyslony (2010) discussed the problem of fuser design which uses discriminants of individual classifiers to make a decision, the main focus is on the fuser which uses weights dependent on classifier and class number. Finally, the author formulates the problem of fuser learning as an optimization task and propose a solver who has its origin in neural computations, evaluation based on several computer experiments on five benchmark datasets and their results confirm the quality of proposed concept.

Chapter 2

Feature analysis, evaluation and comparisons of classification algorithms based on noisy intrusion dataset[1]

2.1 Introduction

Various studies have been carried on an Intrusion Detection System environment to improve the existing model, by comparing the performance of various Machine Learning based on a refined intrusion dataset with an error-free environment. However, the real-world network data deals with a large amount of noisy information on transmission, and the IDS have to work in such an environment frequently. Dealing with such noisy data is, therefore, a challenging issue in an IDS environment for detecting threats from network activities.

In this section, various Data Mining (DM) and ML algorithms are evaluated and compared by normal and noisy dataset prepared from KDD'99 and NSL-KDD dataset (10%-20% Noise). The empirical results demonstrate that NN (SOM) is far better compared to other tested algorithms regarding robustness against noisy environment;

[1] ***Accepted for 2nd Int. Conf. on Intelligent Computing, Communication & Convergence 2016, to be published by Procedia Computer Science, Elsevier, 2016.***

however, JRip and J48 from the tree family outperform others regarding overall performance matrices. After selecting top six classifiers over noisy data, feature dependency on datasets for a particular classifier is analyzed by Performance-based Method of Ranking (PMR). The evaluation results statistically proved that each classifier has a unique combination of a feature subset to results optimal performance. Empirical results from KDD'99, NSL-KDD, 10% and 20% noisy datasets demonstrate that evaluations of IDS based on NSL-KDD give more realistic results compared to the KDD'99 original dataset.

The IDSs are designed to detect real threat at the exact time for lower false alarm rate and maximum detection accuracy. However, due to the occurrence of limitation in resources like computational power, memory and storage usage, IDS sometimes failed to detect abnormal activity at the exact time. To study and design an optimal solution to these limitations, recent research proposed various methods of individual and hybrid classification technique for IDSs. However, a majority of the studies used error free datasets to evaluate various proposed models, while the real world network traffic deals with frequent noisy information. Therefore, an evaluation result from an absence of noise is deceptive in the area of IDS, since classifier performs better in noise free environment. This study investigates and evaluates on various data mining algorithms to explore the performance of each classifier against various datasets, i.e., noise free, noisy (10% & 20%) environment. We have selected top six (6) best classifier among various tested classifier based on evaluation performance. Ranking of a significance of features based on performance is done for each selected classifier to study and compare various feature selection method used by other researchers.

2.2 Theory and algorithms

2.2.1 Dataset organization

In this study, four types of datasets prepared from KDD'99 (KDD Cup, 1999) and NSL-KDD (Tavallaee *et al.*, 2009) intrusion dataset are used to evaluate each classification algorithm. Details of the data preprocessed are as follows:

2.2.1.1 KDD'99 Cup Dataset:

This dataset is built and prepared by Stolfo *et al.* (2000) based on the data captured in DARPA'98 Intrusion Detection System Evaluation program (Lippmann *et al.*, 2000). Which was then widely used as a benchmark dataset in the field of network intrusion detection system studies. The dataset contain a TCP-dump raw data of about 5 million connections collected from 7 weeks of network traffic records of training sets and 2 weeks records of testset data having around 2 million network traffic records. For each TCP/IP connection, 41 quantitative and qualitative features were extracted. For evaluation, we used 10% of the original data. After folding the data onto 13 stratified folds, the first fold containing 39461 instances were used for final evaluation. Data distribution of KDD'99 is demonstrated in Chapter 1 figure 1.1. The KDD'99 still suffers from the issue criticized by McHugh (2000) due to the use of synthetic simulation of normal with scripted anomaly data that can hamper the evaluation results. However, studies and evaluation results support KDD'99 as a benchmark dataset for research in NIDS environment (Thomas *et al.*, 2008).

2.2.1.2 NSL-KDD Dataset:

Tavallaee *et al.* (2009) proposed the NSL-KDD dataset which is an enhanced edition of KDD'99 dataset created by the DARPA at the MIT Lincoln Laboratories USA. The KDD'99 dataset contains extensive records of redundant data, where 78% training

dataset and 75% test dataset are duplicate which may direct classifier algorithm unreasonable towards the other repeated records and therefore, avoid it against harmful networks attack group such as U2R and R2L category. Redundant data found on the test dataset can also affect the evaluation performance into a higher degree of detection accuracy on the repeated data. It was also declared that the NSL-KDD dataset is not perfect and still suffer from some problems criticized in McHugh (2000). However, it can still serve as a testbed dataset benchmark for carrying out various experiments on NIDS (Liu *et al.*, 1995). The refined dataset in KDDtrain+.txt and KDDtest+.txt are combined, all the attack traffic in a dataset is grouped into one class named as an anomaly. The ratio of normal and anomaly instances is maintained to meet the preprocessing requirement. After folding the data onto six (6) stratified folds, the first fold containing 27526 instances is used for evaluation.

2.2.1.3 Noisy dataset (10% & 20%):

Since this study focuses on evaluating the robustness of various data mining algorithms in a noisy environment, we used the NSL-KDD dataset for noise generation and added noisy data varying percentages of 10% and 20% to specific attributes using the KDD features. Noise is added to the specific feature after analyzing the dependency on the feature using NSL-KDD. To evaluate and analyze feature significance, Gain Ratio (Liu *et al.*, 1995) and Info Gain (Ganchev *et al.*, 2006), based on k-folds cross-validation technique is used. It is assumed that the noise is added randomly to a particular attribute label and are distributed evenly among the datasets. Every feature can have a noise characteristic, and a few can be cleaned or filtered. However, filtering may require more time complexity and cause delay undesirable for IDS. Besides, for some features, it is not safe to filter away the noise content. Therefore, while performing the model evaluation, performance assessment in the presence of noisy data becomes relevant for the IDS domain. The main objective of this study is to find whether the data mining

algorithm that performed well on the original data (Noise free environment) could perform well on the noisy data (real world data). Therefore, noise is added to selected features varying the percentage of 10% and 20%.

2.2.2 Algorithms

To accomplish the objectives of the study, we utilized the following set of some ML algorithm to evaluate the applicability and efficiency. A set of ML algorithms from various classifier families consisting of Naive Bayes (NB), Support Vector Machine (SVM), Gaussian Radial Basis Function Network (RBFN), Sequential Minimal Optimization (SMO), Radial Basis Function Classifier (RBFC), Stochastic Variant of Primal Estimated sub-Gradient Solver in SVM (SPegasos), Bayesian Network (BN), Voted Perceptron (VP), Stochastic Gradient Descent (SGD), JRip, J48, Random Forest, Ensembles of Balanced Nested Dichotomies for Multi-class Problems (END), NB-Tree and Artificial Neural Network (NN), Decision Table (DT) algorithms are evaluated and compared.

2.2.2.1 Bayesian Network (BN) & Naive Bayes (NB)

BN is a probabilistic method; representing random variable sets with their conditional dependencies using directed acyclic graph (Pearl, 1985). NB classifier is based on probabilistic method. It typically relies on assumption and assumes that variables are independent of each class or feature. More specifically, the presence of each particular class or feature is isolated from the absence or occurrence of some other features (George *et al.*, 1995). Depending on the precise nature of the probability model, Naive Bayes classifiers can be trained efficiently in a supervised learning approach. Nettleton *et al.* (2010) demonstrated the robustness of NB classifier to a noisy environment. The advantages of Naive Bayes is; fast to train (single scan), fast to classify, not sensitive to irrelevant features, handles real and discrete data, handles streaming data well.

However, the main disadvantage is it assumed independence of features.

2.2.2.2 Support Vector Machine (SVM)

The basic idea of SVM is to raise dimensions of the samples so that they can be in a separable form. The basic plan is to find a hyperplane to place samples of the same class inside it (Ghorbani *et al.*, 2010). Generally, linear boundaries such a polynomial in enlarged space achieve better training class separation and translate to nonlinear boundaries in the original space. Various recent IDS studies used SVM for its advantages, i.e., it produces very accurate classification result, less overfitting, robust to noise. However, it has disadvantages from another classifier, i.e., SVM is a binary classifier, to do a multi-class classification problem; pair-wise can be used (one class against all others, for all classes). SVM is computationally expensive, thus runs slower.

2.2.2.3 Artificial Neural Network (NN)

An artificial neural network (ANN) is adaptive parallel distributed information processing model. It consists of a set of simple processing units called neurons (a set of simple processing units), a set of synapses (connection), the network architecture (pattern of connectivity), and a learning process used to train the network (Ghorbani *et al.*, 2010). The processing element is the fundamental building block of ANN. They are responsible for all the computations that are taking place locally inside the network. Generally, the computations consist of multiplication, summation, and nonlinear mapping operations. Neurons are interconnected to each other via synapses. Each connection is a unidirectional link that takes care of the flow of information between two neurons. The strength and weakness of a connection are dependent on the local biochemical environment of that connection, which in turn is determined by the progressive modification through the course of the learning process. The artificial neural networks fall into two broad categories: (a) Feedforward networks, and (b) Recurrent networks. Learning is

the process of finding an optimal or near optimal pattern of connectivity for the neural network. The existing learning algorithms can be divided into two broad categories, i.e., Supervised (Labeled) and Unsupervised (Unlabeled) learning.

2.2.2.4 J48

It utilizes a divide-and-conquer approach and recursively create Decision Tree based on the greedy algorithm (Quinlan, 1986). It consists of the root node, branches, parent nodes, child nodes and leaf nodes. A node in a tree denotes dataset attributes; every child node derives labeled branches concerning the possibilities of attribute values from the corresponding node called parent node (Kim *et al.*, 2014). The advantages of decision tree methods are; J48 classifier are easy to understand, J48 are easily converted to a set of production rules, it can classify both categorical and numerical data, but the output attribute must be categorical and there are no a priori assumptions about the nature of the data in J48.

2.2.2.5 Sequential Minimal Optimization (SMO)

It is based on support vector machines that utilize optimize training method (Platt, 1998). There are two components in SMO: an analytic method for solving for the two Lagrange multipliers, and a heuristic for choosing which multipliers to optimize. The advantage of SMO lies in the fact that solving for two Lagrange multipliers can be done analytically. In addition, SMO requires no extra matrix storage at all. Thus, very large SVM training problems can fit inside of the memory of an ordinary personal computer or workstation. SMO is found to be more sensitive to noise compared to other algorithms (Nettleton *et al.*, 2010). We used inbuilt libraries in Weka for SMO.

2.2.2.6 Stochastic Variant of Primal Estimated sub-Gradient Solver in SVM (SPegasos)

Spegasos used and implemented the stochastic variation on the Pegasos technique of Shwartz *et al.* (2007). It embedded the structural information into the SVM and using the parallel computing framework 'MapReduce'. All missing values are replaced and transform nominal attributes into binary. All attributes are normalized with the output coefficients based on the normalized values. In this, hinge loss (SVM) is minimized for optimizing the performance of the proposed intrusion detection system. This algorithim can take full advantage of the computing and storage capacity of the computer cluster, and be applicable to the optimization problem of the massive data.

2.2.2.7 Voted Perceptron (VP)

It uses the perceptron algorithm to maps input against one of several feasible non-binary outputs (Freund and Schapire, 1999). The advantages of linearly separable data onto large margins are utilized and are considered to be robust to noisy data (Khardon and Wachman, 2007). The VP has some advantages in ease of implementation and efficiency. The VP algorithm is more stable numerically, as it does not need to compute a partition function and does not require parameter selection that can be costly in terms of run time.

2.2.2.8 Radial Basis Function Classifier (RBFC)

RBF classifier is types of feed-forward network. A general approach is to train the hidden layer of the network based on simple k-means clustering algorithm and the output layer based on supervised learning. However, Wettschereck and Dietterich (1992) found that supervised training technique on hidden layer parameters can elevate the performance of prediction. They investigated local variances of the basis functions, learning center locations and attribute weights in a supervised manner.

2.2.2.9 Ensembles of Balanced Nested Dichotomies for Multi-class Problems (END)

END is Meta classifier for managing multi-class data, having 2-class classification strategy by building an ensemble of nested dichotomies. A method of nested dichotomies is a hierarchical breakdown of a multi-class problem with c classes into $c-1$ two-class problems and can be represented as a tree structure. Ensembles of randomly generated nested dichotomies have proven to be an effective approach to multi-class learning problems. However, sampling trees by giving each tree equal probability means that the depth of a tree is limited only by the number of classes, and very unbalanced trees can negatively affect runtime. More details of this can be seen from (Dong *et al.*, 2005).

2.2.2.10 Stochastic Gradient Descent (SGD)

It uses gradient descent optimization technique by taking similar steps to find local minimum function of the negative gradient based on the objective function, written as a sum of differentiable functions (Bottou, 1998). The advantages of Stochastic Gradient Descent are its efficiency and ease of implementation. The disadvantages include: SGD requires a number of hyperparameters such as the regularization parameter and the number of iterations. It is sensitive to feature scaling.

2.2.2.11 JRip

It is based on the Repeated Incremental Pruning to Produce Error Reduction method. It integrates association rules with reduction error pruning. It divides the dataset into growing sets and pruning set, generating rules for a subset of the training samples and removes all samples covered by that rules for a training set on all samples (Cohen, 1995). JRip (RIPPER rule learner) is a fast algorithm for learning "IF-THEN" rules. Like decision trees rule learning algorithms are popular because the knowledge representation is extremely straightforward to interpret.

2.2.2.12 Random Forest (RF)

RF uses an ensemble technique of unpruned classification; succeed from training data onto the bootstrap samples, using random feature selection in a tree induction process. Final prediction is made based on aggregating the predictions output of the ensemble by majority voting for classification (Breiman, 2001). It suffered from noisy data and also from error like generalization error rate. The learning phase of RF also suffered from high imbalance training datasets. RF is designed to reduce the overall degree of error rate, trying to aim further on the accuracy predictions of the classifier on the popular class, which frequently affects the minority class resulting poor accuracy in classification.

2.2.2.13 Decision Table (DT)

It is one of the simplest possible hypothesis spaces and is simple enough to be understood. It consists of a hierarchical table where each entry to a higher level table gets broken down into more sub-tables based on the values of a pair of additional attributes forming another table (Kohavi, 1995). In DT, each decision corresponds to a variable, relation or predicate whose possible values are listed among the condition alternatives. Each action is a procedure or operation to perform, and the entries specify whether (or in what order) the action is to be performed for the set of condition alternatives the entry corresponds to. Many decision tables include in their condition alternatives the don't care symbol, a hyphen. Using don't care condition can simplify DT, especially when a given condition has little influence on the actions to be performed. In some cases, entire conditions thought to be important initially are found to be irrelevant when none of the conditions influence that actions are performed.

2.2.2.14 Naive Bayes (NB) Tree

NB-Tree is proposed by Kohavi (1996), that integrate a hybrid of a decision-tree classifier and Naive Bayes classification algorithm. It attempts to utilize the advantages of both Decision Trees and Naive Bayes regarding segmentation and evidence accumulation from multiple attributes. A Decision Tree is built based on univariate splits at every node and Naive Bayes classifier at each leaf. NB-Tree appears to be a viable approach to induce classifiers, where: Many attributes are relevant for classification; attributes are not necessarily independent; database is large; interpretability of classifier is important. In practice, NB-Trees are shown to scale to large databases and, in general, outperform Decision Trees and Naive Bayes individually.

2.2.2.15 Gaussian Radial Basis Function Network (RBFN)

RBF Network uses the simple k-means clustering method to give the basis functions and at the top it uses either a logistic regression or linear regression technique. From each cluster, Symmetric Multivariate Gaussians are fit to the data. It tends to use the given number of clusters per class if the class is nominal. All numeric attributes are standardized to zero mean and unit variance (Howlett and Lakhmi, 2001). The RBFN has significant advantages over multilayer perceptrons (MLP), namely faster convergence, smaller extrapolation errors, higher reliability, and a more well-developed theoretical analysis. Another advantage that is claimed is that the hidden layer is easier to interpret than the hidden layer in an MLP. Although the RBFN is quick to train, when training is finished, and it is being used, it is slower than an MLP, so where speed is a factor an MLP may be more appropriate.

2.3 Results and discussion

2.3.1 Experimental setup

Experimental environment is carried on Java environment using Weka 3.7.11 (Hall *et al.*, 2009) which provides inbuilt libraries containing various machines learning algorithm. The embedded data mining algorithm is used with default settings since most of the relevant studies in Weka used them too. The dataset mentioned in previous sections is preprocessed to meet the ARFF format supported by Weka. Selected 16 classifiers from various classifier family were tested based on k-fold cross-validation (K-FCV) technique within each dataset. K-FCV is one of the most common method where dataset gets divided into k, k represents the number of folds or subsets, k-1 subsets are used as training sets and k-(k-1) subset is used for the testing set. More specifically, each fold were analyzed, and the total score result determine the average performance out of k-folds.

Our study aimed at two main objectives. Firstly, various classification algorithm were evaluated based on the four datasets to analyze a statistical performance of each model to find the robustness of the analyzed algorithm. Each algorithm was evaluated based on KDD'99, NSL-KDD, 10% Noisy data and 20% Noisy data. Selection of first six classification algorithm is done based on the performance evaluation metrics. Secondly, dependency on each feature based on each classification algorithm was studied using Performance-based Method of Ranking (PMR). Each selected feature subset was evaluated on each six classification algorithm to explore the effectiveness of each feature selection within each classification algorithm.

Performance evaluation matrices for each simulation result were carefully monitored and measured based on Accuracy Rate (AC), True Positive Rate (TPR), False Positive Rate (FPR), Precision, ROC area, # of incorrectly classified rate, RMS error and time complexity, which is the key point to measure and determine reliability of an

IDS. Accuracy is the intensity of confidence in detecting intrusive activity. TPR is proportion of instances classified as a given class divided by the actual total in this category (equivalent to Recall). False positive are those normal activities in which the system used to identify as an intrusive attempt. Precision indicates the hit rate of the classification method detecting intrusive activity. Root Mean Square Error (RMSE) is the distinction between prediction and resultant observed values at each squared and then averaged over the sample. Performance of a classification algorithm is measured by Receiver Operating Characteristic (ROC or area under ROC curve), an area of 1 represents a 100% perfect test. Incorrectly classified rate is the rate of false alarm + false negative from the maximum instances, and time complexity is building time in seconds taken by each classifier to construct the model.

2.3.2 Results

Table 2.1 and Figures 2.1 - 2.4 demonstrate the overall performance of various classification algorithm with each dataset based on important parameters, i.e., TPR, FPR, Precision and ROC area based on four types of datasets. The evaluation results demonstrate that SGD, Jrip, J48, RF, END and NB-tree results 0.99 detection accuracy based on KDD'99 datasets and 0.98, 0.97 using SMO and RBF Classifier. However, Neural Network (SOM) results the lowest accuracy of 0.77, 0.678, 0.677 and 0.677 based on corresponding datasets (Table 2.1). Evaluation results derived from NSL-KDD, 10% & 20% noisy datasets results lower TPR and higher FPR for each evaluated classification algorithms, which was caused by the removal of redundant instances on NSL-KDD datasets and the existence of noisy information.

Anomaly based IDS generally suffered from the problem of tension between a false alarm and ignored attacks. Reductions of false alarm usually resulted in other ignored attack rates. So, balance ratios between false alarm rate and ignored attack rate is an important parameter to determine high-quality IDS. False alarm rate is the amount of

false positive generated for the abnormal activity. Figure 2.5. demonstrate false alarm rate of each evaluated classification algorithms based on each corresponding datasets. DT displays the lowest false alarm rate of 0.16% derived from KDD'99 dataset while VP results in an extremely high false alarm rate of 38.31% based on the KDD'99 dataset.

Table 2.1: Detection rate of Classification algorithm for four datasets.

Classifier	KDD99	NSL-KDD	10% Noise	20% Noise
Naive Bayes	0.926	0.862	0.833	0.834
SVM	0.958	0.929	0.918	0.907
RBF Network	0.918	0.879	0.868	0.846
SMO	0.982	0.962	0.93	0.913
RBF Classifier	0.973	0.967	0.94	0.921
Spegasos	0.96	0.958	0.93	0.913
Bayesian Network	0.947	0.9	0.886	0.882
Voted Perceptron	0.751	0.76	0.735	0.711
Stochastic Gradient Descent	0.991	0.958	0.93	0.913
JRIP	0.996	0.99	0.984	0.981
J48	0.997	0.989	0.981	0.977
Random Forest	0.998	0.993	0.968	0.946
END	0.998	0.992	0.97	0.949
NB Tree	0.997	0.991	0.963	0.975
Neural Network (SOM)	0.77	0.678	0.677	0.677
Decision Table	0.993	0.975	0.944	0.92

Ignored attack or alarm are false positives for the normal activity. They are anomaly activity that is classified as normal cases. Figure 2.6. demonstrate ignored attack rate of each evaluated classification algorithms. RF, NBTree, END, JRip, J48, SMO, RBF and Spegasos results relatively low ignored attack rate based on NSL-KDD dataset. However, NN (SOM) yields relatively high ignored attack rate using NSL-KDD dataset. Overall, JRip and J48 results in more balanced output compared to the others having unbalanced results between each corresponding datasets.

Another important parameter is the time complexity of classification algorithm, RMS error rate and a number of incorrectly classified instances. Time complexity is the time taken by each classification algorithm to build a model within a given set of

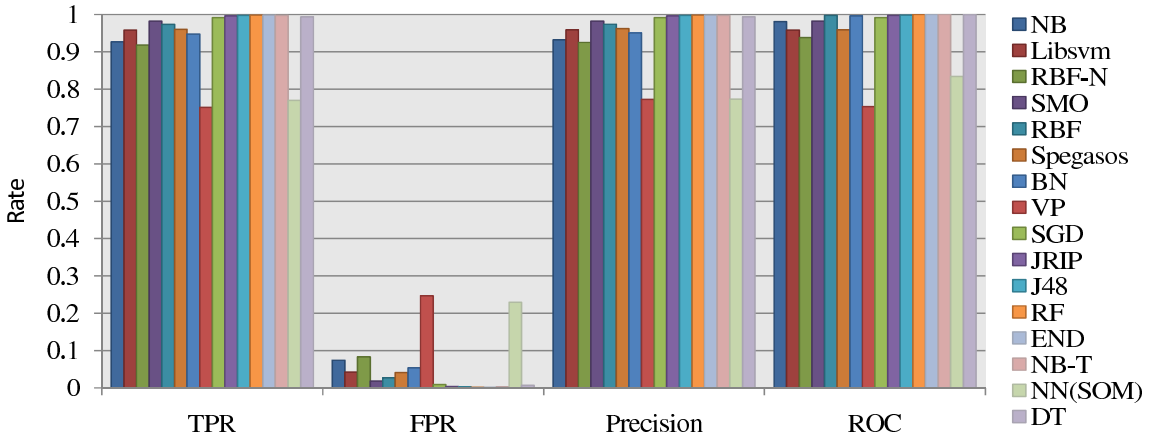

Figure 2.1: Performance of various algorithms on KDD'99.

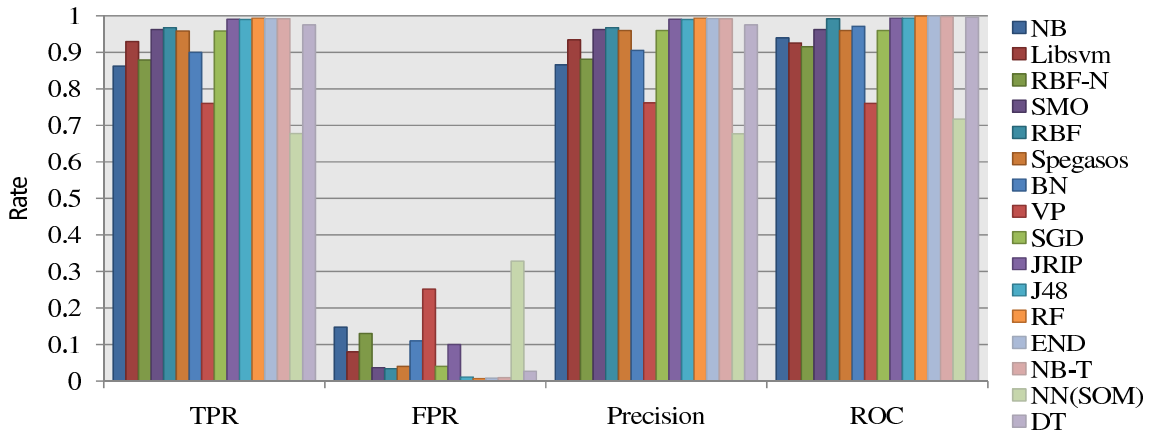

Figure 2.2: Performance of various algorithms on NSL-KDD.

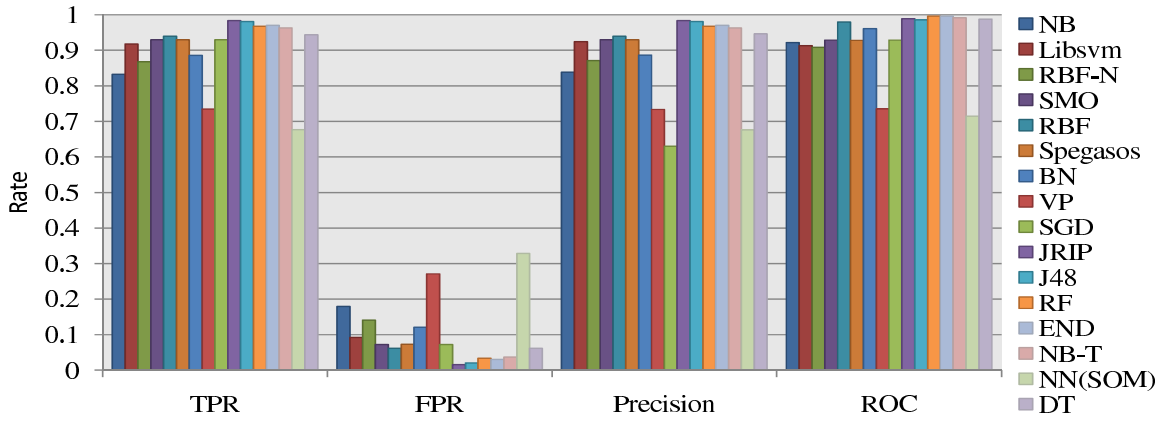

Figure 2.3: Performance of various algorithms on 10% Noisy data.

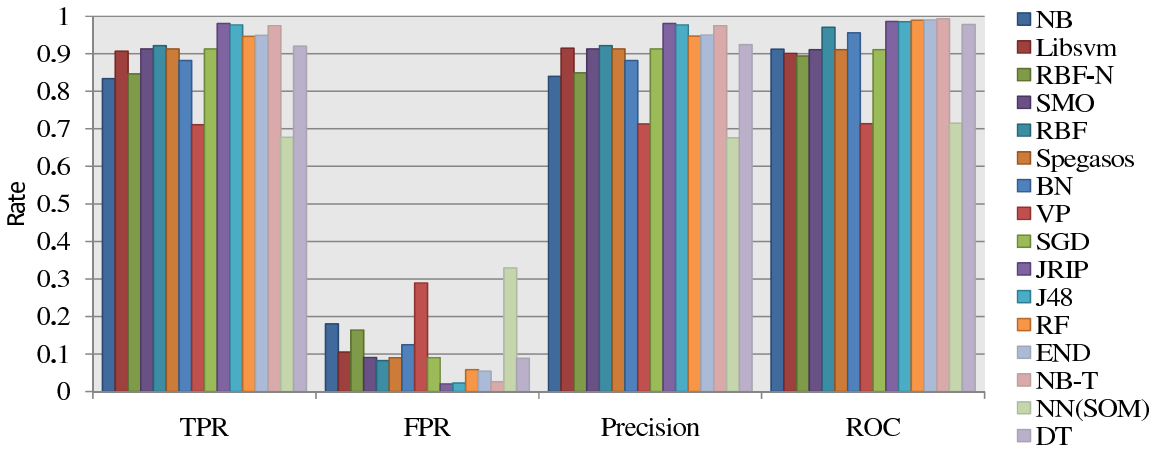

Figure 2.4: Performance of various algorithms on 20 % Noisy data.

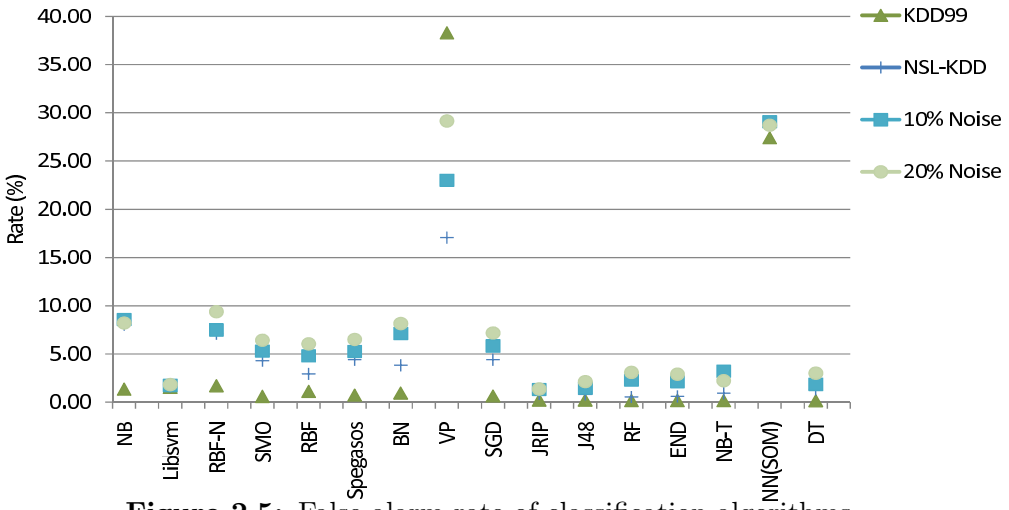

Figure 2.5: False alarm rate of classification algorithms.

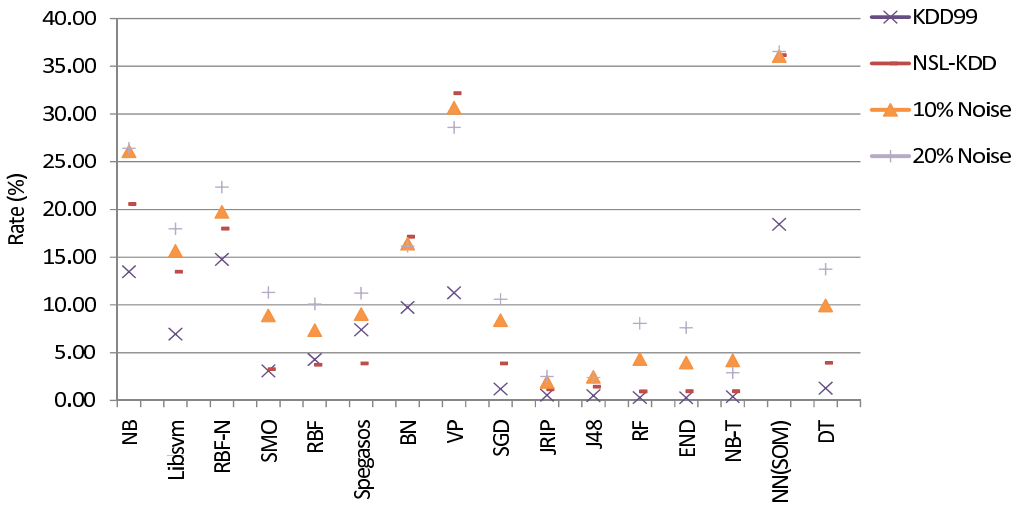

Figure 2.6: Ignored attack rates of classification algorithms.

data and is measured in second (s). # of incorrectly classified instances is the rate of false alarm (Normal instances classified as an anomaly) + False negative (Anomaly instances classified as normal) from the total instances. Figure 2.7. demonstrate the time complexity of each classifier based on various datasets. NB yields the lowest with only 0.19 s derived from both 10% & 20% noisy dataset and proved its robustness to a noisy environment in terms of time complexity. NN (SOM) results in only 0.53 s, 0.59 s, 0.7 s and 1.33 on 20% Noisy, NSL-KDD, 10% noisy and KDD'99. However, SMO yields relatively high time complexity rate, and we observed that the addition of noise badly degrades the SMO classification algorithm. Figure 2.8 shows that the addition of noisy data in selected features seriously degrades various classification algorithms such as NB, SVM, RBF network, SMO, RBF, Spegasos, BN, VP, SGD, and DT. However, NN (SOM), JRip, J48, RF, END and NBTree does not change much of their performance and are found to be more robust to the noisy environment compared to others.

After analyzing each classification algorithm performance based on four datasets, top six classification algorithms are selected based on the entire evaluation matrices and robustness to noisy data. Table 2.2. Shows detail selected classifier that is more robust to the noisy environment. NN (SOM), JRip and J48 are being selected based on its robustness to a noisy environment, while RF, END and NBTree are chosen for the overall performance based on the evaluation matrices. However, NN (SOM) yields the lowest accuracy based on each dataset but scores the highest rank based on robustness to a noisy environment. As this study aimed to select a classifier based on the noise tolerance ability, NN (SOM) is, therefore, placed at the first, where JRip J48, NBTree follows. On the other hand, RF and END are selected based on overall performance though they yield relatively high differences between each dataset.

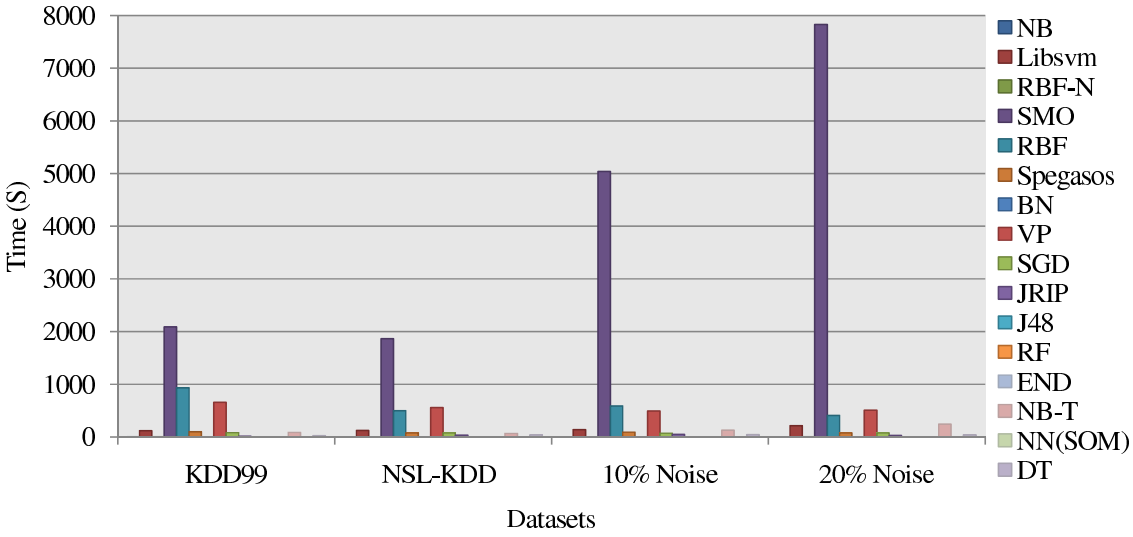

Figure 2.7: Time taken to bulid model on given data (s).

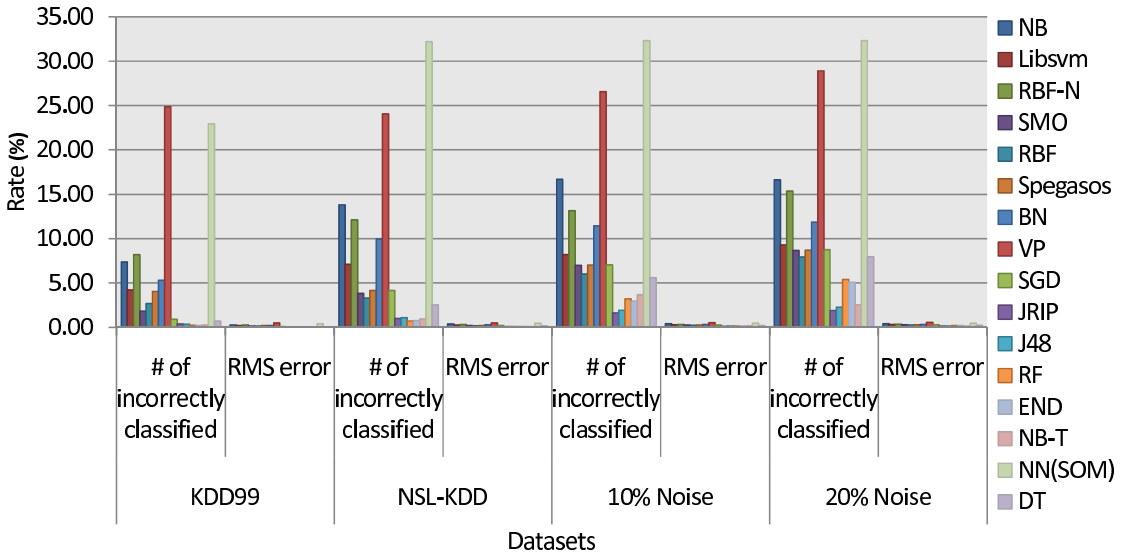

Figure 2.8: Incorrectly classifier and RMS error on given dataset.

Table 2.2: Evaluation performance based on robustness to noise.

Classifier	NSL-KDD	10% Noise	Differences %	20% Noise	Differences %
NN (SOM)	0.678	0.677	0.15	0.677	0.15
JRip	0.99	0.984	0.61	0.981	0.91
J48	0.989	0.981	0.81	0.977	1.21
NBTree	0.991	0.963	2.83	0.975	1.61
END+ND+RF	0.992	0.97	2.22	0.949	4.33
RF	0.993	0.968	2.52	0.946	4.73

Next, the dependency of each feature is analyzed based on performance method by ranking each features using selected classification algorithms and is accomplished by removing unnecessary attributes out of its original dataset. Removal of significant or important features might reduce the performance of the ML algorithm regarding detection accuracy. However, removal of some features that might have a high degree of noise or might not have contributions in any way can extensively advance the performance and search speed of a classification algorithm (Lin *et al.*, 2008).

For PMR, one feature is removed from the dataset at a time, the resultant feature subset data is then used for training and testing each selected classifier (based on k-folds cross-validation method). Then the evaluation performance of the analyzed classifier is compared with the original classifier derived from the original attributes (41 features overall accuracy). Final ranking (significant, insignificant and minor) of the feature is done using the decision rules set (Table 2.3) based on the overall accuracy and time complexity. The Algorithm 1 shown below is used to select performance based ranking method.

Algorithm 1 Performance-based method ranking (PMR)

Choose one classification algorithm at a time, i=0
Read 41 features from supplied original dataset
Train and test the classifier (original set)
while (i++<=41) **do** %comment: Do the following procedure for each feature
    a)Remove one feature out of 41 from the dataset (one at a time)
    b)Use the resultant subset data to train and test the classifier
    c)Compare performance results of the classifier with the original
    d)Based on the decision rules set (Table 2.3) rank the analyzed feature
    e)Continue till all feature are analyzed
end while

Based on the above algorithm, each 41 features are ranked using decision rules set (Table 2.3) for each selected top 6 classification algorithms. Out of 41 original features, a 16 features subset <2, 3, 4, 23, 24, 27, 28, 29, 30, 31, 35, 36, 37, 38, 39, 40> are significant features selected by NN (SOM) Figure 2.9, 37 features subset <1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 27, 28, 29, 30, 31, 32, 33,

Table 2.3: Decision rules set based on performance.

Rules	**Ranked as**
Accuracy reduced and time complexity increased	Significant
Accuracy reduced and time complexity reduced	Significant
Accuracy reduced and time complexity unchanged	Significant
Accuracy increased and time complexity increased	Insignificant
Accuracy increased and time complexity decreased	Insignificant
Accuracy increased and time complexity unchanged	Insignificant
Accuracy unchanged and time complexity increased	Insignificant
Accuracy unchanged and time complexity decreased	Minor
Accuracy unchanged and time complexity unchanged	Minor

34, 35, 36, 37, 38, 39, 40, 41> by JRip Figure 2.10, 23 features subset <2, 3, 4, 6, 10, 12, 14, 16, 17, 22, 24, 25, 26, 28, 32, 34, 35, 36, 37, 38, 39, 40, 41> by J48 Figure 2.11, 35 features subset <1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 22, 23, 24, 25, 26, 27, 28, 29, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41> by RF Figure 2.12, 28 features subset <1, 3, 4, 7, 8, 10, 12, 14, 15, 17, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 32, 35, 36, 37, 39, 40, 41> by END Figure 2.13 and 33 features subset <1, 2, 3, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 28, 30, 32, 34, 35, 36, 37, 38, 39, 40, 41> are selected based on NBTree Figure 2.14. Comparisons of each classification algorithm based on Time complexity are shown in Figure 2.15.

To evaluate the reliability of the performance-based method ranking feature selection, the six feature subsets selected by each classifier are again assessed and tested for each six classification algorithm (Table 2.4). Here, each feature subsets are evaluated using six classifiers until all six feature subsets are tested and analyzed. It is observed that each performance based feature ranking method of each classification algorithm has a unique subset of a feature. Each feature subset selected by a classifier is the best for the based classification algorithm. Therefore, it is seen that, feature subset selected by NN (SOM) classifier resulting 0.75 TP, 0.74 TP, 0.73 TP, 0.73 TP, 0.73 TP, 0.75 TP for NN (SOM), JRip, J48, RF, END and NBTree where 0.75 TP with only 0.26 FP based on NN (SOM) classifier is selected as an optimal performance for

the analyzed feature subset. JRip classifier shows 0.679 TP, 0.99 TP, 0.98 TP, 0.98 TP, 0.98 TP, 0.99 TP for NN(SOM), JRip, J48, RF, END and NBTree respectively, showing that 0.99 TP with only 0.01 FP is scored based on the JRip classifier for the analyzed feature subset. The same observations for each classification algorithms J48, RF, END and NB-Tree are shown in Table 2.4.

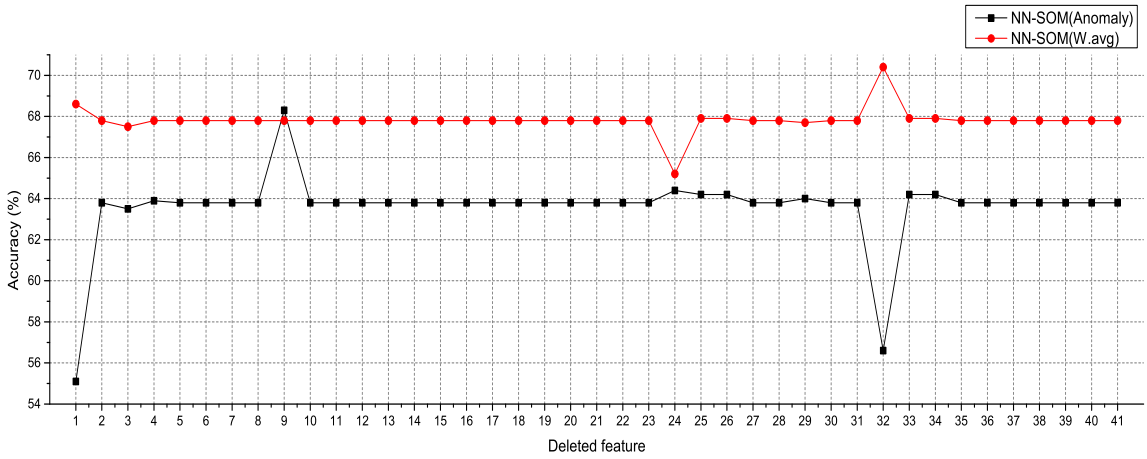

Figure 2.9: Performance of Neural Network (SOM) based on 41 features for Anomaly.

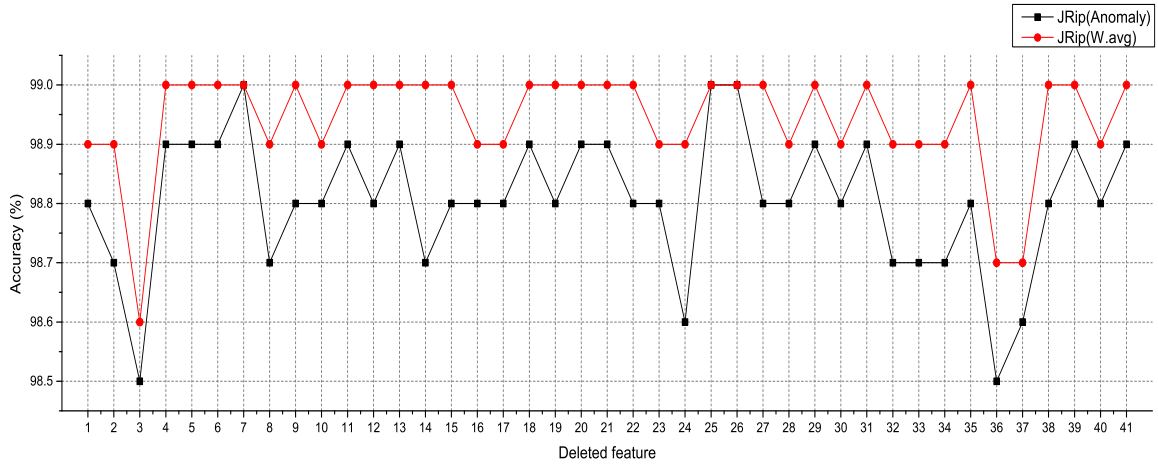

Figure 2.10: Performance of JRip based on 41 features for Anomaly.

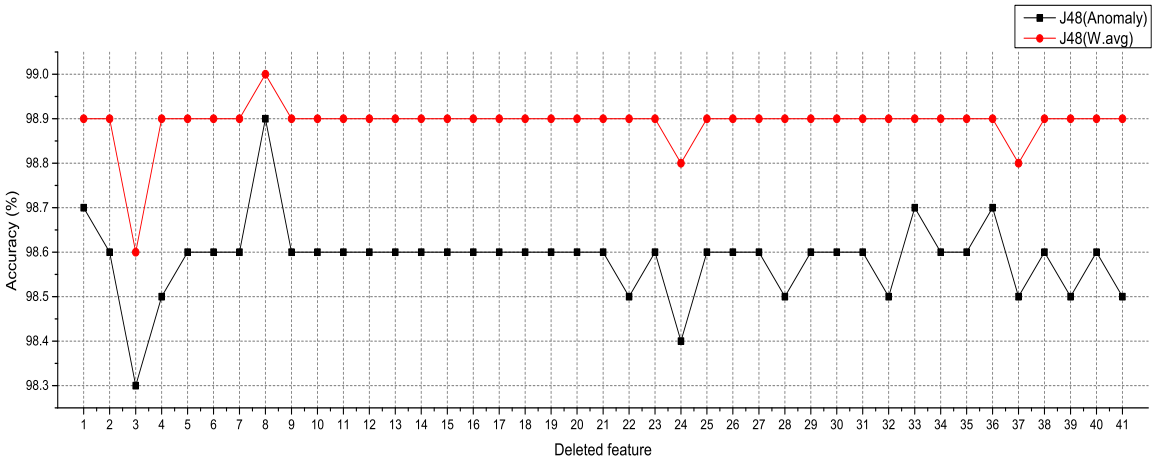

Figure 2.11: Performance of J48 based on 41 features for Anomaly.

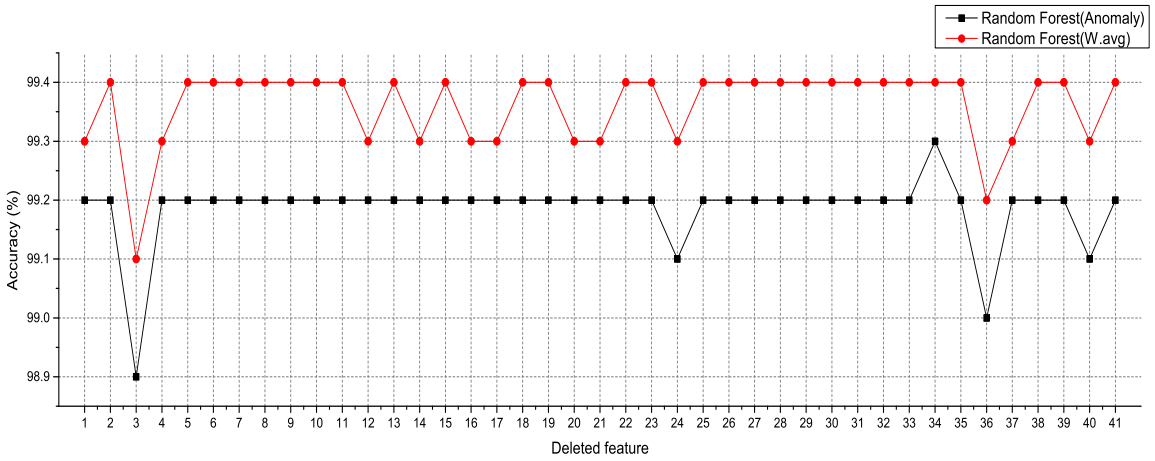

Figure 2.12: Performance of Random Forest based on 41 features for Anomaly.

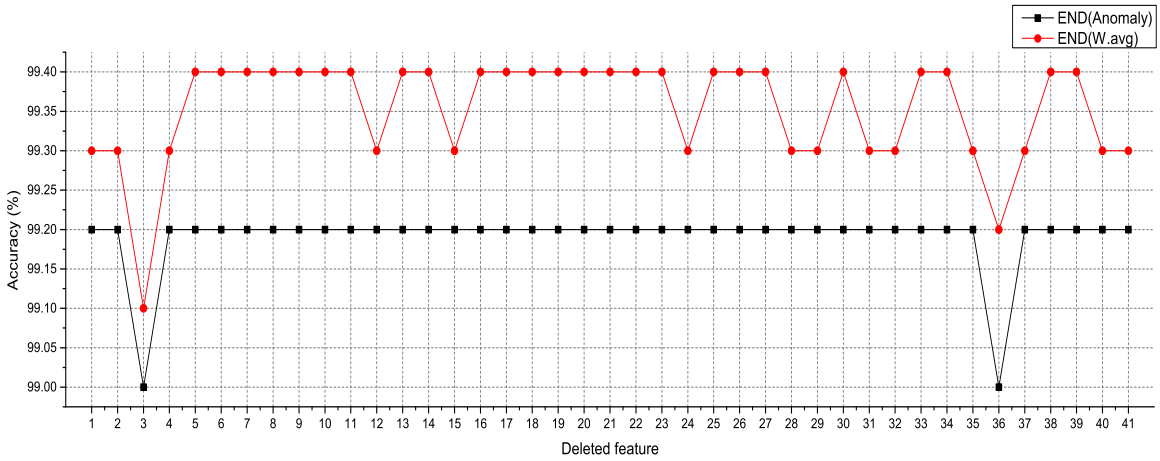

Figure 2.13: Performance of END based on 41 features for Anomaly.

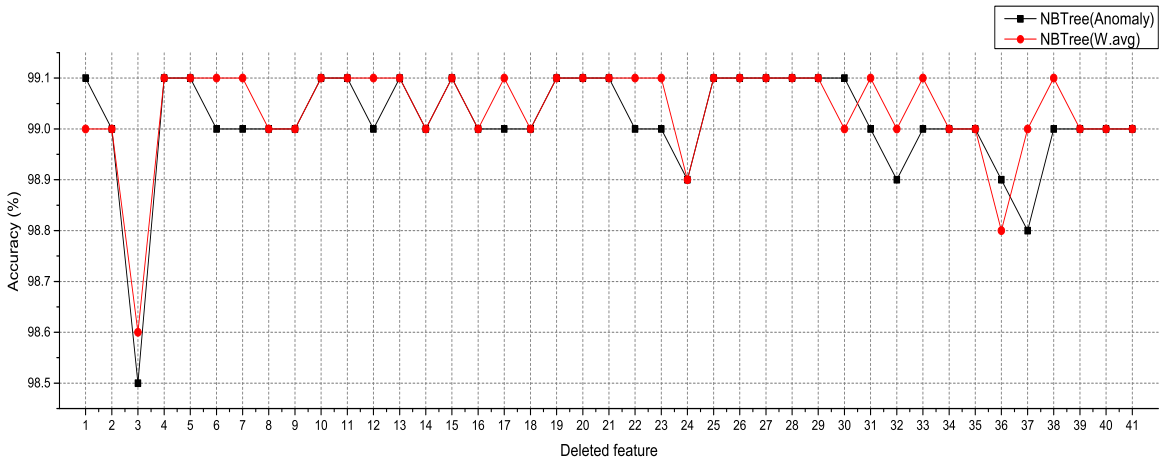

Figure 2.14: Performance of NB Tree based on 41 features for Anomaly.

Table 2.4: Performance of each classification algorithms based on six feature subsets.

Classifier	Six feature subsets from each classifier											
	1. NN(SOM)		2. JRip		3. J48		4. RF		5. END		6. NBTree	
	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
NN(SOM)	0.75	0.26	0.679	0.327	0.68	0.333	0.678	0.328	0.677	0.328	0.677	0.329
JRip	0.74	0.31	0.99	0.01	0.98	0.011	0.98	0.012	0.98	0.013	0.99	0.015
J48	0.73	0.38	0.98	0.011	0.99	0.009	0.98	0.011	0.98	0.011	0.99	0.011
RF	0.73	0.38	0.98	0.07	0.98	0.017	0.99	0.006	0.99	0.008	0.99	0.012
END	0.73	0.37	0.98	0.07	0.99	0.013	0.99	0.007	0.99	0.007	0.99	0.013
NBTree	0.75	0.29	0.99	0.016	0.99	0.012	0.99	0.009	0.99	0.015	0.99	0.009

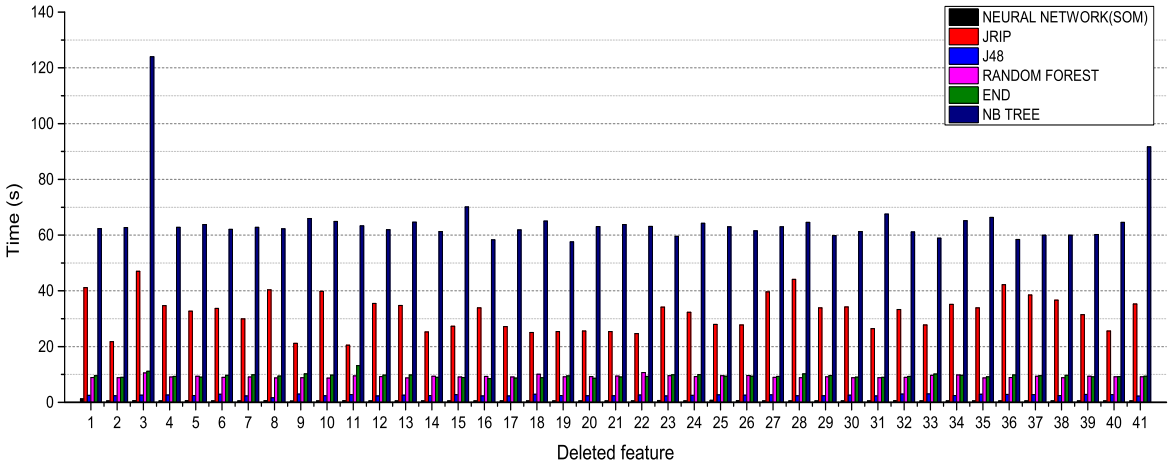

Figure 2.15: Time complexity of each classifier based on 41 features for Anomaly.

2.4 Conclusion

In this section, the performance of various classification algorithms has been compared and evaluated based on the KDD'99 dataset, NSL-KDD dataset and a noise-added dataset with 10% & 20% noise data added to NSL-KDD dataset. Various ML algorithms from various classification algorithm family were tested and compared.

Finally, the comparative studies show that the recent studies by other researchers using various classification algorithms in the absence of noisy environment or noise free dataset could misinform about evaluation performance to a much higher degree. The empirical results demonstrate that the algorithm that performs well on the original KDD'99 dataset does not produce the same result with NSL-KDD, 10% noisy data and 20% noisy data, which proves that the NSL-KDD dataset represents more realistic environment for evaluation of classification algorithms compared to the KDD'99 dataset. Among various tested classification algorithms, JRip and J48 were generally (overall performance, Figures 2.1- 2.4) advanced compared to the other tested algorithms followed by RF, END and NB-Tree. However, Neural Network (SOM) is far more superior to all the others regarding robustness to a noisy environment (Table 2.2). The presence of noise in the datasets does not harm the performance of the algorithm (i.e., 10% & 20% noisy data).

The study of feature selection evaluation based on Performance-based Method of Ranking show that each classification algorithm has a unique combinations of feature subset for the best optimal performance (Table 2.4). Empirical results demonstrated that the feature subsets selected by each classification algorithm are different from each other; dependency of each feature subset depends on the type of classification algorithm selection. It is proved that each classification algorithm has its own unique combination of feature subsets. In other words, the use of significant or dependent features based on PMR for each class in a given classifier, results in the most significant performance.

Chapter 3

A two-stage hybrid classification technique for network intrusion detection system [2]

3.1 Introduction

Conventional network intrusion detection system mostly uses individual classification techniques, such system fails to provide the best possible attack detection rate. In this section, we propose a new two-stage hybrid classification method using Support Vector Machine as anomaly detection in the first stage, and Artificial Neural Network as misuse detection in the second. The key idea is to combine the advantages of each technique to ameliorate classification accuracy along with a low probability of false positive. The first stage (Anomaly) detects abnormal activities that could be an intrusion. The second stage (Misuse) further analyze if there is a known attack and classifies the type of attack into four classes namely, Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R) and Probe. Simulation results demonstrate that the proposed algorithm outperforms conventional model including individual classification of SVM and ANN algorithm. The empirical results demonstrated that the proposed system

[2] ***Accepted for publication in International Journal of Computational Intelligence Systems, Taylor & Francis 2016.***

has a reliable degree of detecting anomaly activity over the network data. Simulation results on both stages are based on NSL-KDD datasets that are an enhanced version of KDD'99 intrusion dataset.

3.2 Dataset description

This study used NSL-KDD dataset (Tavallaee *et al.*, 2009) to demonstrate the superiority of our proposed system. NSL-KDD dataset was an enhanced version of the KDD'99 datasets KDD Cup (1999) created by the Defense Advanced Research Projects Agency (DARPA) at the MIT Lincoln Laboratories located in the United States of America. The KDD'99 contains an enormous number of repeated records of 78% and 75% redundant data on training and test dataset. As mentioned in Chapter 2, the redundant dataset can harm the evaluation result to a much higher degree of detection accuracy. The necessary adjustment made on KDD'99 dataset results in a new NSL-KDD dataset. The Chapter 1 Tables 1.5, 1.6 & 1.7 illustrate the detail modifications made between KDD'99 with attack name and types of attack found in NSL-KDD. NSL-KDD dataset is not perfect and still suffered from some problem criticized by McHugh (2000), but as our main effort is on anomaly based NIDS, it can still be used as a testbed dataset for carrying out various experiments on NIDS. The NSL-KDD dataset classified the different attacks into four broad categories as mentioned in section 1, i.e., Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R) and Probe.

3.3 The proposed hybrid classification method

3.3.1 Support Vector Machine (SVM)

The SVM model was first introduced by Boser *et al.* (1992). The basic idea of SVM is to increase the dimensionality of the samples so that they can be separable. Therefore,

despite the usual trend toward dimensionality reduction, in SVM the dimensionality is increased. The idea is to find a hyperplane to place samples from the same class inside it. SVM with linear and nonlinear kernels have become one of the most promising supervised learning algorithm and able to construct a nonlinear separating that is implicitly defined by a kernel function. In this study, we treated categorizing network traffic into normal and abnormal activity using LIBSVM (Chang and Lin, 2011) C-Support Vector Classification (CSVC) multiclass classification, formulated by Boser *et al.* (1992) and Cortes and Vapnik (1995).

In this context, let given training vectors $x_i \in R^n, i = 1, 2, ..., l$, belong in two classes, and an indicator vector $y \in R^l$ such that $y_i \in \{1, -1\}$. Then to separate the datasets from its origin one needs to solve the following primal optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{3.1}$$

subject to

$$y_i(w^T \Phi(x_i) + b) \geq 1 - \xi_i, \tag{3.2}$$

$$\xi_i \geq 0, i = 1, ..., l,$$

Where $\Phi(x_i)$ maps x_i into a higher-dimensional space and $C > 0$ is the regularization parameter. If w and b solved this problem, then the decision function

$$sgn(w^T \Phi(x) + b) = sgn\left(\sum_{i=1}^{l} y_i \alpha K(x_i, x) + b \right) \tag{3.3}$$

will be positive for most examples x_i contained in the training set.

In this study, we used LIBSVM (version 3.20) available at http://www.csie. ntu.edu .tw/cjlin/libsvm, which is an integrated tool for support vector classification and can handle a binary class or multiclass SVM using Boser *et al.* (1992) and Cortes and Vapnik (1995) algorithm.

3.3.2 Artificial Neural Network

An ANN usually called Neural Network (NN), is a mathematical model or computational model that tries to emulate the structure and functional aspects of biological neural networks (Sammany *et al.*, 2007). ANN is adaptive parallel distributed information processing models that consist of:

- a set of simple processing units (nodes, neurons).
- a set of synapses (connection weights).
- the network architecture (pattern of connectivity).
- a learning process used to train the network.

NN have the potential to address many of the problems encountered by rule-based approaches (Jawhar and Mehrotra, 2010). They are designed to classify statistically significant variations from their established behavior. To apply this approach to IDS, we would first introduce training data representing attacks to the NN to adjust automatically coefficients of this network during the training phase. In other words, it will be required to gather data containing attack behavior and train the network with those collected data. After training the network, a certain number of performance tests with real network traffic data and attacks should be conducted (Novikov *et al.*, 2006). Instead of processing program instruction sequentially, NN based models on simultaneously explore several hypotheses, make the use of numerous computational corresponding elements, this parallel processing may involve time-saving in abnormal traffic analysis (Silva *et al.*, 2004).

3.3.3 Backpropagation

Backpropagation is one of the most commonly used supervised artificial neural network algorithm (Bahrololum *et al.*, 2009). Backpropagation Figure 3.1, aims to train the

network to achieve a balance between the ability to respond correctly to the input patterns that are supplied for training the network and the ability to give reasonable responses to input that is similar to that used in training.

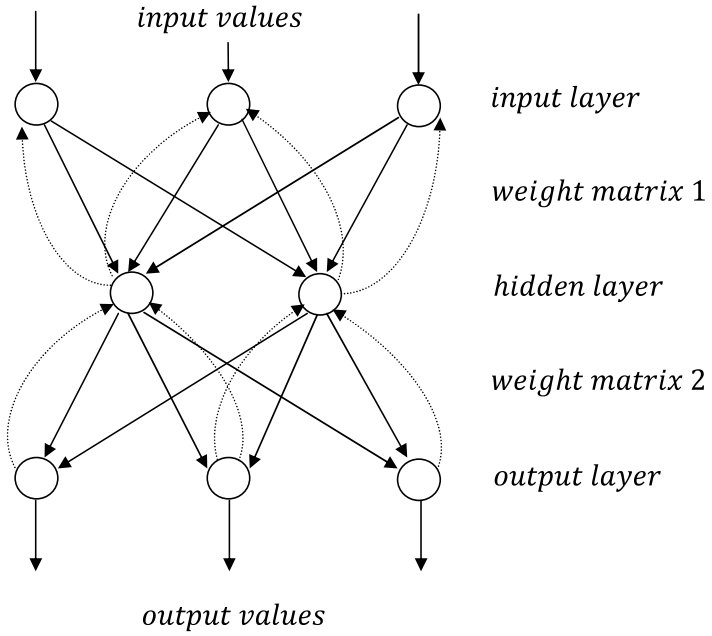

Figure 3.1: Backpropagation Neural Network.

The training of a network by Backpropagation involves three stages: The feed forward of the input training pattern, the calculation, and Backpropagation of the associated error and the tuning of the weights, so that the forward pass produces an output vector for a given input vector based on the current state of the network weights. Since the network weights are initialized to random values, it is unlikely that reasonable output will result before training. The weights are adjusted to reduce the error by propagating the output error backward through the network. This process is where the Backpropagation NN gets its name and is known as the backward pass, and Backpropagation uses the following sequences:

- Calculate error values for each node in the output layer.
- Calculate the error for the middle layer nodes.

- Alter the weight values to progress network performance using the Delta rule.
- Calculate the overall error to test network performance.

The training set is repeatedly presented to the network, and the weight values are altered until the overall error is below a predetermined tolerance. Since the delta rule follows the path of greatest decent along the error surface, local minima can impede training (Shihab, 2006).

3.4 The proposed SVM-ANN (Anomaly-Misuse) hybrid design

In this studies, a network intrusion detection system utilizing both anomaly and misuse technique is proposed. The proposed architecture consists of data preprocess module, a detection and classification module integrating anomaly detection module (Stage-1) and misuse detection and classification module (Stage-2) followed by a final module called alarm module. Stage-1 used SVM to detect traffic anomalies that can be an intrusion and the stage-2 used ANN that further classifies attacks if they exist. The proposed hybrid intrusion detection system (Figure 3.2) illustrate the modules in detail.

3.4.1 Data preprocess

The network traffic was first prepared and preprocessed in the data preprocess module. The two modules in stage-1 (SVM) and stage-2 (ANN) classifiers have their supported data format, all the necessary conversion was accomplished by this module. Subsubsection 3.4.1.1 and 3.4.1.2 gives more detail explanation of the steps accomplished for stage-1 and 2 datasets. For our experimentation, we used full 41 features obtained from NSL-KDD datasets to demonstrate the superiority of our proposed architecture.

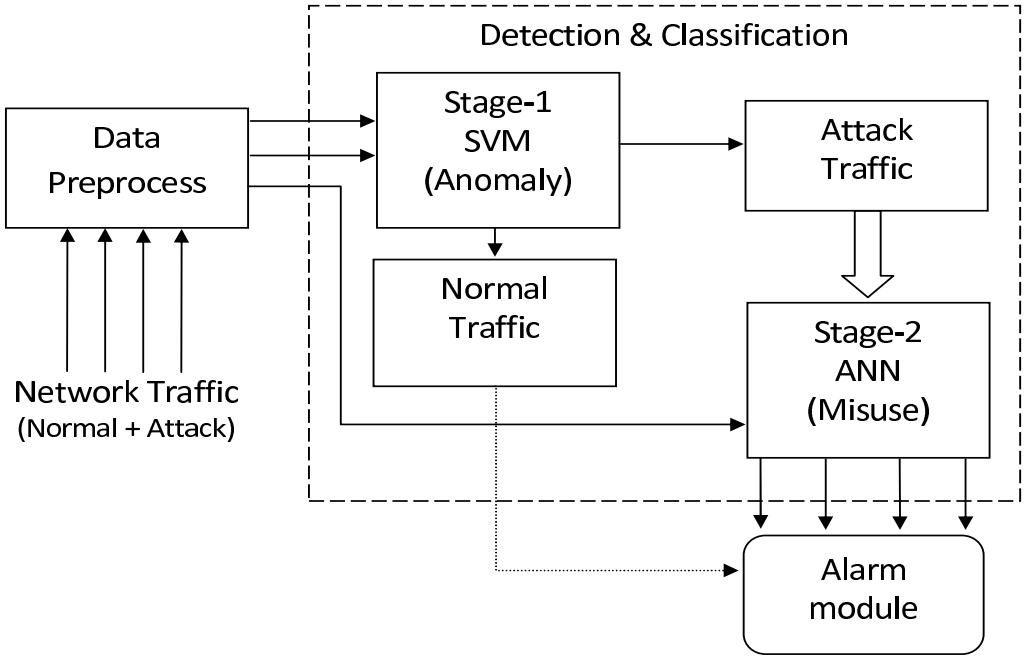

Figure 3.2: Main stages of the proposed approach.

3.4.1.1 Dataset for first stage classifier (DFSC)

The NSL-KDD dataset was analyzed, after preprocessing and reducing redundant data, 161050 instances are selected for experimentation dataset. As shown in Table 3.1, trainset get divided into five sets randomly, containing normal and attack data that appears in NSL-KDD dataset. The attacks contained in NSL-KDD namely, back, land, neptune, pod, smurf, teardrop, satan, ipsweep, nmap, portsweep, guess_password, ftp_write, imap, phf, multihop, warezmaster, warezclient, spy, buffer_overflow, load-module, perl and rootkit. Two test datasets are selected randomly, 500 instances of unknown normal and 500 instances of an unknown attack were employed in the testset, unknown normal or attack means, the normal and attacks traffic data that have neither been used for training nor been seen by the network before. The datasets (Table 3.1) are used for training and testing stage-1 SVM (Anomaly) classifier.

Table 3.1: Distribution of data for first stage clssifier.

Dataset Name	No. of Feature	Normal	Attack
Trainset data 1	41	23665	8545
Trainset data 2	41	21081	11129
Trainset data 3	41	20206	12004
Trainset data 4	41	24628	7582
Trainset data 5	41	22101	10109
Testset data 1	41	28084	4126
Testset data 2	41	26854	5356

3.4.1.2 Datset for second stage classifier (DSSC)

DSSC consist of an attack instances, grouping all the 22 attack into four attack types, i.e., Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R) and Probe. Detail attack types with corresponding attack name are described in Chapter 1 Table 1.7. A trainset consists of 42000 instances employing instances of attack data. Testset consists of 42000 instances of attack data employing 500 unknown attack types; the key idea was to test the reliability of proposed new hybrid algorithm against unknown or anomaly attack using misuse technique. Table 3.2 describes detail organizations of dataset for stage-2 ANN (Misuse) classification level for training and testing the network.

Table 3.2: Distribution of data for second stage clssifier.

Dataset Name	No. of feature	Attack category	Input
Train	41	DoS	36110
		R2L	102
		U2R	9
		Probe	5779
Test	41	DoS	35612
		R2L	101
		U2R	8
		Probe	6279

3.4.2 Detection and classification

In this section, we design two-stage network intrusion detection system using SVM as an anomaly at stage-1 and ANN as misuse at stage-2. The block diagram of the proposed model is shown in Figure 3.2. The NSL-KDD dataset with full 41 original features are used to demonstrate the superiority of the proposed system. The network traffic mixed with normal and attack first passes through the stage-1 (SVM) which classifies the data into normal and attack classes. Stage-2 (ANN) modeled with attack traffic; further classify attack traffic into 4 corresponding attack group. The two-stage architecture reduces the computational complexity while using the full features dataset, resulting the higher degree of accuracy with a low probability of false alarm rate.

3.4.2.1 Stage-1: Anomaly detection module

A multiclass-SVM (stage-1) anomaly classifier using radial basis kernel function was first modeled based on the training set seen on subsubsection 3.4.1.1 containing both normal and attack traffic. The test datasets that include unknown normal and attack are used to test the anomaly module. The attack seen on the original dataset were grouped into two classes, i.e., normal and abnormal or anomalies. Anomalies are defined as the abnormal network behavior in the network. Detection of such activities is the main purpose of this module. The classification results were either normal or abnormal, and all the abnormal traffic were passed to the next stage classifier where misuse technique did further detection and classification.

3.4.2.2 Stage-2: Misuse detection and classification module

In this module, ANN (stage-2) classifier as misuse detection technique using feed-forward network with resilient Backpropagation training function was modeled. The purpose of this module is to further classify the attack data from stage-1 into corresponding 4 classes classification strategies, i.e., Denial of Service (DoS), Remote to

Local (R2L), User to Root (U2R) and Probe. In ML, misuse technique was first trained with the attack traffic to create a model that defines the baseline profile for attack traffic only. On the trained model, a testset was supplied to test whether the traffic is normal or abnormal. An alarm module was triggered if a match is found.

3.4.3 Alarm module

The purpose of this module is to interpret events result on both stage-1 and stage-2 module. It is the final module of the proposed architecture that reports the intrusion detection activity to the administrator or end user.

3.5 Experimental results

In this section, superiority of the proposed method is carefully evaluated throughout experiments using the NSL-KDD datasets via normal classification, attack classification, false positive rate, false negative rate, true positive rate, detection accuracy and error rate. To evaluate the performance of the proposed method LibSVM (Matlab) and Neural Network Tool (Matlab) is used with Windows XP Professional as the test bed operating system on Intel i5 650 @ 3.20GHz processor, 4GB of RAM.

3.5.1 Stage-1 Classification using SVM (Anomaly)

The SVM algorithm with Radial Basis Kernel Function was first trained for each training datasets. The dataset vector consists of 41 features, which is a full feature seen from NSL-KDD dataset as stated in subsection 3.4.1.1 DFSC is used to evaluate stage-1 anomaly classifier. DFSC contain 2 classes, i.e., normal and attack. After applying Radial Basis Kernel Function SVM to 5 different datasets with 2 common test datasets. Various kernel and parameter were evaluated to find the optimal solution, kernel and parameters are experimented aiming to improve the detection performance of the pro-

posed model. The multiclass SVM was tested with parameter γ varied from 0.01 to 0.0001. When γ is 0.01, the multiclass SVM model loses its detection accuracy. As parameter γ decrease, the decision boundary of multiclass SVM becomes more flexible resulting the higher degree of detection accuracy, an increase in parameter γ results to have a high false alarm. Thus, it appears appropriate to set parameter γ to 0.0001 for SVM with RBF kernel. Tables 3.3 & 3.4 describes the detail simulation results obtained after setting the SVM model with appropriate γ on the dataset.

As shown in Table 3.1, the total input data of trainset 1 is 32210 records, 23665 normal and 8545 records as an attack. After applying SVM classification on trainset DFSC with C-SVC with RBF function (γ 0.0001), we get the classification result as trainset 1= 99.95%, trainset 2=99.95%, trainset 3=99.97%, trainset 4=99.90% and trainset 5=99.99%. In Table 3.3 & 3.4, highest accuracy achieved rate is 99.87 %(dataset 1) with 0.92% false positive and 99.97% (dataset 2) with 0.19% false positive rate which is extremely low false alarm rate. Each training set gets evaluated with two testset-1 and testset-2, simulation results shown in Figure 3.3 demonstrate that trainset-1 with testset-1 and trainset-2 with testset-2 scores 99.87 % having 1614 support vectors and 99.97% having 1389 support vectors with an error rate of only 0.0013 and 0.0003, low false positive of 0.92% and 0.19%. Figure 3.4 & 3.5 demonstrate ROC curve for a trainset-1 with the testset-1, trainset-2 with the testset-2 showing comparative results.

Table 3.3: SVM classification results on testset-1.

Name	Train set1	Train set2	Train set3	Train set4	Train set5
Normal classification	28081	28039	27842	28059	27960
Attack classification	4088	4121	4096	4103	4123
False positive rate (%)	0.92	0.12	0.73	0.56	0.07
False negative rate (%)	0.01	0.16	0.86	0.09	0.44
True positive rate (%)	99.99	99.84	99.14	99.91	99.56
Accuracy (%)	99.87	99.84	99.16	99.85	99.61
Error rate	0.0013	0.0016	0.0084	0.0015	0.0039

Evaluation of each simulation results was carefully monitored and measured based on numerical evaluation stated in Wu and Banzhaf (2010) i.e., Accuracy rate, false positive rate (FPR), false negative rate (FNR) and true positive rate (TPR) using the following equations 3.4, 3.5, 3.6, 3.7, 3.8 and 3.9 which is the key point to measure and determine reliability of the system.

Table 3.4: SVM classification results on testset-2.

Name	Train set1	Train set2	Train set3	Train set4	Train set5
Normal classification	26854	26854	26738	26854	26838
Attack classification	5280	5346	5296	5310	5350
False positive Rate (%)	1.42	0.19	1.12	0.86	0.11
False negative Rate (%)	0	0	0.43	0	0.06
True positive Rate (%)	100	100	99.57	100	99.94
Accuracy (%)	99.76	99.97	99.45	99.86	99.93
Error rate	0.0024	0.0003	0.0055	0.0014	0.0006

$$Classification = \frac{Numberclassifiedpatterns}{Totalnumberofpatterns} X100 \tag{3.4}$$

$$FPR = \frac{FP}{FP + TN} \tag{3.5}$$

$$FNR = \frac{FN}{TP + FN} \tag{3.6}$$

$$TPR = \frac{TP}{TP + FN} \tag{3.7}$$

$$Accuracy(AC) = \frac{TP + TN}{TP + FN + FP + TN} \tag{3.8}$$

$$Errorrate = 1 - AC \tag{3.9}$$

False positive are normal data that the system used to detect as attack data. The false positive alarm rates, calculated as the number of normal instances that were classified as attack divided by the total number of normal instances. False negative alarm rate is calculated as the total number of attack data that were classified as normal divided by the total number of attack instances. Recall or Sensitivity is calculated as the proportion of positive cases that were correctly identified divided by total positives.

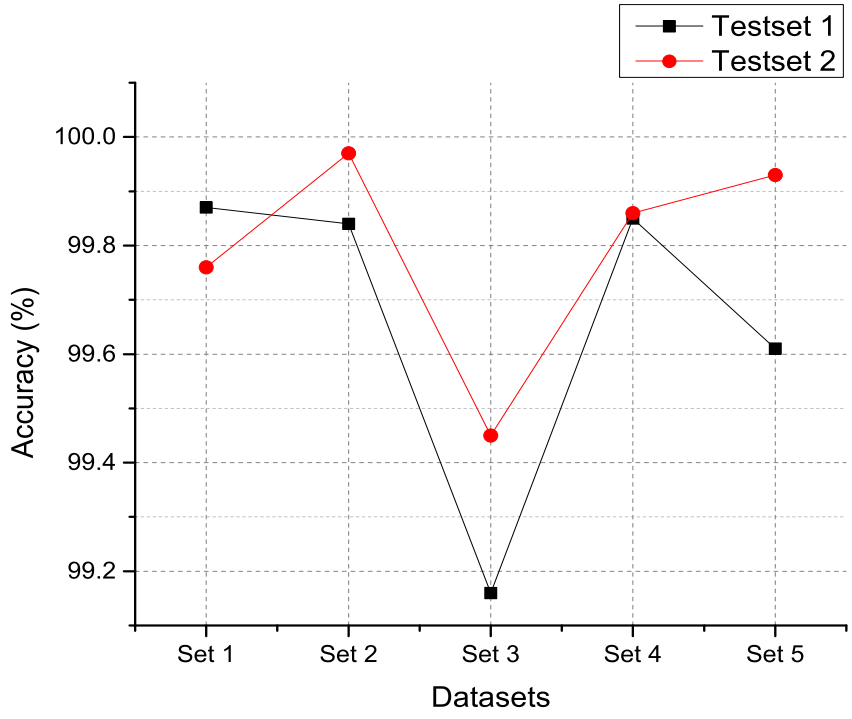

Figure 3.3: SVM classification accuracy on testset.

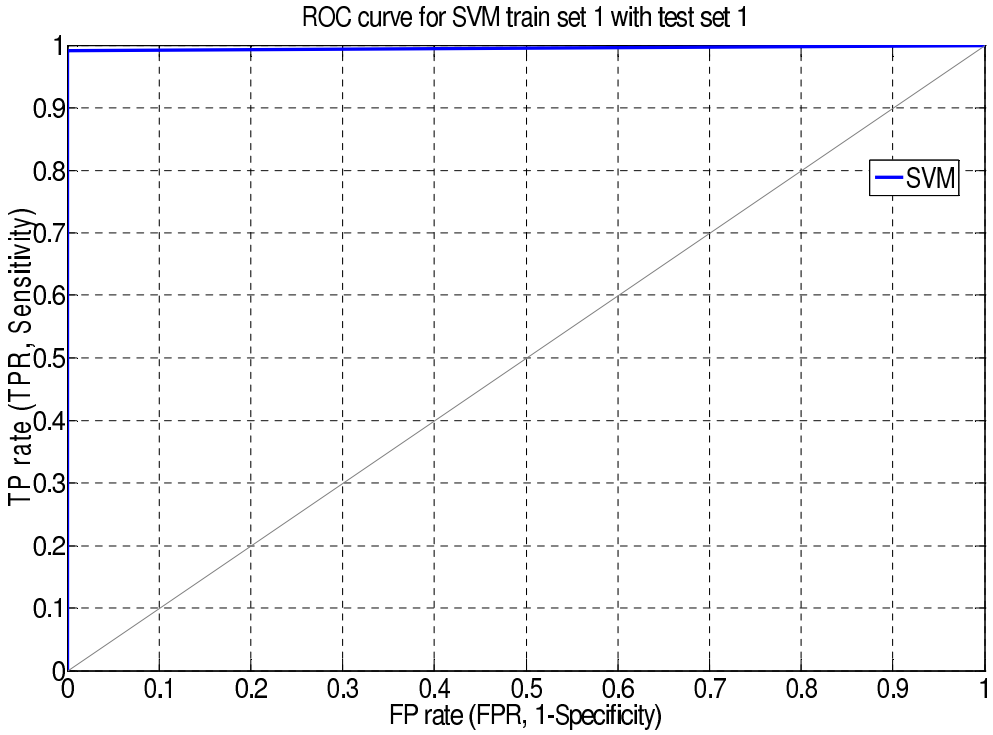

Figure 3.4: ROC curve for SVM (stage-1) trainset 1 with testset 1.

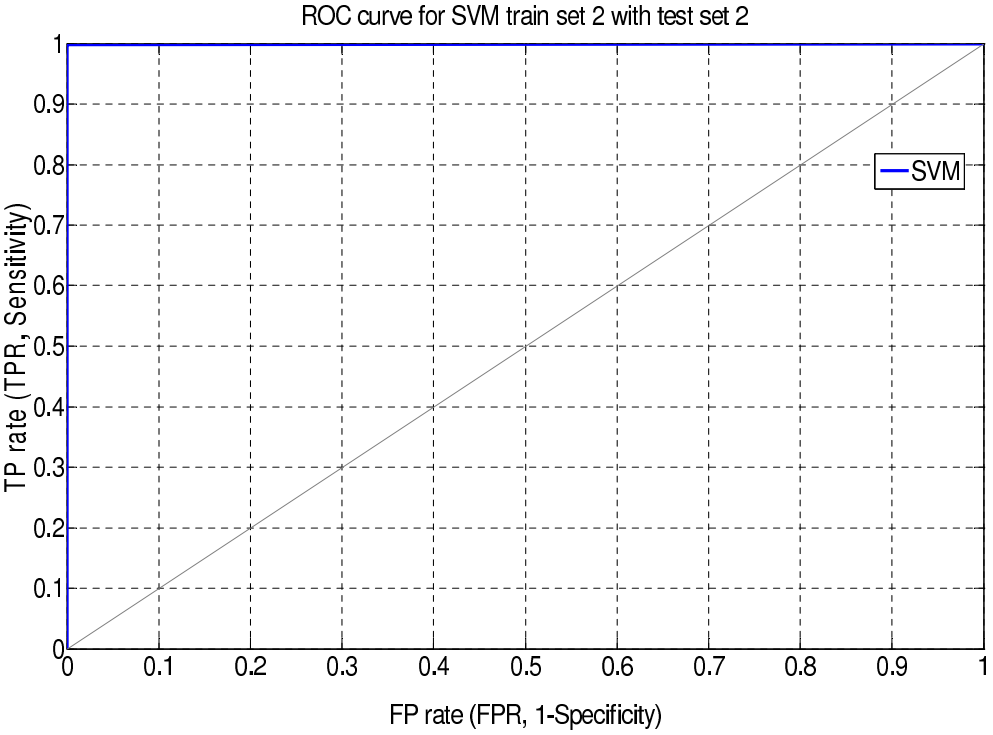

Figure 3.5: ROC curve for SVM (stage-1) trainset 2 with testset 2.

3.5.2 Stage-2 Classification using ANN (Misuse)

In the second stage, ANN algorithm modeled to classify the attack type instances into an attack group of four classes, i.e., DoS, R2L, U2R, and Probe. After testing various network and parameter, a multilayer feed-forward network is found to be the best. The number of hidden layers and number of nodes in the hidden layer was determined based on the process of trial and error. After evaluation of various training functions, a Resilient Back Propagation performed to be the best for our work. While training with Resilient Back Propagation, if the generated output result does not satisfy the target output result, the error from the distortion of the target output was adjusted which leads to re-train or stop training the network depending on the value of error resulted. Once the training is over and satisfies, the weighted value is stored to be used in recall stage. Training and testing dataset are obtained from subsubsection 3.4.1.2 DSSC dataset.

In this module, the neural network is first trained with the training data employing only attack instances creating a network model that is again simulated with a supplied testset data. Various ANN network type was tested with corresponding training functions. Thus, it appears appropriate to set ANN using a feed-forward network with resilient back-propagation training functions.

Table 3.5, Figures 3.6 to 3.10 describe the evaluation result on training and testing phase simulated on ANN model, results in 99.9% detection accuracy at 25 hidden layer with 270 epochs (best validation performance of 0.001455 in Figure 3.7), 35 hidden layer with 180 epochs (best validation performance of 0.0012271 in Figure 3.8) and 40 hidden layer with 90 epochs (best validation performance of 0.0022577 in Figure 3.9).

As shown in Figure 3.10 and simulation results shown by Table 3.5, it appears to set ANN (feed-forward network with resilient back-propagation training functions) with 35 hidden layers with 180 epochs, and results the best detection accuracy of 100%, 87.1%, 87.5% and 100% for DoS, R2L, U2R and Probe attack types with relatively low

false positive rate of only 0.1%.

Table 3.5: Simulation results of ANN multi-layer feed-forward network with resilient back propagation.

Test data	Attack category	25 Hidden layer 270 epochs	35 Hidden layer 180 epochs	40 Hidden layer 90 epochs
1	DoS	100%	100%	100%
	R2L	84.20%	87.10%	76.20%
	U2R	75%	87.50%	87.50%
	Probe	99.70%	100%	99.80%
	Overall Ac	99.90%	99.90%	99.90%

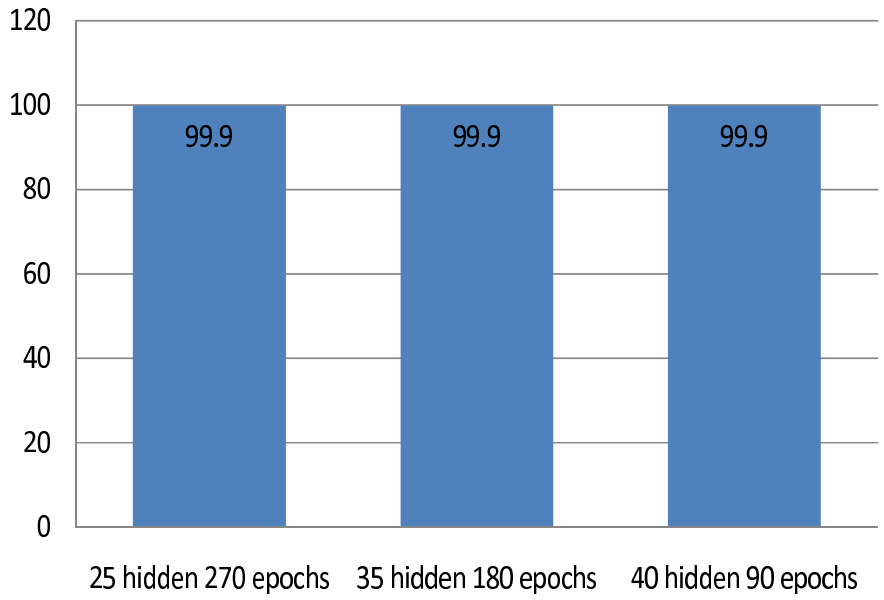

Figure 3.6: ANN (stage-2) classification accuracy on testset (%).

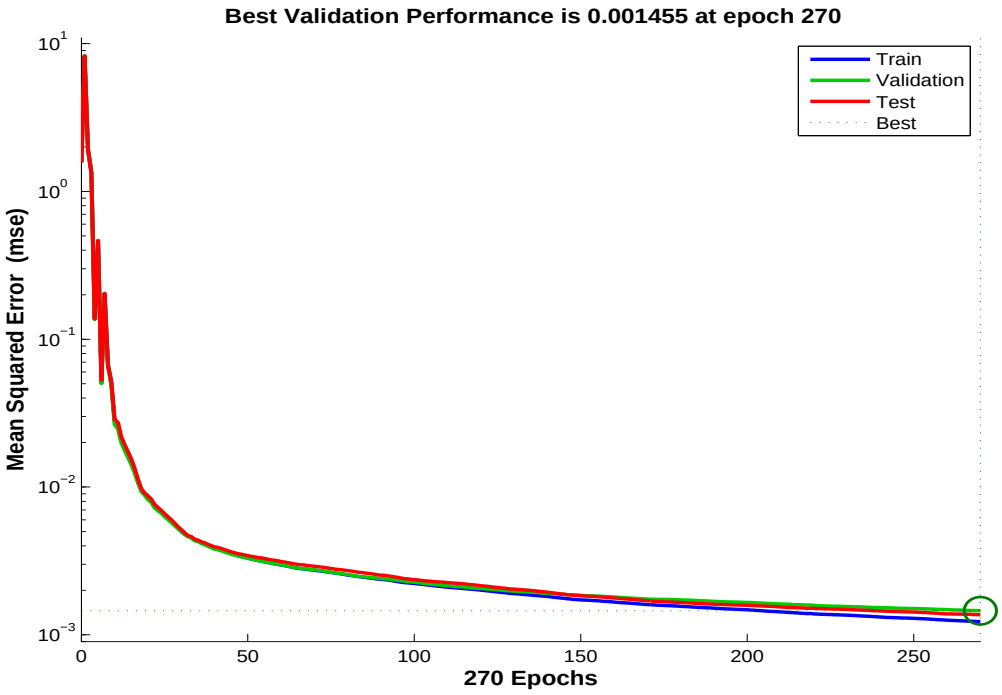

Figure 3.7: Performance of stage-2 classifier with 25 hidden layers at 270 epochs.

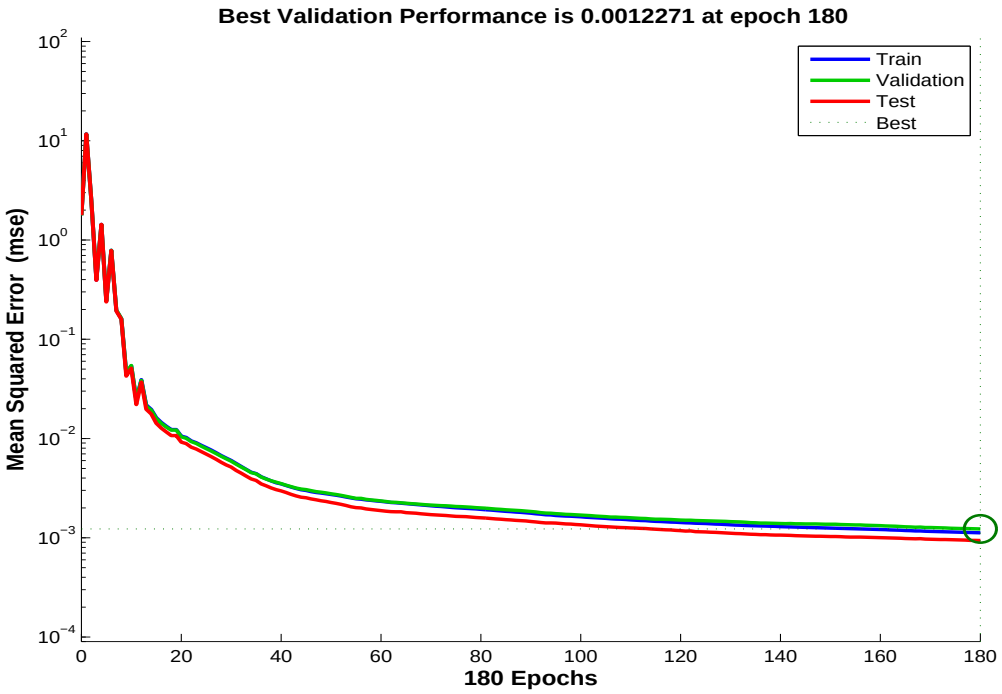

Figure 3.8: Performance of stage-2 classifier with 35 hidden layers at 180 epochs.

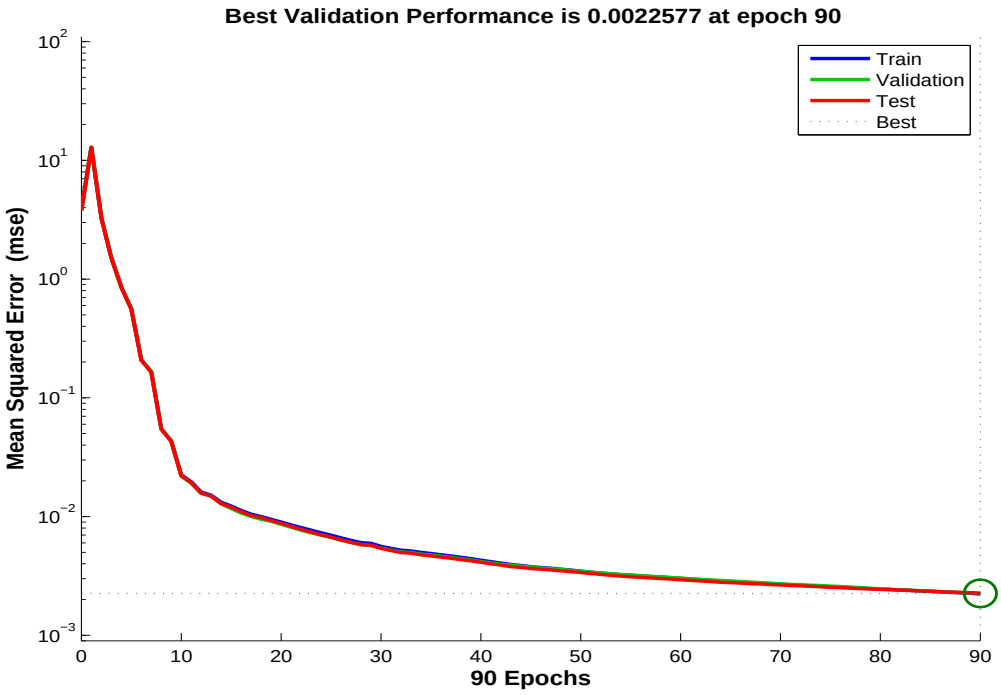

Figure 3.9: Performance of stage-2 classifier with 40 hidden layers at 90 epochs.

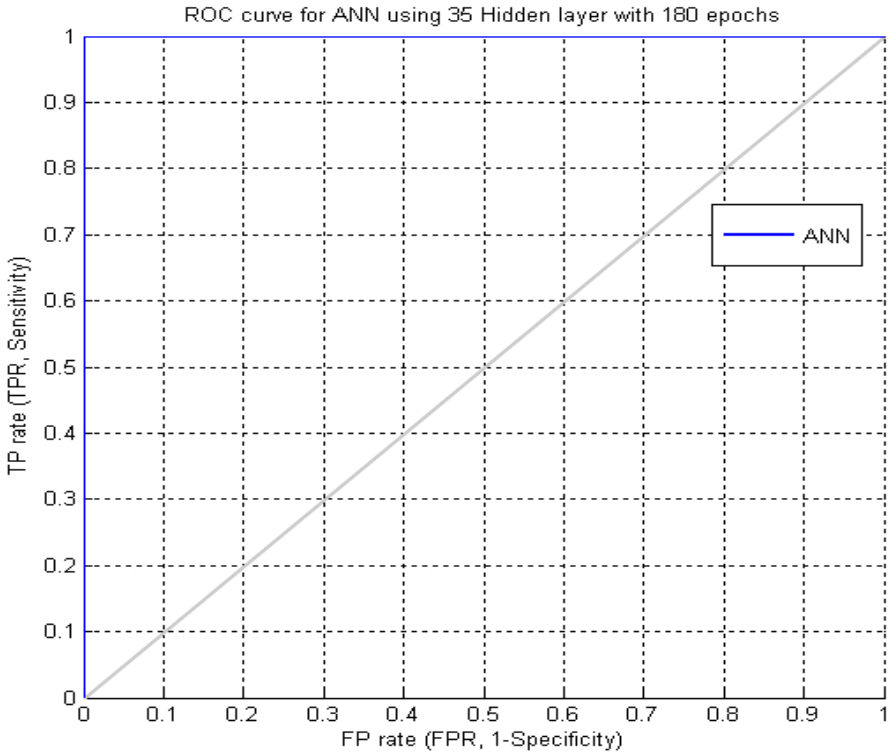

Figure 3.10: ROC curve for ANN stage-2 (35 hidden layers with 180 epochs).

3.5.3 Hybrid classification (two-stage) anomaly-misuse compared to single stage classification

This section combines the whole datasets DFSC from subsubsection 3.4.1.1 and DSSC from subsubsection 3.4.1.2. Both SVM and ANN were tested separately with the corresponding training and test datasets using 5 - classes (Normal, DoS, R2L, U2R, and Probe). After training and testing the individual anomaly module, 98.72% of detection rate with 0.7% probability of false alarm was achieved by SVM using the same function and parameter as subsection 3.5.1. The individual misuse detection module with ANN scores weighted average of only 86% detection rate along with the high false positive rate of 5.6%. As shown in Table 3.6, the weighted average of hybrid (two-stage) classification outperforms single and conventional hybrid classification technique, scoring high probability of detection accuracy 99.95% with a low false positive rate of only 0.2%. Table 3.7 also demonstrate the comparison between conventional hybrid serial, parallel and two-stage network intrusion detection system results with the other researcher's result available.

Table 3.6: Comparisions of individual model with the proposed Two-stage (Hybrid SVM-ANN) classification accuracy.

Classification Algorithm	Individual SVM	Individual ANN	Proposed hybrid model(Two-stage) SVM - ANN
Normal	99.90%	81.30%	99.91%
DoS	66.60%	93.60%	100%
R2L	79.20%	0%	77.40%
U2R	0.10%	0%	88.60%
Probe	77.10%	99.70%	99.90%
Avg. FPR	0.70%	5.60%	0.20%
Avg. AC	98.72%	86%	99.95%
Dataset	NSL-KDD	NSL-KDD	NSL-KDD
No. of features	41	41	41

Table 3.7: Comparisions of conventional model hybrid IDS classification model.

Classification Algorithm	Conventional hybrid model(Two-stage) (Khan & Khan, 2008)	Conventional hybrid model(Two-stage) (Yousef et al, 2014)	Conventional hybrid model (Serial) (Kim *et al.*, 2014)	Conventional hybrid model (Two-stage Parallel) (Depren *et al.*, 2005)	Conventional hybrid model (Ghanem *et al.*, 2015)
Weighted Avg. AC (%)	84.8	94.2	99.1	99.8	96.1
WeightedAvg. FPR (%)	0.1	5.8	1.2	1.25	3.3
Dataset	KDD 99	KDD 99	KDD 99	KDD 99	NSL-KDD
No. of features	10	7+12	6	6	41

3.6 Conclusion

In this chapter, a new hybrid network intrusion detection system using two-stage (Anomaly-Misuse) hybrid classification technique have been proposed and evaluated. Stage-1 used one SVM to detect traffic anomalies that can be attack and the stage-2 used one ANN that classifies attacks if they exist. A full 41 dimension features of NSL-KDD data set was used throughout the experiment.

Various functions and parameter are tested in both algorithms (stage-1 & stage-2). The evaluation results show that high detection rate 99.97% with a low false positive rate of only 0.19% achieved by stage-1 anomaly detection (Figure 3.5 & Table 3.4). Table 3.5 demonstrates that 99.9% detection accuracy with only 0.1% false positive rate achieved at stage-2 misuse detection and classification (Figure 3.10). This was achieved through the design of a classification model using SVM with Radial Basis Kernel Function at the first-stage (Anomaly) and Neural Network using Multi-layered Feedforward Neural Network with Resilient Backpropagation at the second-stage (Misuse).

The key idea of the proposed two-stage classification is to combine the advantage of

both Anomaly and Misuse classification technique, the proposed two-stage classification technique helps in reducing the computational complexity in both stages resulting an improvement on detection rate for anomaly intrusion detection.

Finally, we have found that the proposed two-stage system (Table 3.6) outperformed single-stage classification technique using the whole datasets from section 3.4.1.1 & 3.4.1.2 with 5 classes, resulting 99.95% detection accuracy with the low false positive rate of only 0.2%. Individual classification using SVM results in 98.72% accuracy along with 0.7% false positive while single-stage ANN results in 86% detection rate with the relatively high false positive rate of 5.6%.

We have concluded that this study gives evidence for improvements on anomaly intrusion detection. The combinations of SVM-ANN (Anomaly-Misuse) have proven their effectiveness to detect new attacks over single and conventional hybrid classification technique. Figure 3.3 to 3.10 demonstrate that our work contributes to design a new classification model to achieve higher detection accuracy along with the lower probability of false alarm rate (false positive). As shown in Table 3.6 & 3.7, the proposed new hybrid model is found to be comparative for classification that outperform the recent conventional model, i.e., Depren *et al.* (2005) results 99.8% AC along with 1.25% FPR, Kim *et al.* (2014) results in 99.1% average AC with 1.2% FPR, Ghanem *et al.* (2015) result in 96.1% AC along with high degree of 3.3% FPR, Yousef *et al.* (2014) results in 94.2% AC with high probability of 5.8% FPR, Khan and Khan (2008) that results in 84.8% AC along with 0.1% FPR. The compared conventional model are various proposed hybrid model for IDS that uses the same KDD99/NSL-KDD datasets for evaluation and help us to conclude that our proposed hybrid approach delivers better detection accuracy among the existing models.

Chapter 4

A hybrid classification for network intrusion detection system based on ensemble method[3]

4.1 Introduction

A new era of classification technique, called hybrid intelligent system, has been proposed to improve the degree of accuracy of anomaly IDS compared to the individual classification technique (Pan *et al.*, 2003). Hybrid intelligent system is a combination of multiple classification approaches to give better result of classification algorithms, and it result in more detection accuracy (Mukkamala *et al.*, 2003; Bouzida and Electric, 2006; Peddabachigaria *et al.*, 2007; Shon and Moon, 2007) compared to individual classification. The most common approach used in hybrid intrusion detection system is Feature Selection (FS), FS is of two techniques: filter and wrapper method. Recent research has proposed various feature selection system for better IDS evaluation result

[3]The content of this chapter is published in two research article:

1. A hybrid approach for determining the efficient network intrusion detection system, ***IUP Journal of Computer Sciences***, **8(3)**, 34-46 (2014)
2. An intelligent hybrid decision approach with feature selection for anomaly network intrusion detection system, ***In. Proc. 5th Int. Conf. on Internet Technologies and Society, Taiwan,*** 3-10 (2014)

(Lee *et al.* 2003; Pajares *et al.*, 2004; Pai *et al.*, 2005; and Lin *et al.*, 2008a, 2008b, 2008c and 2009).

Therefore, this Chapter presents a new hybrid approach for NIDS using ensemble method, combining Adaboost (AB) algorithm with C4.5 decision tree (DT) classification. DT can find the optimal feature selection to ameliorate the accuracy of anomaly intrusion detection, by automatically adjusting the optimal parameter settings for the proposed model. After applying feature selection using unsupervised wrapper method over the whole dataset, the output data is again applied to a classifier for further classification. To carry out our experiment we used k-fold cross validation over two/five-class (i.e., normal and anomaly "Probe, U2R, R2L, and DoS") classification strategy where k is set to 10. Our proposed new hybrid NIDS technique was evaluated with the NSL-KDD datasets, which is a customized and enhanced edition of KDD'99 datasets developed by DARPA. The proposed new model outperforms other existing conventional approaches to both individual and hybrid classification; simulation results describes that the proposed new model is more reliable in detecting anomaly intrusion detection system based on 2 and 5-classes classification technique.

There are many critiques in describing the attack taxonomies among the NIDS community (Panda *et al.*, 2012), which cause difficulty in completely describing the normal behavior of the network. So, this study will concentrate on anomaly based IDS using 2 and 5-class classification strategies. i.e., normal and attack (Probe, U2R, R2L and DoS) rather than identifying the attacks detail information.

4.2 Theory and Algorithms

To carry out our study and experimentation, we used data Wrapper (Snchez-Maroo, 2009) technique for feature selection based on both supervised/unsupervised technique to select the most considerable sub-feature among the original NSL-KDD dataset, which was again followed by testing each combination of classification algorithm model

to choose the best hybrid combination approach for NIDS. In this section, we experiment various combination of a classifier such as AdaBoost, Decision Tree (DT), Random Forest (RF), Chi-Squared, GainRatio, Infogain, Ensembles of Balanced Nested Dichotomies for Multi-class Problems (END), Radial Basis Function (RBF) Network, Naive Bayes, Stochastic variant of Primal estimated sub-gradient solver in SVM (Spegasos) with each performance metrics describing the most significant models.

4.2.1 AdaBoost

AdaBoost was first proposed by Freund and Schapire (1996) and won the Godel Prize 2003. It can be used to improve the performance of other learning algorithms by extensively reducing the error of any other weak learning algorithm. The Adaboost repetitively runs a particular weak learning algorithm on different distributions over the given training dataset; a single composite classifier is formed from the combination of classifier produced by the weak learner algorithm. In each iteration, the correctly identified data gets decreased while the weight of incorrectly identified data is increased. One drawback of this algorithm is that its sensitivity to noisy data and outliers lead Adaboost to overfitting problems as most learning algorithm does.

AdaBoost has two versions that are denoted as AdaBoost.M1 and AdaBoost.M2 algorithm. Adaboost.M1 is the first release which simpler to AdaBoost.M2, and its main disadvantages are that it is unable to handle weak hypothesis with an error $> 1/2$. The projected error of a hypothesis label by randomly guessing is $1 - 1/k$, where k is the amount of available label.

The AdaBoost.M1 algorithm takes input from a training set of m examples, $\langle(x_1, y_1)$, ..., $(x_m, y_m)\rangle$ where x_i is instance drawn from some space X that is represented as vector of attributes values and, $y_i \in Y$ is a class label that is associated with x_i. The boosting algorithm calls WeakLearn repeatedly in a series of rounds denoted as t. It provides WL with a distribution D_t over a training set S. WL calculate hypothesis

$h_t : X \rightarrow Y$ which should misclassify a non trival fraction of training examples, relative to D_t which is the goal of the learner to find hypothesis h_t to reduce the error rate at the training phase $\in_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$. This process continues up to T rounds and at the end the algorithm used to combine the weak hypothesis $h_t, ., h_T$ into a final single hypothesis h_{fin}. Algorithm 2 demonstrate details of the Adaboost.M1 algorithm used in this study and more detail can be obtained from Freund and Schapire (1996).

Algorithm 2 AdaBoost.M1 Algorithm

Input: sequence of m examples $\langle (x_1, y_1), ..., (x_m, y_m) \rangle$with label $y_i \in Y = \{1, ..., k\}$ weak learning algorithm **Weak Learn** integer T specifying the number of iterations.

Initialize $D_1(i) = 1/m$ for all i.

Do While $t = 1, 2, ..., T$

1. Execute **WeakLearn(WL)**, providing with the distribution D_t
2. Get back a hypothesis $h_t : X \rightarrow Y$.
3. Calculate error of $h_t : \in_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$.

   If $\in_t > 1/2$, then $T = t - 1$ then abort loop.
4. Set $\beta_t = \in_t / (1 - \in_t)$.
5. Update the distribution

$$D_t : D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t \ if \ h_t(x_i) = y_i \\ 1 \ otherwise \end{cases}$$

Where Z_t is normalization constant (chosen so that D_{t+1} will be distribution)

Output final hypothesis: $h_{fin}(x) = \arg\max_{y \in Y} \sum_{t:h_t(x)=y} \log \frac{1}{\beta_t}$

4.2.2 C4.5 Decision Tree

A Decision Tree is a classification algorithm where the instance space is expressed as a recursive partition. It composed of nodes forming a root node, which means it is a directed tree with no inner edges. All the other nodes in a tree precisely have 1

inside edge. Each internal node gets divided into 2 or more sub-spaces depending on the certain discrete function of the input attributes values (Chen *et al.*, 2003). In the simplest and most general case, each test considers a single attribute; such that the instance space gets partitioned in correspond with each attribute's value (Maimon and Rokach, 2010). C4.5 (Quinlan, 1993) is a successor of ID3 (Iterative Dichotomiser 3) (Quinlan, 1979). C4.5 deeply outperforms ID3. It can receive both nominal as well as numeric features for input datasets. In general, C4.5 decision tree algorithms uses divide and conquer technique to perform classification that can be expressed as decision trees or in the form sets of the rule (Wu *et al.*, 2008).

4.2.3 Dataset descriptions and Performance evaluation

For carrying out our experiment, we used the dataset proposed by Tavallaee *et al.* (2009) NSL-KDD (Available at http://nsl.cs.unb.ca/NSL-KDD/) dataset that is an improved edition of KDD'99 data, the benchmarked dataset created by the DARPA at the MIT Lincoln Laboratories USA. This study used two distinct classification strategies, i.e., 2-class and 5-class, the first evaluation dataset deals with 2-class (normal and attack only) while the second evaluation dataset contains 5-class (Normal, Probe, U2R, R2L, and DoS) classification strategies. Detail distribution of particular attack with corresponding attack group is shown in Chapter 1 Table 1.7.

Performance of the classifier is monitored and measured as the following pattern:

- True Positive Rate (TPR): It is a proportion of class classified as class-A by actual total in class-A. Also successful detection of class-A among all data which exactly belongs to class-A. It is also sometimes called detection rate and Recall.
- False Positive Rate (FPR): It is a proportion of incorrectly classified as class-A by an actual total of all classes except class-A. Also known as False Alarm, it corresponds to an anomalous event that is inoffensive from a security point of view.

- Precision: It is the proportion of that class that has class-A by total classified as class-A.
- Time taken to build model: It is the time taken by each classifier to build the model, it used to measure in second.
- F-measure: It computes the average of information retrieved by precision and recall i.e. 2*Precision*Recall/ (Precision + Recall).

4.3 Proposed system

This study used NSL-KDD dataset, to demonstrate the reliability of the proposed hybrid algorithm over the conventional individual and hybrid classification technique. There are 41 features in the original NSL-KDD dataset, the relationship between each 41 features might relate to the classification output and complexity of computation. Removal of some key feature might reduce the degree of classification accuracy while removal of some feature having no result at all, or containing a high degree of noise ratio might improve the level of classification accuracy and search speed (Lin *et al.*, 2008).

Feature selection is done by filtering the dataset after applying C4.5 decision tree using wrapper method on original dataset, selecting the most relevant subsets features from the supplied original dataset, and proposed 13 features (for 2-class classification) as shown in Table 4.1 and 11 features (based on 5-class classification) shown in Table 4.2 for the new hybrid algorithm. The output data get applied to the AdaBoost in combinations with C4.5 decision tree model based on the k-fold cross validation (FCV) method. To evaluate the suitability of our proposed hybrid algorithm over two-class (i.e., normal and anomaly) classification strategy. The value of k is set to 10. 10-FCV is a technique where the original dataset get divided into 10 subsets, where 9 subsets of those are used as training datasets, 1 subset out of 10 used as test dataset and each

fold has the same ratio of the original class. Both the selected subset of features is applied to a hybrid classification model combining Adaboost with a C4.5 decision tree. The C4.5 algorithm gets boosted up to 10 iterations, resulting in optimal classification accuracy with an improved time complexity.

Table 4.1: Proposed 13 features selected from NSL-KDD dataset by C4.5 decision tree using wrapper method based on 2-class classification.

Name of Feature	Data type	Description
duration	Continuous	Length of the connection
service	Nominal	Destination service
src_bytes	Continuous	Bytes sent from source to destination
logged_in	Nominal	1 if successfully logged in; 0 otherwise
su_attempted	Nominal	1 if su root command attempted; 0 otherwise
num_file_creations	Continuous	Number of the file creation operations
count	Continuous	Number of connection to the same host as the current connection in the past two seconds
srv_count	Continuous	Number of connections to the same service as the current connection in the past two seconds (same-service connections)
dst_host_srv_count	Continuous	% of connections having the same destination host and using the same service
dst_host_same_srv_rate	Continuous	% of connections having the same destination host and using the same service
dst_host_diff_srv_rate	Continuous	% of different services on the current host
dst_host_serror_rate	Continuous	% of connections to the current host that have an S0 error
dst_host_rerror_rate	Continuous	% of connections to the current host that have RST error

To select the most optimal detection accuracy of each classifier with low false positive rate, the performance of each model over the dataset was monitored carefully with the basic pattern mentioned in section 4.2.3. Figure 4.1 illustrate a block diagram of the proposed hybrid network intrusion detection system.

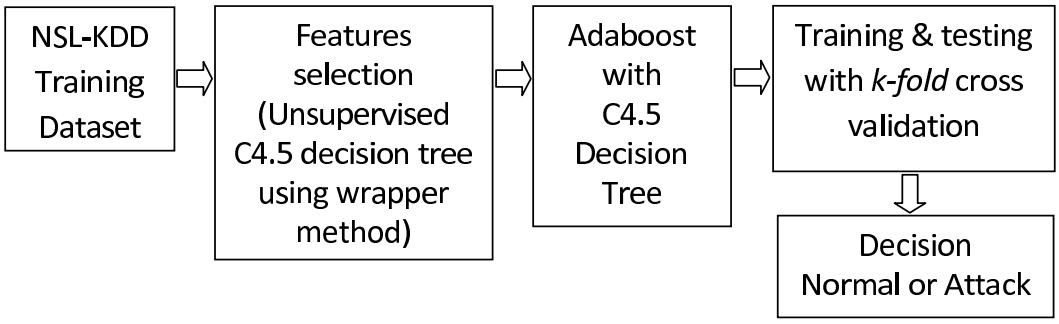

Figure 4.1: Block diagram of proposed hybrid network intrusion detection system.

Table 4.2: Proposed 11 features selected from NSL-KDD dataset by C4.5 decision tree after applying wrapper method based on 5-class classification.

Name of Feature	Data type	Description
service	Nominal	Destination service, e.g., http, telnet, etc.
flag	Nominal	Status flag of the connection.
src_bytes	Continuous	Number of data bytes sent from source to destination.
dst_bytes	Continuous	Number of data bytes sent from destination to source.
count	Continuous	Number of connection to the same host as the current connection in the past two seconds.
serror_rate	Continuous	% of connections that have SYN errors (same-host connections).
same_srv_rate	Continuous	% of connections to the same service (same-host connections).
diff_srv_rate	Continuous	% of connections to different services (same-host connections).
dst_host_same_srv_rate	Continuous	% of connections having the same destination host and using the same service.
dst_host_diff_srv_rate	Continuous	% of different services on the current host.
dst_host_serror_rate	Continuous	% of connections to the current host that have an S0 error.

4.3.1 Experimental setup and results

We have conducted our experiment in Weka 3.7.11 (Hall *et al.*, 2009) Java environment using Windows XP Professional as the test bed operating system, Intel i5 650 3.20GHz processor, 4GB of RAM. To test various intelligent technologies, NSL-KDD Dataset,

an enhanced edition of KDD'99 dataset is used containing 25192 training instances having 41 attributes/features and k-fold cross validation method is used for modeling an efficient NIDS.

Feature selection is an essential task for a high dimensional dataset in data mining environment. To remove those irrelevant features from classification rules, feature selection is done based on supervised or unsupervised filtering on the said dataset. Various feature selection algorithm like Chi-Squared Attribute Evaluator, Info Gain Attribute Evaluator, Gain Ratio Attribute Evaluator and Classifier subset evaluator employed with Naive Bayes, Decision Tree, RBF Network and Feature Vitality Based Reduction Method (FVBRM) (Mukherjee and Sharma, 2012) were used. To select the most optimal feature subset, the performance of various feature selection algorithm were studied and evaluated on each hybrid model.

4.3.1.1 Evaluation results based on 2-class classification

Table 4.1 describes the proposed features containing 13 out of 41 original features, selected by wrapper method-classifier subset evaluator employed with a C4.5 decision tree. Table 4.3 describes detail feature selection process for the proposed method. Table 4.4 shows eight datasets created by the said features selection algorithm, resulting 8 features for Naive Bayes, 15 features perform best for Chi-Squared, Info Gain and Gain Ratio, original 41 features are best for both Exp. No. 4 & 5, 13 features are optimal for the Decision Tree, only 5 features are selected by RBF Network and 25 features by FVBRM.

The detection performance of the proposed model was carefully evaluated and compared with different types of NIDS model including conventional individual and hybrid model. Each model was evaluated with eight datasets and records the comparable performance to choose the most optimal set for the model. After evaluation of each model, a new hybrid NIDS model that used combinations of AdaBoost with Decision

Tree classifier based on 13 features selected by subset classifier evaluator (Wrapper method) performed to be the best model scoring 99.7% attack detection accuracy with only 0.1% false alarm rate. The proposed model takes only 6.38 seconds to built the model having 271 leaves, 337 trees and weight of 0.43. From our studies and experimentation, it is observed that boosting the decision tree more than 10 iterations does not further improve the detection accuracy, but only increase model building time. Detail experiments resulted matrix is shown in Table 4.4 showing that the proposed hybrid intrusion detection system is better than the other conventional methods including both individual and hybrid classification system regarding attack detection accuracy, built time, false alarm and roc area.

Table 4.3: Proposed feature selection process for 2-class classification technique.

Feature selection process for the proposed method	
Step 1	Prepare NSL-KDD dataset training instances containing 25192 instances with 41 original features.
Step 2	Use Classifier subset evaluator employed with j48 Decision Tree to estimate merit of a set of attributes from 41 via predictive accuracy.
Step 3	Remove the last feature from the merit list and test with the tested model.
Step 4	If the classification result is improved on the tested model, repeat Step 3, Otherwise Stop.

Our proposed model is quite fast taking only 6.38 seconds to built the model, scoring 99.7% of recall rate, 99.8% of precision rate and high F-Value 99.8% correspondingly, and outperforms various individual and hybrid classification including conventional hybrid classification molde for the network intrusion detection system. Table 4.4 & Figure 4.2 shows that the combinations of Chi-squared; as a filter with Radial Basic Function (RBF) result in lowest performance resulting only 80.8% detection accuracy rate scoring highest degree false positive rate of 2%. The Weighted Avg. of 11.2%, 97.2% of precision rate, F-Value 88.2% and has more root mean square error of 0.281 respectively, and Exp No. 5 obtained the second best hybrid approach, which was a combination of Random forest with nested dichotomies and END, proposed by (Panda *et al.*, 2012), scoring the intrusion detection rate of 99.5% with false alarm rate of 0.1%

Table 4.4: Comparison between experimented models among various tested individual and hybrid models based on 2-class classification.

Exp. No.		1	2	3	4	5
Performance metrics		FS-NB Classifier-C4.5	FS-Chi2 Classifier-RBF Network	FS-Infogain Classifier-Spegasos	END+ ND+ C4.5	Hybrid END+ ND+RF (Panda *et al.* 2012)
No. of selected features		8	15	15	41	41
Detection Rate (%)	Normal	97.7	98	97.3	99.6	99.9
	Attack	98.7	80.8	95.9	99.6	99.6
	W. Avg.	98.2	90	96.6	99.6	99.7
False Positive Rate (%)	Normal	1.3	19.2	4.1	0.4	0.4
	Attack	2.3	2	2.7	0.4	0.1
	W. Avg.	1.8	11.2	3.5	0.4	0.3
F- Value Rate (%)	Normal	98.3	91.2	96.9	99.6	99.8
	Attack	98	88.2	96.4	99.5	99.7
	W. Avg.	98.2	89.8	96.6	99.6	99.7
Precision (%)	Normal	98.8	85.4	96.4	99.6	99.6
	Attack	97.4	97.2	96.9	99.5	99.9
	W. Avg.	98.2	90.9	96.6	99.6	99.7
Recall (%)	Normal	97.7	98	97.3	99.6	99.9
	Attack	98.7	80.8	95.9	99.6	99.6
	W. Avg.	98.2	90	96.6	99.6	99.7
RMS Error		0.13	0.281	0.184	0.0651	0.0489
Model built (seconds)		0.41	1.81	29.97	2.86	1.44
ROC area (%)	Normal	99.1	94.7	96.6	99.8	100
	Attack	99.1	94.7	96.6	99.8	100
	W. Avg.	99.1	94.7	96.6	99.8	100

Exp. No.		6	7	8	9	10
Performance metrics		Proposed Model FS-C4.5 Classifier-AB+C4.5	FS-Gainratio Classifier-SMO	FS-RBF Network NB	FS-C4.5 Classifier-RT	Con.model FS-FVBRM Classifier-NB (Mukherji and Sharma, 2012)
No. of selected features		13	15	5	13	25
Detection Rate (%)	Normal	99.9	97.9	96.5	99.6	95.9
	Attack	99.7	95	95.7	99.6	86.3
	W. Avg.	99.8	96.5	96.1	99.6	91.4
False Positive Rate (%)	Normal	0.3	5	4.3	0.4	13.7
	Attack	0.1	2.1	3.5	0.4	4.1
	W. Avg.	0.2	3.7	3.9	0.4	9.2
F- Value Rate (%)	Normal	99.8	96.8	96.4	99.6	92.3
	Attack	99.8	96.2	95.9	99.6	90.4
	W. Avg.	99.8	96.5	96.1	99.6	91.4
Precision (%)	Normal	99.8	95.7	96.3	99.7	88.9
	Attack	99.8	97.5	96	99.5	94.9
	W. Avg.	99.8	96.6	96.1	99.6	91.7
Recall (%)	Normal	99.9	97.9	96.5	99.6	95.9
	Attack	99.7	95	95.7	99.6	86.3
	W. Avg.	99.8	96.5	96.1	99.6	91.4
RMS Error		0.0426	0.186	0.196	0.063	0.283
Model built (seconds)		6.38	293.23	0.5	0.2	5.31
ROC area (%)	Normal	100	96.4	98.3	99.6	97.3
	Attack	100	96.4	98.3	99.6	96.9
	W. Avg.	100	94.9	98.3	99.6	97.1

, 0.0426 root mean square error, 99.7% F-value rate, 99.9% rate precision and 99.9% recall rate that is quite close to our proposed model Exp. No.6. After comparing those two models, we observed that our proposed model outperform Table 4.4 Exp. No. 5 in most important features of attack detection accuracy, root mean square error and complexity that are the main important key to designing an effective network intrusion detection system. Figures 4.3 & 4.4 shows detection performance based on ROC (Receiver Operating Characteristics) curves. ROC analysis as a standard tool for evaluating the performance of classification models in machine learning, it is a methodology for evaluating, comparing and selecting classifiers on the basis of their predicting performance.

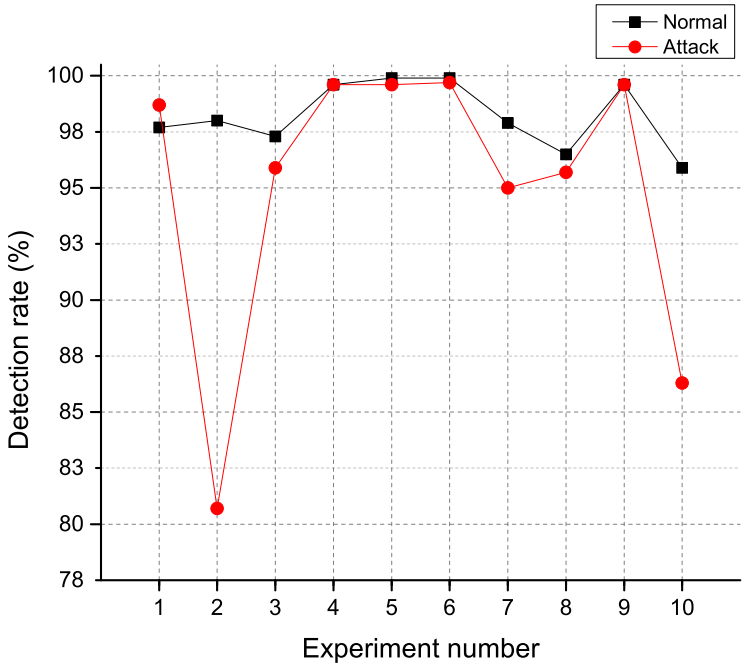

Figure 4.2: Detection accuracy obtained by different hybrid model on both normal and attack(2-class).

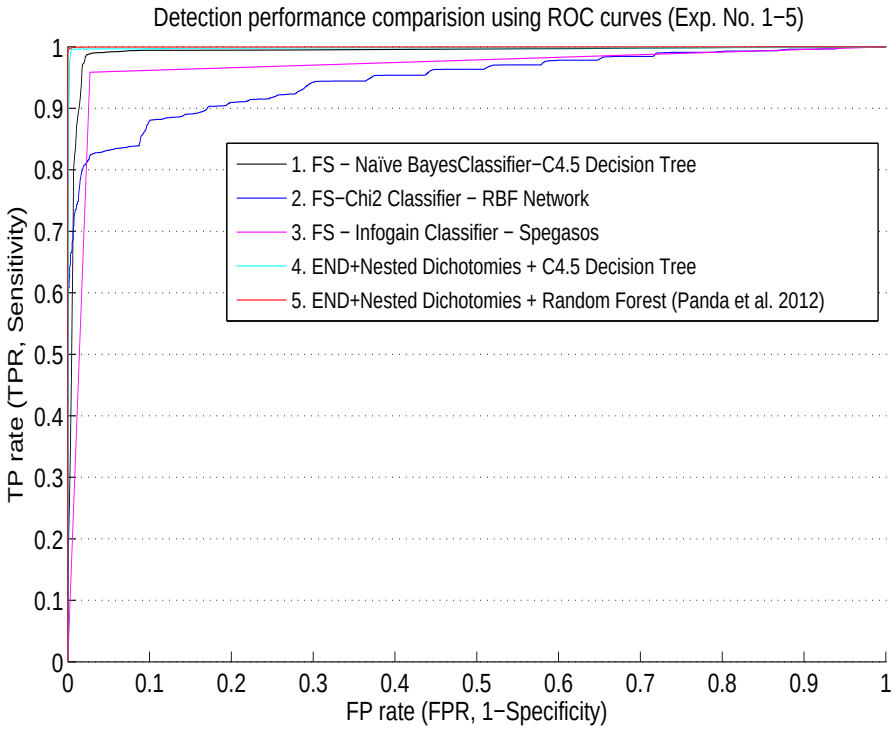

Figure 4.3: ROC curve showing Exp. No. 1-5 (2-class classification).

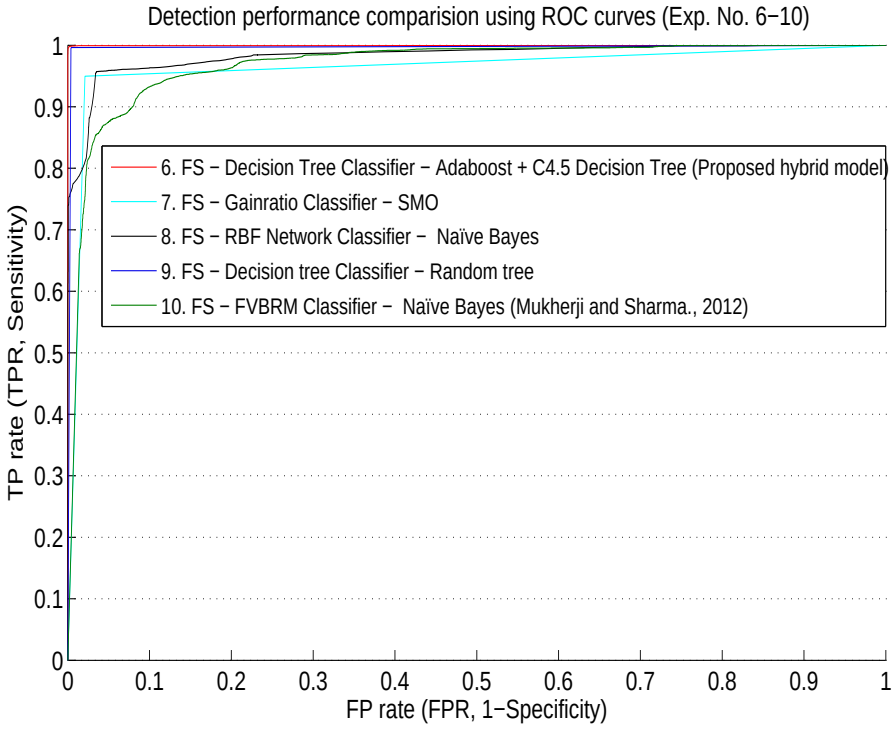

Figure 4.4: ROC curve showing Exp. No. 6-10 (2-class classification).

4.3.1.2 Evaluation results based on 5-class classification

For 5-class classification evaluation, separate datasets that include Normal, Probe, U2R, R2L, and DoS is used within the same experimental environment as 2-class classification strategies, the distributions of attack corresponding to attack groups are shown in Section 1 Table 1.7. Various feature selection and classification algorithms are evaluated and compared with their optimal performance, Table 4.3 shows detail procedure for proposed feature selection. The 11 selected features for the evaluation is shown in Table 4.2 using a C4.5 classifier based on Wrapper method.

Simulation results demonstrated in Table 4.5 confirmed that Experiment No. 7, which is a combination of the C4.5 decision tree with meta-classifier Adaboost using C4.5 decision tree as feature selection, still scores the highest detection accuracy of 99.8% with extremely low false alarm rate of 0.2% and lowest Root Mean Square Error (RMSE) of 0.0296. The proposed model outperforms Experiment No. 6 which is a combination of the same meta-classifier AdaBoost with C4.5 decision tree using info gain feature selection with wrapper method, having 99.8% detection accuracy with 0.2% false alarm and RMSE of 0.0299 which is quite close to Experiment No. 7.

The proposed system is quite fast and takes only 5.66 seconds to build the model, scoring 99.8% recall rate, 99.8% precision rate and high F-Value of 99.8%. The combination of the proposed methodology outperformed other tested models for the network intrusion detection system. Experiment No. 1 shows that, the combinations of decision tree as a filter with Radial Basic Function (RBF) network results in the lowest performance, scoring only 92% detection accuracy rate with the highest degree of false positive rate of 6.8%, precision rate of 91.4%, F-Value of 91.4% and RMSE of 0.1661.

Figures 4.5- 4.9 depicted the detection performance using Receiver Operating Characteristic (ROC) curve for our proposed model (5-class classification), showing each attack group with their corresponding area under ROC. An area under ROC is 0.9998 in Normal class (Figure 4.5), 1 in DoS (Figure 4.6), 0.9996 in Probe (Figure 4.7),

Table 4.5: Comparison results showing performance matrices (5-class).

Exp. No.		1	2	3	4	5	6	7
Performance matrics	Class	FS-C4.5 Classifier-RBF	FS-C4.5 Classifier-SMO	END+ND+C4.5	FS-RF END+ND+RF	FS-Chi2 Classifier-AB+C4.5	FS-Infogain Classifier-AB+C4.5	FS-C4.5 Classifier-AB+C4.5
No. of features		11	11	41	11	11	11	11
No. of class		5	5	5	5	5	5	5
Detection Rate (%)	Normal	96.1	97.9	99.9	99.9	99.9	99.9	99.9
	DoS	95.9	97.6	100	99.9	99.9	100	100
	R2L	1.9	70.8	89.5	90	90.9	91.4	92.3
	U2R	0	0	36.4	45.5	54.5	45.5	36.4
	Probe	61.2	91.4	99	98.6	99	99.2	99.3
W. Avg.(%)		92	96.9	99.7	99.7	99.7	99.8	99.8
False Positive Rate(%)	Normal	10.8	3.3	0.4	0.4	0.4	0.4	0.3
	DoS	2.4	1.2	0	0.1	0	0	0
	R2L	0	0.2	0	0	0	0	0
	U2R	0	0	0	0	0	0	0
	Probe	1.5	0.6	0.1	0	0	0	0
W. Avg.(%)		6.8	2.3	0.2	0.3	0.2	0.2	0.2
F - Value Rate (%)	Normal	93.5	97.5	99.8	99.8	99.8	99.8	99.8
	DoS	95.9	97.8	100	99.9	99.9	100	100
	R2L	3.7	70.8	93.5	93.8	93.8	93.6	94.1
	U2R	0	0	47.1	55.6	70.6	62.5	47.1
	Probe	69.3	92.7	99.2	99.2	99.3	99.4	99.4
W. Avg.(%)		91.4	96.9	99.7	99.7	99.7	99.8	99.8
Precision (%)	Normal	91.1	97.1	99.6	99.7	99.6	99.7	99.7
	DoS	95.9	97.9	100	99.8	99.9	100	100
	R2L	50	70.8	97.9	97.9	96.9	96	96
	U2R	0	0	66.7	71.4	100	100	66.7
	Probe	80	94	99.5	99.7	99.6	99.6	99.6
W. Avg.(%)		91.4	96.9	99.7	99.7	99.7	99.8	99.8
Recall (%)	Normal	96.1	97.9	99.9	99.9	99.9	99.9	99.9
	DoS	95.9	97.6	100	99.9	99.9	100	100
	R2L	1.9	70.8	89.5	90	90.9	91.4	92.3
	U2R	0	0	36.4	45.5	54.5	45.5	36.4
	Probe	61.2	91.4	99	98.6	99	99.2	99.3
W. Avg.(%)		92	96.9	99.7	99.7	99.7	99.8	99.8
Time to build model in seconds		12	101.22	43.56	10.92	6.27	5.83	5.66
RMS-Error		0.1661	0.318	0.0333	0.0327	0.0324	0.0299	0.0296

0.9994 in R2L (Figure 4.8) and 0.9794 in U2R class (Figure 4.9). After comparing various models, we observe that our proposed model outperforms other models in features like attack detection accuracy, false alarm rate, RMSE, and complexity, which are important in designing an efficient network intrusion detection system.

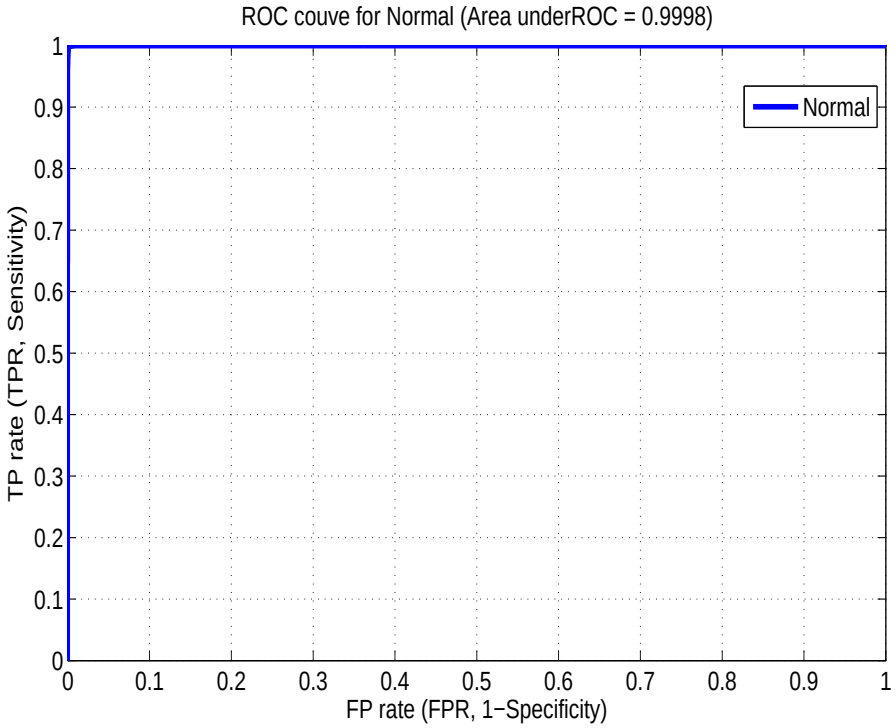

Figure 4.5: ROC Curve for Normal-class.

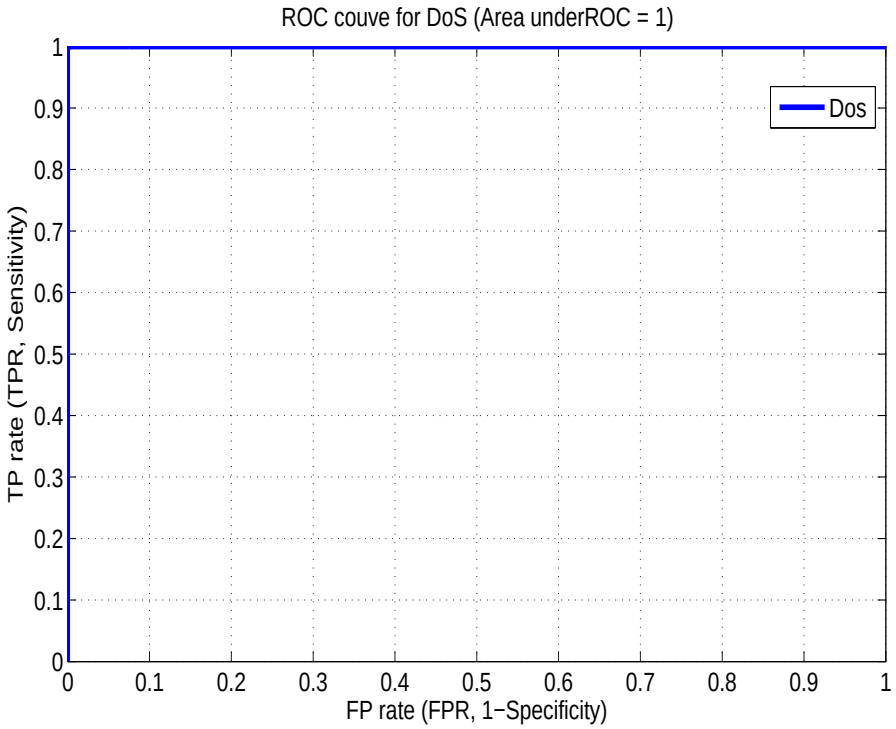

Figure 4.6: ROC Curve for DoS-class.

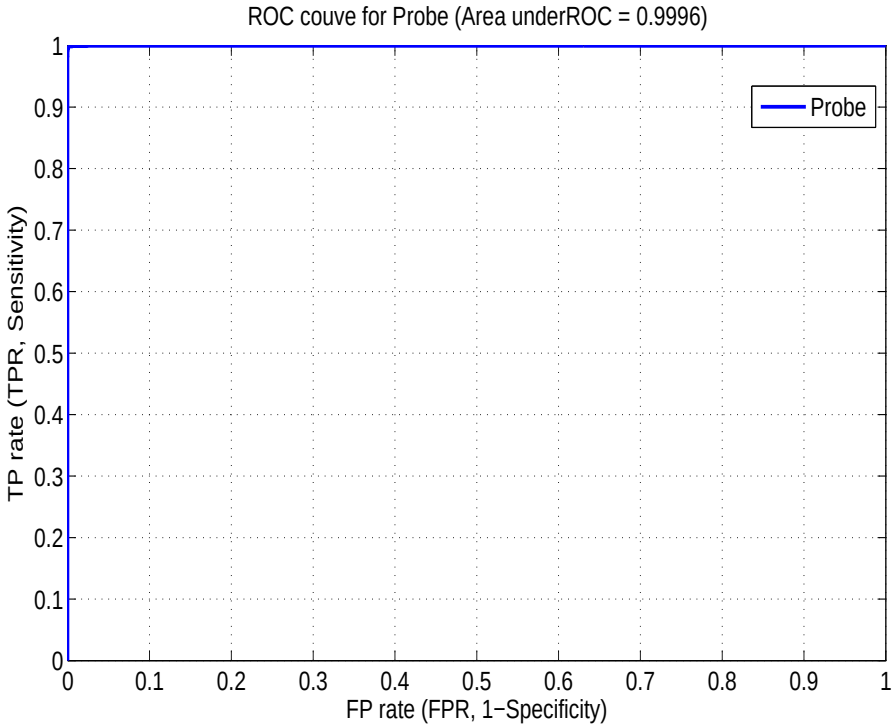

Figure 4.7: ROC Curve for Probe-class.

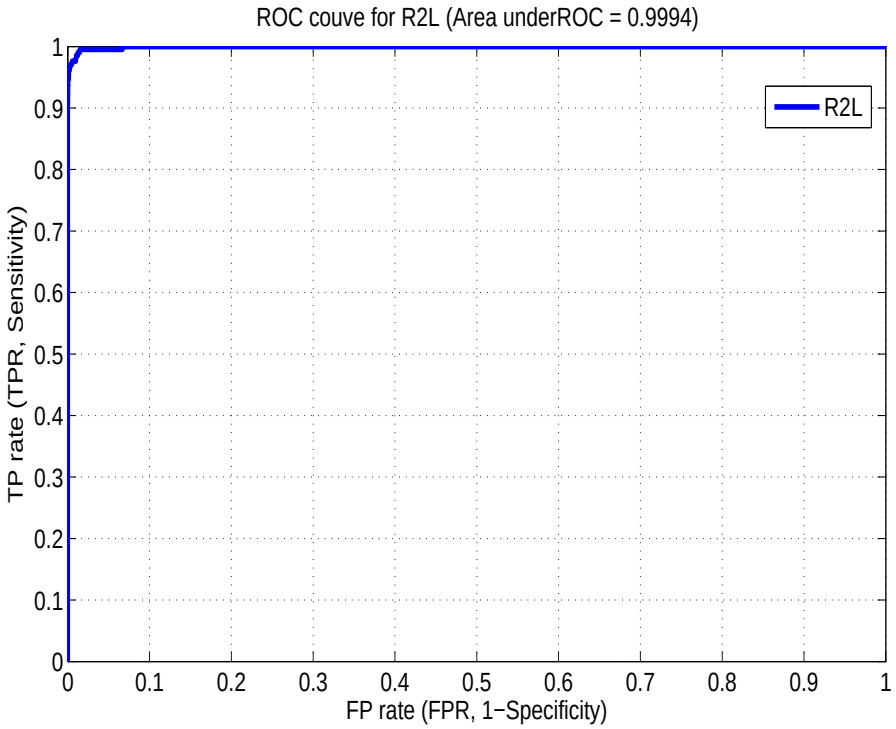

Figure 4.8: ROC Curve for R2L-class.

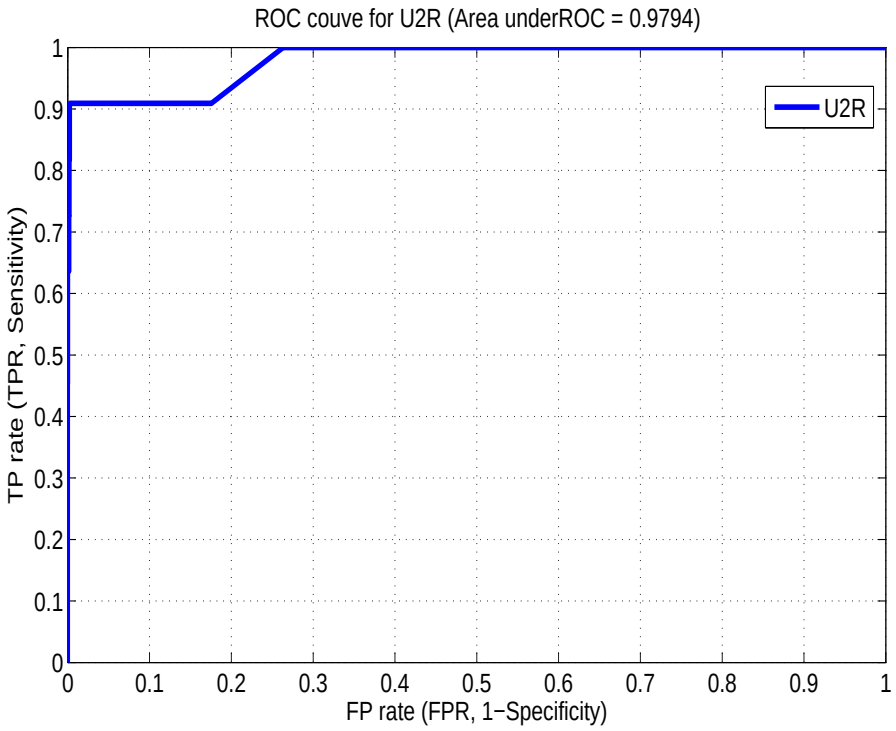

Figure 4.9: ROC Curve for U2R-class.

4.4 Conclusion

This section investigates and experiments with various novel hybrid intrusion detection technologies by using the different types of feature selection algorithm, based on supervised or unsupervised method along with classifier to make intrusion detection system to detect network intrusion while aiming higher degree of detection accuracy along with the lower degree of false alarm rate. NSL-KDD dataset was used for evaluating our proposed model to show its superiority over the other tested model including conventional method. The performance of each algorithm over the dataset was monitored to select the optimal classification accuracy with low false alarm.

Finally, we have concluded that DT as a feature selection with Wrapper method performed to obtain the best feature subsets for both 2 and 5-class classification strate-

gies. Simulation results (Table 4.4) proved that based on 2-class classification strategy (13 features shown in Table 4.1) AdaBoost with a C4.5 Decision Tree using ensemble method outperforms other existing approaches. The evaluation results of the proposed model result in 99.7% attack detection accuracy rate with Weighted Avg. of 99.8% along with 0.1% false alarm. The results were almost 100% rate of accuracy, which make the approach as most efficient among other different tested hybrid model.

The proposed model was again tested with various novel hybrid intrusion detection technologies based on 5-class classification strategies. Evaluation and comparison of various feature selection algorithms with supervised or unsupervised method were done to find the superiority of the proposed model.

Finally, as shown by Table 4.5 we have concluded that, for 5-class classification strategies, the same Adaboost with C4.5 decision tree, using DT as a feature selection with Wrapper method (11 features are shown in Table 4.2) scores 99.8% attack detection accuracy rate along with 0.2% false positive, which is almost 100% rate of accuracy. The evaluation results proved that the proposed approach was the most efficient among other different tested hybrid models.

Chapter 5

Fusion of misuse detection with anomaly detection technique for novel hybrid network intrusion detection system[4]

5.1 Introduction

In this chapter, we propose a new hybrid IDS model with feature selection that integrates misuse detection technique and anomaly detection technique based on a decision rule structure. The key idea is to take the advantage of Naive Bayes feature selection, misuse detection technique based on Decision Tree and anomaly detection based on One-class Support Vector Machine (OCSVM). First, misuse detection is built using single DT algorithm where the training data get decomposed into multiple subsets with the help of decision-rules. Then, anomaly detection models are created for each decomposed subset based on multiple OCSVM.

In the proposed model, NB and DT can find the best-selected feature to ameliorate the detection accuracy by obtaining decision rules for known normal and anomalies.

[4] ***Accepted for In. Conf. on Intelligent Computing, Communication & Devices (ICCD-2016), to be published by Advances in Intelligent Systems and Computing, Springer Book Series.***

Then, the OCSVM can detect a new attack that result in an improvement in detection accuracy of classification. The proposed new hybrid model was evaluated based on the NSL-KDD dataset, which is an upgraded version of KDD'99 dataset developed by the DARPA. Simulation results demonstrate that the proposed hybrid model outperform conventional models in terms of time complexity and detection rate with the much lower rate of false positives.

The purpose of this section is to develop a new hybrid approach to overcome the limitation of both misuse and anomaly technique for the network intrusion detection system. The proposed technique involves two steps, that hierarchically incorporate misuse detection technique based on C4.5 decision tree (Quinlan, 1986; Quinlan, 1987; Quinlan, 1993; Quinlan, 1996; Zhang *et al.*, 2006a; Sindhu *et al.*, 2012; Amor *et al.*, 2004; Premaratne *et al.*, 2009; Bouzida *et al.*, 2006; Kim *et al.*, 1995; Osei-Bryson, 2007) and anomaly detection technique using one-class SVM (Manevitz and Yousef, 2001; Sachs *et al.*, 2006; Shin et al,. 2005; Unnthorsson; Unnthorsson *et al.*, 2003; Gogoi *et al.*, 2011; Kim *et al.*, 2005; Chang *et al.*, 2011; Perdisci *et al.*, 2006; Song *et al.*, 2009; Zhang *et al.*, 2006b; Mohammed *et al.*, 2012) with feature selection using Naive Bayes (Benferhat and Tabia, 2005; Wu *et al.*, 2008; Koc *et al.*, 2012; Mukherjee and Sharma, 2012) to ameliorate the classification accuracy. The proposed model is also compared with other conventional technique, and the evaluation result demonstrated that the proposed technique outperforms the conventional models.

5.2 Proposed hybrid intrusion detection methodology

In this study, we propose a new hybrid novel network intrusion detection system with feature selection, that fuse misuse detection technique with anomaly detection technique based on a decision rules structure using misuse technique that results to decom-

posed subsets of the original datasets. Then, anomaly detection technique based on a multiple one-class SVM classifications for each decomposed subset was designed to detect an outlier from the normal baseline profile. The key idea of the proposed hybrid detection technique is to combine the advantages of misuse detection well-known for its low level false positive rate and anomaly detection technique that can detect novel or unknown attack traffic. Figure 5.1 illustrate the proposed model involving three stages: (i) feature preparation module (ii) misuse analyzer module and (iii) anomaly analyzer module. Details of these three modules are discussed in the following subsections.

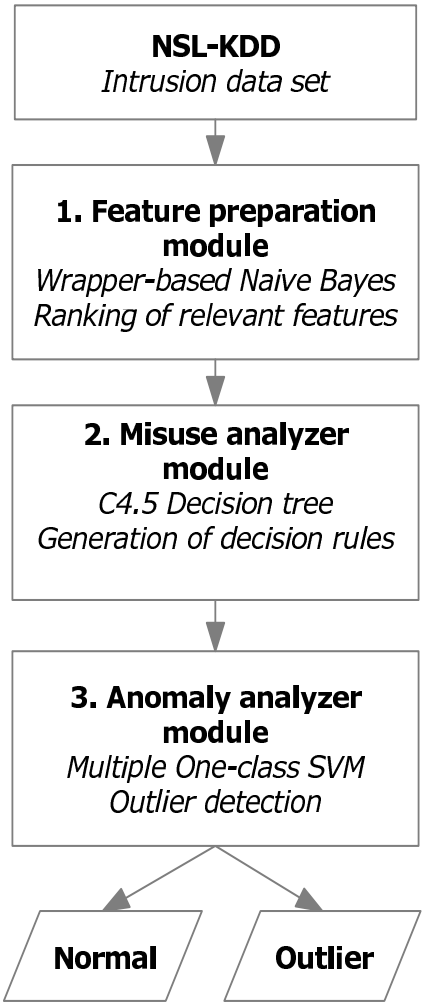

Figure 5.1: Proposed hybrid detection technique fusing misuse and anomaly technique.

5.2.1 Feature preparation module

Feature selection or extraction is the main technique used by recent research to improve the performance of a classification model (De Souza *et al.*, 2006; Louvieris *et al.*, 2013; Yang and Olafsson, 2006). The main purpose of feature selection is to reduce the computational time of the classification technique and improve the classification performance by removing redundant and unrelated attributes or to find a subset of feature for the proposed approach (Luo and Xia, 2014). This was accomplished by removing redundant or irrelevant feature out of its original set. Removal of significant or relevant feature might decrease the performance of the classification model regarding detection rate. However, some features might have a high degree of noise or might not have a contribution in any way. Removal of such features can significantly improve the detection accuracy and search speed of a classification model (Lin *et al.*, 2008b). Generally, there are two major types of feature selection method: filter method and wrapper method (Mukherjee and Sharma, 2012; Lin *et al.*, 2012).

In this module, feature selection based on Naive Bayes is used. The Naive Bayes classifier was used to identify relevant features and rank them accordingly to create a subset of features for each normal and attack instances. Naive Bayes classifier is well known for its robust against noise and missing values, high-speed training time, simplicity in approach with clear semantics and interpretation (Wu *et al.*, 2008). In this section, the NB classifier was used within a wrapper-based feature selection technique. The wrapper-based method uses the classifier to identify subsets of relevant features by evaluating the subsets and correlations of each feature (Yang and Olafsson, 2006) within a greedy-stepwise wrapper method. The proposed feature selection was evaluated on NSL-KDD intrusion dataset (Tavallaee *et al.*, 2009). We use two class classification strategies along with k-folds cross-validation (K-FCV) technique within the proposed

feature selection module. Classification approach of the Naive Bayes classifier is based on probabilistic method. It typically relies on assumption and assumes that variables are independent within each class or feature. More specifically, the presence of each particular class or feature is isolated to the absence or occurrence of some other characteristic. Given a vector of random variables data point, $X = (x_i, ..., x_n)$ denoting the observed attribute value, the Naive Bayes classification algorithm will predict that data point x test case belongs to the class c (i.e., normal or abnormal) random variable denoting the class of instances with the maximum posterior probability:

$$P(x \mid c) = \prod_{i=1}^{n} P(x_i \mid c) \tag{5.1}$$

5.2.2 Misuse analyzer module

In this module, a misuse detection model was design based on one of the most frequently use classification algorithm called Decision Tree (DT). DT utilizes a divide-and-conquer approach and recursively create a Decision Tree based on the greedy algorithm (Quinlan, 1986; Quinlan, 1987). DT consists of the root node, branches, parent nodes, child nodes and leaf nodes. It construct a tree-like structure in a series of Boolean formation 'yes' or 'no' until no more related branches can be derived. A node in a tree denotes dataset attributes, and every child node derives labeled branches with the possibilities of attribute values from the corresponding node called parent node (Kim *et al.*, 2014). A branch connects either one or two nodes or a leaf, and each labeled leaf node represents the classification value. A classification for new data is obtained starting from the root node and moving down towards the branches until and unless a leaf node is found, a decision rule has been created to categorize the data point according to the value of a feature.

DT calculates and selects the maximum value of information gain (Eqn. 5.2), to choose those feature having a maximum value of information gain. DT starts from

the root node and then divide the dataset into more subsets until no more relevant branches can be derived or, until all data in the current subset fit into the same class. If $C_1 ... C_n$ denotes classes, and T is for DT representing a leaf that identifies class C_i, then the information gain ratio has been calculated as below (Quinlan, 1993):

$$InfoGain(F) = Info(Y) - Info_x(G) \tag{5.2}$$

$$Info(Y) = -\sum_{i=1}^{k} \frac{freq(C_i, Y)}{\mid Y \mid} \times log_2\left(\frac{freq(C_i, Y}{\mid Y \mid}\right) \tag{5.3}$$

$$Info_x(G) = \sum_{j=1}^{n} \frac{\mid G_j \mid}{\mid G \mid} \times Info(G_j) \tag{5.4}$$

Here, C_i denotes classes where i starts from 1 to k; k is the maximum number of classes, $\mid Y \mid$ represents total cases amount of trainset. The standard quantity of information required to categorize the case in class Y is represented by $Info(Y)$, for partition G the $Info_x(G)$ denotes those features F having an applicable amount of information. The total amount of cases integrated in C_i is represented by $freq(C_i, Y)$, n represents total quantity of outputs intended for F, G_j represent T subset in relation to j^{th} term, and $\mid G_j \mid$ denote the total cases amount of the G_j subset.

In this work, a misuse analyzer module for the proposed hybrid model was designed based on DT, using the normal and attack data to train the model. DT divides input data into more decomposed regions based on the information gain. The decomposed subsets created by this module serves as input to the next level anomaly based classification technique. To get the optimal performance of a DT algorithm, it is required to set the necessary parameters like confidence value, the number of folds for cross-validation and minimum cases (Quinlan, 1993; Quinlan, 1996).

5.2.3 Anomaly analyzer module

In this module, an anomaly based detection technique was designed based on one-class SVM classification algorithm (Chang and Lin, 2011). It is popularly known for its outlier detection ability for various application. Classification algorithm that make use of only one class label, mainly normal class is called one-class classifier. A one-class classifier needs to be trained before it can classify any data points.

Based on the normal profile decomposed structure from the misuse analyzer module, we trained multiple OCSVM classifiers to create a multiple normal baseline profiles, throughout the training phase each model locate the decision margin separation between the inlier and outlier instances. In the inspection stage, the decision function of each one-class model detects outlier connections that could be an attacked. The outlier can be known or unknown attack while the inliers are those normal activities.

Vapnik (1995) first proposed the SVM model based on the idea of increasing dimensionality of the binary class samples so that they can be separable. The basic idea of SVM is to find a maximum hyperplane to separate binary samples from the same class inside it. The extension of SVM, OCSVM model, was proposed by (Schlkopf *et al.*, 2001) it was formulated to find a maximum hyperplane that separates a desired portion of the one-class training instances in feature space (F) from its origin. In the testing phase, the outliers of a testing instance have been detected based on this hyperplane, to determine which class the instance belongs.

Let us consider a training data $x_1, ..., x_l \in X$, where X is the original space and the number of instances denoted by l. Let Φ be a feature map $X \rightarrow F$ to locate a hyperplane that best separates training data pattern from the original space X, which transforms the instances non-linearly to the feature space from its original so as to establish the best hyperplane in F. The one-class SVM is formulated as following:

$$\min_{w,\xi,p} \frac{1}{2}\|w\|^2 + \frac{1}{vl}\sum_{i=1}^{l}\xi_i - p \tag{5.5}$$

subject to

$$(w.\Phi(x_i)) \geq p - \xi_i,$$

$$\xi_i \geq 0, i = 1, ..., l$$

where, p represents the distance of the origin from the hyperplane, w is vector orthogonal to the hyperplane, ξ_i is a slack variable of a vector $\xi_i, ..., \xi_l$. As mentioned in (Perdisci *et al.*, 2006) there are some difficulty in calculations of the feature space that is caused by curse of dimensionality, and then, simple kernel function $k(x, y) = (\Phi(x).\Phi(y))$, such as Gaussian $k(x, y) = e^{-\gamma\|x-y\|^2}$ were utilize to compute the feature space (Kim *et al.*, 2014). Each instance (n) were generally tested by a function $f(n)$ which return the decision results of which side of the hyperplane each encountered instances falls on in the feature space, formulated as:

$$f(n) = sgn\left(\sum_{i=1}^{l}(\alpha_i k x_{i(n-p)})\right) \tag{5.6}$$

If the $f(n)$ return a positive value, it means the encountered instance belongs to the inlier feature space, but if there is a negative result, it says the encountered instance is an outlier. In this section, the term inlier indicates the normal activity, as this anomaly module is built based on only the normal profile from the decomposed subset of the misuse module, while outlier indicates those attack that might harm the system, i.e., known or unknown attack.

5.3 Simulation results

The proposed hybrid system is evaluated carefully based on the NSL-KDD intrusion dataset (Tavallaee *et al.*, 2009). The NSL-KDD dataset was proposed after resolving such problems mentioned in Chapter 1 and is found to be more efficient to have more realistic environment compared to the KDD'99 intrusion dataset. To evaluate a performance of the proposed hybrid system Weka 3.7.11 (Hall *et al.*, 2009) and LibSVM Matlab (Chang and Lin, 2011) is used. The proposed hybrid system was evaluated as follows.

First, feature selection is done on the preprocessed dataset from NSL-KDD data, we organized the evaluation data by modifying the original dataset KDDTrain+ _20Percent with KDDTrain+, KDDTest21 with KDDTest+. Here we transform the ordinary multi-class dataset into two class dataset so that they can be used for evaluation of the proposed hybrid system. This modification is done to include unknown attack to the test dataset. Unknown attack means attacks traffic data that has neither been used for training nor been seen by the network before. Then, feature selection module selects those relevant features from the original 41 characteristics of the NSL-KDD dataset based on wrapper feature selection with k-fold cross validation technique. The wrapper-based method uses the Naive Bayes classifier to identify subsets of relevant features by evaluating the subsets and correlations of each feature within a greedy-stepwise wrapper method. Table 5.1 illustrates the feature sets for the proposed hybrid system. The main idea of feature selection in this section is to reduce the computational time of the classification algorithm and improve the classification performance by removing redundant and unrelated attributes. Evaluation results (Table 5.2) demonstrate that the time complexity of the proposed misuse detection is reduced to 18.48 s compared to the conventional method (Kim *et al.*, 2014), achieving much more detection rate with an acceptable rate of false alarm.

Once the feature set is identified, they were taken onto the misuse classification

Table 5.1: Selected feature set for proposed technique based on Naive Bayes feature selection.

Feature name	Data type	Description
duration	Continuous	Length of the connection.
protocol_type	Nominal	Connection protocol.
service	Nominal	Destination service.
src_bytes	Continuous	Bytes sent from source to destination.
su_attempted	Nominal	1 if su root command attempted; 0 otherwise.
count	Continuous	Number of connections to the same host as the current connection in the past two seconds.
srv_count	Continuous	Number of connections to the same service as the current connection in the past two seconds (same-service connections).
dst_host_srv_count	Continuous	% of connections having the same destination host and using the same service.
dst_host_same_srv_rate	Continuous	% of connections having the same destination host and using the same service.
dst_host_serror_rate	Continuous	% of connections to the current host that have an S0 error.
dst_host_rerror_rate	Continuous	% of connections to the current host that have an RST error.

stage where the data get divided into more decomposed subsets. The basic idea of decomposing the normal data into multiple subsets are that the conventional hybrid anomaly model intended to profile the normal activity based on an outlier detection technique. However, there is multiple normal activities profile based on the types of service, protocol, src_byte, etc. in a real environment. The anomaly analyzer based on OCSVM can be extremely responsive to the trainset and may lead towards a high degree of false alarm rate. To resolve this situation, the normal dataset get decomposed into more appropriate subsets and then a single oneclass SVM classification model is built for each subset. After applying the identified features to a C4.5 decision tree, the original dataset gets decomposed into multiple subsets based on decision rules created by the DT classification algorithm (Table 5.3) based on the information gained from each feature.

Table 5.2: Misuse detection based on C4.5 DT.

Methods	Conventional Model (Kim *et al.*2014)	Proposed model
Feature selection	NULL (41f)	Wrapper-based Naive Bayes(11f)
Time taken to build model(s)	23.22	4.74
TPR	99.75	99.99
FPR	0.3	0.10%
RMS error	0.0492	0.0107

Table 5.3: Decision rules obtained by proposed misuse detection technique based on C4.5 decision tree.

No.	Decision rules	Type
1	src_bytes>28 and src_bytes<=333 and dst_host_srv_count <=205 and service=domain_u	Normal
2	src_bytes<=28 and dst_host_srv_count<=89 and count <=6 and src_bytes>1 and count<=1	Normal
3	src_bytes>28 and src_bytes<=333 and dst_host_srv_count <=205 and service=ftp and dst_host_srv_count> 2	Normal
4	src_bytes>28 and src_bytes<=333 and dst_host_srv_count <=205 and service=ftp_data and dst_host_same_srv_rate<=0.94	Normal
5	src_bytes<=28 and dst_host_srv_count<=89 and count <=6 and service=http and dst_host_serror_rate<=0.5 and dst_host_same_srv_rate>0.25	Normal
6	src_bytes>28 and src_bytes<=333 and dst_host_srv_count <=205 and service=http	Normal
7	src_bytes<=28 and dst_host_srv_count>89 and src_bytes <=0 and dst_host_serror_rate<=0.7 and service=http	Normal

Continued on next page

Table 5.3 – continued from previous page

No.	Decision rules	Type
8	src_bytes>28 and src_bytes<=333 and dst_host_srv_count <=205 and service=other and src_bytes<=145 and dst_host_serror_rate<=0	Normal
9	src_bytes>28 and src_bytes<=333 and dst_host_srv_count <=205 and service=private and src_bytes<=156 and src_bytes>102	Normal
10	src_bytes>333 and src_bytes>334 and service=ecr_i and service=ftp_data and dst_host_same_srv_rate<=0.98 and dst_host_serror_rate <=0.01	Normal
11	src_bytes>28 and src_bytes<=333 and dst_host_srv_count <=205 and service=telnet and dst_host_rerror_rate<=0.27	Normal
12	src_bytes>333 and src_bytes>334 and service=ecr_i and service=ftp_data and dst_host_same_srv_rate<=0.98 and dst_host_serror_rate<=0.01	Normal
13	src_bytes>28 and src_bytes<=333 and dst_host_srv_count <=205 and service=urp_i and dst_host_same_srv_rate<=0.35	Normal
14	src_bytes<=28 and dst_host_srv_count<=89 and count>6 and src_bytes<=10	Attack
15	src_bytes>333 and src_bytes>334 and service=ecr_i	Attack
16	src_bytes<=28 and dst_host_srv_count<=89 and count<=6 and service=http and dst_host_same_srv_rate<=0.06	Attack
17	src_bytes>333 and src_bytes<=334 and service=ftp_data	Attack
18	src_bytes>40494 and dst_host_same_srv_rate<=0.98 and duration>1398 and service=http	Attack

Continued on next page

Table 5.3 – continued from previous page

No.	Decision rules	Type
19	src_bytes<=28 and dst_host_srv_count<=89 and count <=6 and service=eco_i	Attack
20	src_bytes>334 and src_bytes<=40494 and service=telnet and dst_host_same_srv_rate>0.75 and su_attempted<=0	Attack
21	src_bytes<=28 and dst_host_srv_count<=89 and count>6 and src_bytes>10 and protocol_type=udp	Attack
22	src_bytes>333 and src_bytes>334 and service=ecr_i and service=ftp and duration<=13	Attack
23	src_bytes<=28 and dst_host_srv_count>89 and src_bytes>0 and service=eco_i	Attack
24	src_bytes<=28 and dst_host_srv_count<=89 and count <=6 and service=ftp_data and src_bytes<=4 and srv_count<=6 and duration<=2511 and dst_host_serror_rate>0.51	Attack

It is been observed that the classification results of the proposed system outperform the conventional model (Kim *et al.*, 2014) in terms time complexity, detection rate, false positive rate and root mean square error (Tables 5.3 & 5.4, Figures 5.2, 5.3 & 5.4). To get the optimal performance of the DT algorithm, necessary parameters are carefully set until an optimal result is obtained. A consistent detection rate of 99.99% with only 0.1% false alarm rate with 0.0107 RMS errors were achieved after setting the minimum instance per leaf approx to 1.0% and the confidence factor of 1%. It also found that the time complexity of the misuse model decreases up to 79.6% as compared to the conventional method.

Once the decomposed structure is established, a multiple OCSVM classification algorithm model is build based on the each normal activity. Conventional anomaly

detection system built a classification algorithm based on only the normal traffic information. The proposed hybrid model also followed this method; however a decomposed structure on the normal training instances is proposed to improve the normal activity profiling performance of the anomaly module. Because the whole normal traffic has a range of normal associations, so there is a problem on profiling those model accurately for an anomaly technique, and can degrade performance (Song *et al.*, 2009). Each decomposed subset was tested on multiple one-class SVM with normal traffic and creates a decision function that describes the normal behavior that separates the inlier from the outlier. Each one-class SVM model was carefully evaluated based on the critical parameter γ, with variation from 0.001 to 1 and compared with the conventional model. Too narrow or too broad in the parameter γ may affect the detection problem on detecting unknown attack for OCSVM. In this experiment, it is being observed that the best value for parameter γ is 0.01. An increase in parameter γ results to an elevation of detection performance for OCSVM model (Figure 5.2). The detection performance was investigated based on anomaly detail; this was achieved by testing various kernels like Linear, Polynomial, Sigmoid and Gaussian $k(x, y) = e^{-\gamma\|x-y\|^2}$. In Figure 5.2, it has been observed that Gaussian kernel outperforms other kernels regarding detection accuracy of 99.98% along with a much lower false positive rate of 0.1%.

However, it requires much time complexity compared to others (Figure 5.3), but the main focus of this study is to improve the ability of unknown attack detection rate along with an acceptable rate of false alarm and time complexity. As a result, training and testing time are calculated based on Weka and Matlab application (Table 5.4). As expected the time complexity of the proposed model is improved to 37.97 s (training time) and 6.71 s (testing time) which is shorter than the conventional model 280.99 s (training time) along with 19.17 s (testing time). So, we can conclude that the time complexity of the proposed model is improved up to 86% (training time) with 65% (testing time) compared to the conventional model. This was achieved due to the

application of feature selection procedure and decomposition of the original dataset into more decomposed subsets. Since each decomposed data structure is lesser complex as compared to the original data structure, a single one-class SVM model for the whole original data pattern can be more flexible than multiple OCSVM models for each decomposed subsets.

Table 5.4: Comparison of detection time between conventional model and the proposed new model.

Decomposed subset @ Decision rules	# of training instances	Training time(s)	# of testing instances	Testing time(s)	# of support vectors
Subset-1	2507	1.39	499	0.06	251
Subset-2	10681	26.41	9840	5.5	2136
Subset-3	261	0.02	1389	0.014	48
Subset-4	1399	0.42	834	0.057	126
Subset-5	685	0.17	7942	0.42	130
Subset-6	68	0.01	907	0.005	7
Subset-7	176	0.02	164	0.004	62
Subset-8	7564	8.8	434	0.18	2042
Subset-9	1001	0.41	390	0.29	230
Subset-10	209	0.03	269	0.005	75
Subset-11	519	0.11	339	0.013	36
Subset-12	34	0.01	49	0.0006	12
Subset-13	685	0.17	7942	0.42	199
Proposed model	25789	37.97	30998	6.71	5353
Conventional model (Kim *et al.* 2014)	25789	280.99	30998	19.17	7891

ROC curve performance (Figure 5.4) and comparison (Table 5.4) demonstrate that the proposed model outperform the conventional model regarding detection rate of unknown attack, training and testing time complexity. The average number of encountered support vector for the proposed model is reduces up to 32% compared to the conventional model. Since the number of support vector affects the complexity of the testing computation, and a lesser number of avg_SVs is suggested to improve the time complexity of classification algorithm.

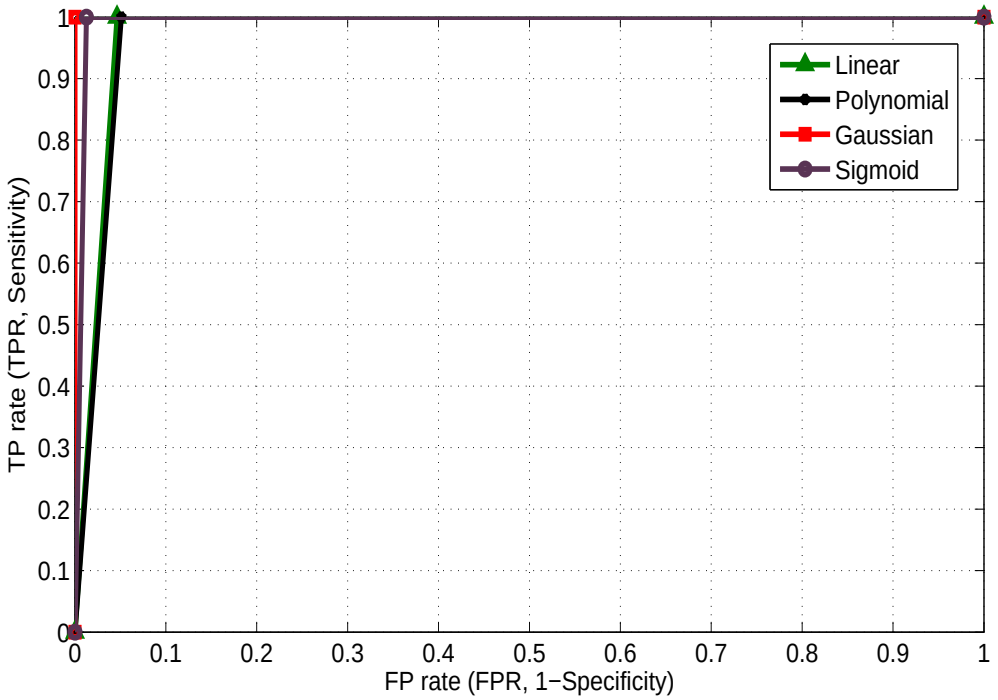

Figure 5.2: ROC curve performance of various kernel in OCSVM.

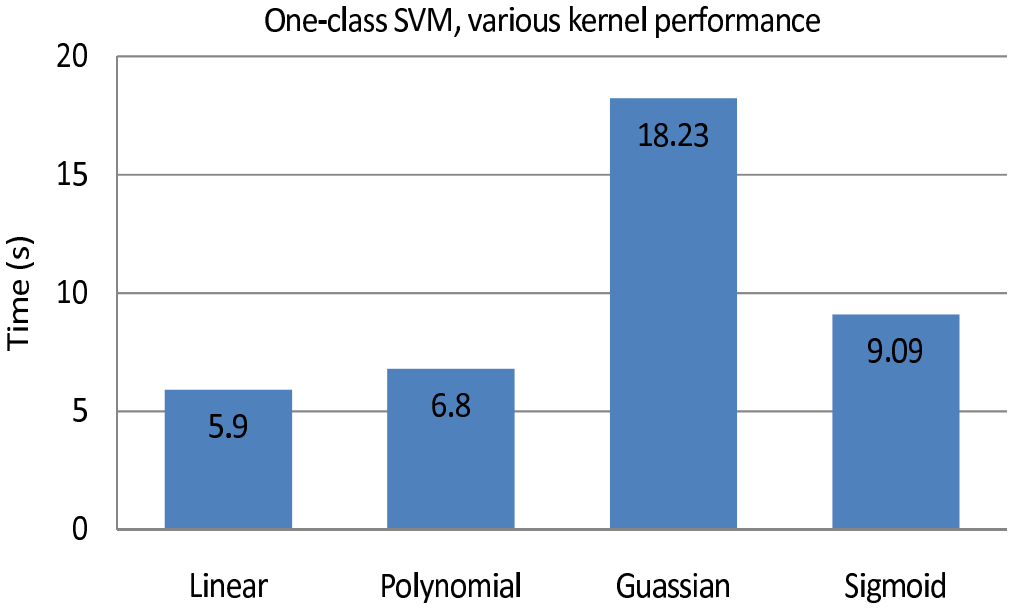

Figure 5.3: Performance of various kernel (Time complexity) in OCSVM.

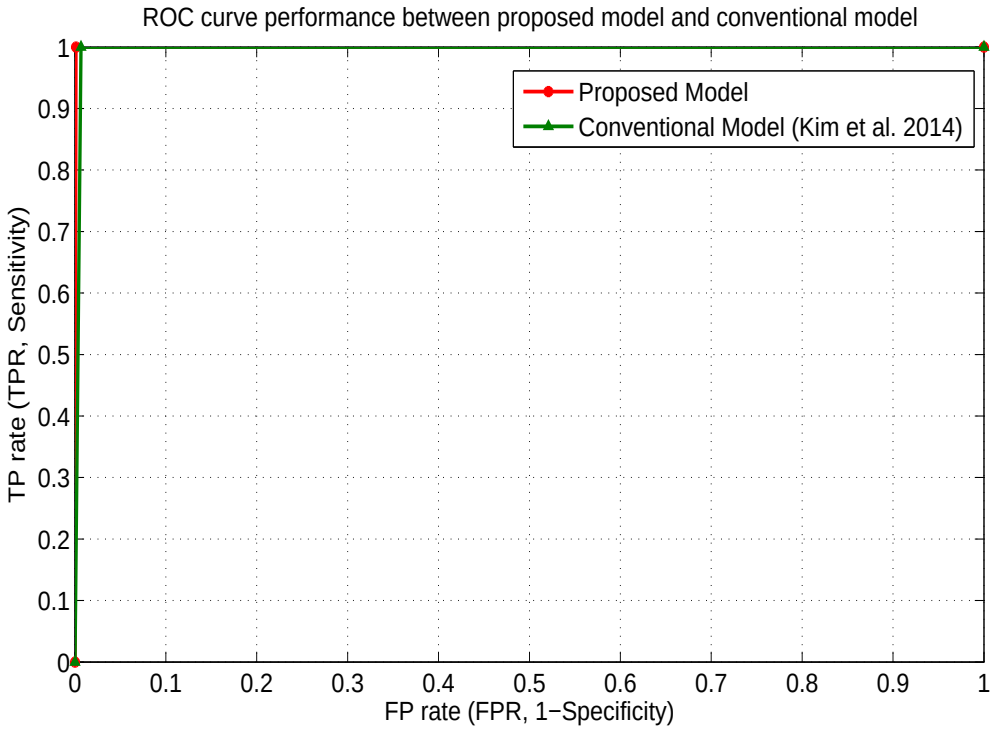

Figure 5.4: ROC curve comparison of proposed model with conventional model.

5.4 Conclusion

In this study, we propose a new hybrid network intrusion detection system, which integrates misuse based detection system with anomaly-based detection technique. Our proposed model include wrapper-based feature selection to improve the intrusion detection system regarding detection performance on an unknown attack and to improve training/testing time complexity of a classification algorithm. The key idea of our proposed hybrid model is to combine the advantages of misuse detection, well-known for its low level false positive rate and anomaly detection techniques, which can detect novel traffic activity.

First, a Naive Bayes classifier based on wrapper method is used to identify relevant

feature and rank them accordingly to create a subset of features, this attempt focus on the particular attribute that describe both attack and normal activity of intrusion data. This reductions of feature also filtered out those non-relevant features that associate with noisy data and also decrease the computational power. Then, a misuse based detection model is design based on the C4.5 decision tree that decomposed the original dataset into smaller decomposed subsets. Then, multiple anomaly based detection model was created based on multiple OCSVM for each decomposed subsets. The anomaly-based used the normal activity for profiling the normal baseline, any deviation from the model is treated as an outlier. Outliers could be known or unknown attack traffic. The experimental results demonstrate that the proposed system can improve the intrusion detection regarding novel attack detection and time complexity of intrusion detection system.

Finally, we have concluded that our proposed new hybrid IDS technique results in improvement on both misuse and anomaly based. The time complexity of misuse and anomaly based has been reduced up to 86.5% and 65% with an overall of 85.1%. It is also encountered that the average number of support vectors has been reduced up to 32% with a high detection rate of 99.98% along with an acceptable rate of only 0.1% false alarm. Therefore, the evaluation results proved that the proposed approach was more efficient compared to conventional hybrid model.

Chapter 6

Summary and Conclusion

The key idea of the present thesis is to combine the advantages of misuse detection well-known for its low level false positive rate and anomaly detection techniques that can detect novel or unknown attack traffic. Various hybrid IDS classification that use various techniques (i.e., ensemble, serial, parallel or fusion methods) are studied and discussed/improved, and proposed a new method of a hybrid classification system for IDS.

Chapter 1 is the general introduction that includes basic definitions of IDS, various intrusion methods and its detection system, different types of IDS, types of detection approaches, detail analysis of KDD'99 intrusion dataset based on features and review of literature.

In **Chapter 2**, the performance of various classification algorithm has been compared and evaluated based on the KDD'99 dataset, NSL-KDD dataset and a noise-added dataset with 10% & 20% noisy data added to NSL-KDD dataset. Various classification algorithms like NB, SVM, RBF Network, SMO, RBF classifier, Spegasos, BN, VP, SGD, JRIP, J48, RF, END, NB-Tree, NN (SOM) and DT from various algorithm family were tested and compared.

The comparative studies show that those recent studies from various classification algorithms in the absence of noisy environment or noise free dataset could misinform

about evaluation performance to a much higher degree. The algorithm that performs well on the original KDD'99 dataset does not produce the same result with NSL-KDD, 10% noisy data and 20% noisy data, which proves that the NSL-KDD dataset represents more realistic environment for evaluation of classification algorithms compared to the KDD'99 dataset. Among various tested classification algorithms, JRip and J48 were advanced compared to the other tested algorithms followed by RF, END and NB-Tree. However, Neural Network (SOM) is far more superior to all the others regarding robustness to a noisy environment.

The studies of feature selection evaluation based on Performance-based Method of Ranking shows that each classification algorithm has unique combinations of feature subset to give optimal performance. Empirical results demonstrate that the feature subsets selected by each classification algorithm are different from each other; dependency of each feature subset depends on the type of classification algorithm selection. It is proved that each classification algorithm has its unique combination of feature subsets.

In **chapter 3**, a new hybrid network intrusion detection system using two-stage (Anomaly-Misuse) hybrid classification technique have been proposed. The Stage-1 used one SVM to detect traffic anomalies that can be attack and the stage-2 used one ANN that classifies attacks if they exist. The evaluation result shows that high detection rate 99.97% with a low false positive rate of only 0.19% is achieved on stage-1 anomaly detection and 99.9% detection accuracy with only 0.1% false positive rate at stage-2. This was achieved through the design of a classification model using SVM with radial basis kernel function at the first stage (Anomaly) and neural network using multi-layered feed forward neural network with Resilient back propagation at the second stage (Misuse). The proposed model helps in reducing the computational complexity in both stages and outperformed single-stage classification technique based on 5-class classification. The evaluation results in 99.95% detection accuracy with a low false

positive rate of only 0.2% while individual classification using SVM results in 98.72% accuracy along with 0.7% false positive and single-stage ANN results in 86% detection rate with the relatively high false positive rate of 5.6%.

The proposed new hybrid model is found to be comparative with the recent conventional model, i.e., Depren *et al.* (2005) results 99.8% AC along with 1.25% FPR, Kim *et al.* (2014) results in 99.1% average AC with 1.2% FPR, Ghanem *et al.* (2015) results in 96.1% AC along with high degree of 3.3% FPR, Yousef *et al.* (2014) lead to 94.2% AC with high probability of 5.8% FPR, Khan and Khan (2008) that results in 84.8% AC along with 0.1% FPR. The compared conventional model are various proposed hybrid model for IDS that uses the same KDD99/NSL-KDD datasets for evaluation and help us to conclude that our proposed hybrid approach delivers better detection accuracy among the existing models.

Chapter 4 investigate and experimented on various novel hybrid intrusion detection technologies (Ensemble) by using the different types of feature selection algorithm, which uses supervised or unsupervised method along with classifier to make intrusion detection system to detect network intrusion while aiming higher degree of detection accuracy along with the lower level of false alarm rate. We have concluded that Decision Tree as a feature selection based on Wrapper method performed to obtain the best feature subsets for both 2 and 5-class classification strategies. Simulation results proved that based on 2-class classification strategy (13 features) AdaBoost with a C4.5 Decision Tree using ensemble method outperforms other existing approaches. The proposed model results in 99.7% attack detection accuracy with Weighted Avg. of 99.8% along with 0.1% false alarm. The results make the approach as most efficient among other different tested hybrid model.

The proposed model was again tested with various novel hybrid intrusion detection technologies based on 5-class classification strategies. We have concluded that, for 5-class classification strategies, the same Adaboost with a C4.5 decision tree, using

DT as a feature selection based on wrapper method (11 features) scores 99.8% attack detection accuracy along with 0.2% false positive rate. The evaluation results proved that the proposed approach was the most efficient among other different tested hybrid models.

Chapter 5 introduced a new hybrid network intrusion detection system, which integrates misuse based detection system with anomaly-based detection technique, and include wrapper-based features selection to improve the intrusion detection system regarding detection performance on an unknown attack, and improve training/testing time complexity of a classification algorithm.

First, a Naive Bayes classifier based on wrapper method is used to identify relevant features and rank them accordingly to create a subset of features, this attempt filtered out those non-relevant features that associate with noisy data and also decreases the computational power. Then, a misuse based detection model is design based on the C4.5 decision tree that decomposed the original dataset into smaller decomposed subsets. Then, multiple anomaly based detection model was created based on multiple OCSVM for each decomposed subsets. The experimental results demonstrate that the proposed system can improve the intrusion detection regarding novel attack detection and time complexity of intrusion detection system. We have concluded that our proposed new hybrid IDS technique results in improvement in both misuse and anomaly based. The time complexity of misuse and anomaly based has been reduced up to 86.5% and 65% with an overall of 85.1%. It is also encountered that the average number of support vectors has been reduced up to 32% with a high detection accuracy of 99.98% along with an acceptable rate of 0.1% false alarm.

These studies and evaluation results encourage us for further research on various hybrid IDS techniques, creating self-captured dataset with 2 and 5 class, and exploration of various classification algorithms against real network traffic along with the effect of various noisy data over machine learning algorithm may be the focus of our future

works. A challenging issue like an application of ML and DM in various Medical area specifically in CANCER Prognosis, Diagnosis and Prediction based on susceptibility, recurrence and survival may fall under our future research plan.

Bibliography

Alexandre, L., Campilho, A. and Kamel, M. (2000). Combining independent and unbiased classifiers using weighted average, *In: Proc. Int. Conf. on Pattern Recognition*: 2495–2498.

Amiri, F., Yousefi, M.M.R., Lucas, C., Shakery, A. and Yazdani, N. (2011). Mutual Information-Based Feature Selection for Intrusion Detection Systems, *Journal of Network and Computer Applications*. **34(4)**: 1184-1199.

Amor, N.B., Benferhat, S. and Elouedi, Z. (2004). Naive Bayes vs decision trees in intrusion detection systems, *In: Proc. of the Int. ACM Symposium on Applied Computing*: 420-424.

Amorso, E. (1999). *Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response*, 1st edn, AT&T Inc.

Anderson, R. (2001). *Security Engineering: A Guide to Building Dependable Distributed Systems*, New York, John Wiley & Sons Inc.

Bae, C., Yeh, W.C., Shukran, W,M., Chung, Y. Y. and Hsieh, T. J. (2012). A novel anomaly-network intrusion detection system using ABC algorithms, *Int. Journal of Innovative Computing, Information and Control*. **8(12)**: 8231-8248.

Bahrololum, M., Salahi, E. and Khaleghi, M. (2009). An Improved Intrusion detection technique based on two strategies using Decision Tree and Neural Network, *Journal of Convergence Information Technology*. **4(4)**: 96-101.

Banfield, R., Hall, L., Bowyer, K. and Kegelmeyer, W. (2005). Ensemble diversity measures and their application to thinning, *Information Fusion*. **6(1)**: 49–62.

Baram, Y. (1998). Partial classification: the benefit of deferred decision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **20(8)**: 769–776.

Bartlett, P. and Wegkamp, M. (2008). Classification with a reject option using a hinge loss, *Journal of Machine Learning Research*. **9**: 1823–1840.

Baruque, B., Porras, S. and Corchado, E. (2011). Hybrid classification ensemble using topology-preserving clustering, *New Generation Computing*. **29**: 329–344.

Bay, S. (1999). Nearest neighbor classification from multiple feature subsets, *Intelligent Data Analysis*. **3(3)**: 191–209.

Beauquier, J. and Hu, Y. (2008). Intrusion detection based on distance combination. *Int. Journal of Computer Science*. **2(3)**: 178–186.

Benferhat, S. and Tabia, K. (2005). On the combination of Naïve Bayes and decision trees for intrusion detection, Computational Intelligence for Modelling, Control and Automation. 1: 211–216.

Bi, Y. (2012). The impact of diversity on the accuracy of evidential classifier ensembles, *Int. Journal of Approximate Reasoning*. **53(4)**: 584–607.

Biggio, B., Fumera, G. and Roli, F. (2007). Bayesian analysis of linear combiners, *In: Proc. of the 7th Int. Conf. on Multiple Classifier Systems, Springer-Verlag, Berlin, Heidelberg*: 292–301.

Bishop, M. (2004). *Introduction to computer security*, Addison-Wesley Professional.

Boser, B.E., Guyon, I.M. and Vapnik, V.N. (1992). A training algorithm for optimal margin classifiers, *In: Proc. 5th Annual Workshop on Computational Learning Theory, AMC press*: 144-152.

Bottou, L. (1998). Online Algorithms and Stochastic Approximations, *Online Learning and Neural Networks, Cambridge University Press.*

Bouzida, Y. and Electric, M. (2006). Neural networks vs. decision trees for intrusion detection, *In: Proc. of the IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation*: 1–8.

Breiman, L. (2001). Random Forests, *Machine Learning*: **45(1)**: 5-32.

Brown, G. and Kuncheva, L. (2010). "good" And "bad" diversity in majority vote ensembles, *In: Proc. of Multiple Classifier Systems*: 124–133.

Bryll, R., Osuna, R.G. and Quek, F. (2003). Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets, *Pattern Recognition.* **36(6)**: 1291–1302.

Bryson, K.M.O. (2007). Post-pruning in decision tree induction using multiple performance measures, *Computers and Operations Research.* **34 (11)**: 3331–3345.

Chang, C.C. and Lin, C. J. (2011). LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology.* **2(3)**. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm

Chang, F., Guo, C., Lin, X. and Lu, C. (2010). Tree decomposition for large-scale SVM problems, *Journal of Machine Learning Research.* **10**: 2935–2972.

Cheeseman, P., Self, M., Kelly, J., Stutz, J., Taylor, W. and Freeman, D. (1988). AutoClass: a Bayesian classification system, *In: Proc. of the 5th Int. Workshop of Machine Learning, Morgan Kaufman.*

Chen, Y.L., Hsu, C.L. and Chou, S.C. (2003). Constructing a multi-valued and multi-labeled decision tree, *Expert System with Applications*. **25(2)**: 199–209.

Chung, Y. Y. and Wahid, N. (2012). A hybrid intrusion detection system using simplified swarm optimization (SSO), *Applied Soft Computing*. **12(9)**: 3014-3022.

Clark, P. and Niblett, T. (1989). The CN2 induction algorithm, *Machine Learning*. **3(4)**: 261–283.

Cohen, W.W. (1995). Fast effective rule induction, *In: Proc. of the 12th Int. Conf. on Machine learning*: 115–123.

Cordella, L., Foggia, P., Sansone, C., Tortorella, F. and Vento, M. (2000). A cascaded multiple expert system for verification, In: Multiple Classifier Systems, *Lecture Notes in Computer Science, Springer, Berlin/Heidelberg*. **1857**: 330–339.

Cortes, C. and Vapnik, V. (1995). Support-vector network, *Machine Learning*. **20**: 273-297.

Cunningham, P. and Carney, J. (2000). Diversity versus quality in classification ensembles based on feature selection, *In: Proc. of the 11th European Conf. on Machine Learning, ECML '00, Springer-Verlag, London, UK*: 109–116.

Dai, Q. (2012). A competitive ensemble pruning approach based on cross-validation technique, *Knowledge-Based Systems*. **37**: 394-414.

Depren, O., Topallar, M., Anarim, E. and Ciliz, M. K. (2005). An intelligent intrusion detection system for anomaly and misuse detection in computer networks, *Expert Systems with Applications*. **29(4)**: 713–722.

Didaci, L., Giacinto, G., Roli, F. and Marcialis, G. (2005). A study on the performances of dynamic classifier selection based on local accuracy estimation, *Pattern Recognition*. **38(11)**: 2188–2191.

Dong, L., Frank, E. and Kramer, S. (2005). Ensembles of balanced nested dichotomies for multiclass problems, *In: Proc. of the 9th European Conf. on Principles and practice of knowledge discovery in databases*: 84-95.

Du, W. and Zhan, Z. (2002). Building decision tree classifier on private data, *In: Proc. of the IEEE Int. Conf. on Privacy, Security and Data Mining, CRPIT '14, Australian Computer Society, Inc., Darlinghurst, Australia.* **14**: 1–8.

Faroun, K. M. and Boukelif, A. (2007). Neural network learning improvement using k-means clustering algorithm to detect network intrusions, *Int. Journal of Computational Intelligence.* **3(2)**: 161-168.

Feng, W., Zhang, Q., Hu, G. and Huang, J. X. (2014). Mining network data for intrusion detection through combining SVMs with ant colony networks, *Future Generation Computer Systems.* **37**: 127-140.

Fleiss, J. and Cuzick, J. (1979). The reliability of dichotomous judgments: unequal numbers of judgments per subject, *Applied Psychological Measurement.* **4(3)**: 537–542.

Freund, Y. and Schapire, R.E. (1996). Experiments with a New Boosting Algorithm, *In: Proc. of the 13th Int. Conf. on Machine Learning:* 148-156.

Freund, Y. and Schapire, R.E. (1997). A decision-theoretic generalization of online learning and an application to boosting, *Journal of Computer and System Sciences.* **55(1)**: 119–139.

Freund, Y. and Schapire, R.E. (1999). Large margin classification using the perceptron algorithm, *Machine Learning.* **37(3)**: 277-296.

Fumera, G., Pillai, I. and Roli, F. (2004). A two-stage classifier with reject option for text categorisation, *In: Proc. 5th Int. Workshop on Statistical Techniques in Pattern Recognition, Springer, Lisbon, Portugal.* **3138**: 771–779.

Gabrys, B. and Ruta, D. (2006). Genetic algorithms in classifier fusion, *Applied Soft Computing*. **6(4)**: 337–347.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H. and Herrera, F. (2011). An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes, *Pattern Recognition*. **44(8)**: 1761–1776.

Ganchev, T., Zervas, P., Fakotakis, N. and Kokkinakis, G. (2006). Benchmarking feature selection techniques on the speaker verification task, *In: Proc. of the 5th Int. symposium on Communication system, network and digital signal processing*: 314-318.

Giacinto, G. and Roli, F. (2001a). Design of effective neural network ensembles for image classification purposes, *Image Vision Computing*. **19(9-10)**: 699–707.

Giacinto, G. and Roli, F. (2001b). Dynamic classifier selection based on multiple classifier behavior, *Pattern Recognition*. **34(9)**: 1879–1881.

Giacinto, G., Perdisci, R., Rio, M.D. and Roli, F. (2008). Intrusion detection in computer networks by a modular ensemble of one-class classifiers, *Information Fusion*. **9**: 69–82.

Giray, S.M. and Polat, A.G. (2013). Evaluation and comparison of classification techniques for Network Intrusion Detection, *In: Proc. of the 13th IEEE Int. Conf. on Data Mining Workshop, Dallas*: 335-342.

Goebel, K. and Yan, W. (2004). Choosing classifiers for decision fusion, *In: Proc. of the 7th Int. Conf. on Information Fusion*: 563–568.

Gogoi, P., Bhattacharyya, D.K., Borah, B. and Kalita, J.K. (2011). A survey of outlier detection methods in network anomaly identification, *The Computer Journal*. **54(4)**: 570–588.

Gorbani, A. A., Lu, W. and Tavallaee, M. (2010). *Network Intrusion Detection and Prevention*, Springer New York Dordrecht Heidelberg London, ISBN 978-0-387-88770-8.

Guan, Y., Ghorbani, A. A. and Belacel, N. (2003). Y-means: a clustering method for intrusion detection*, In: Proc. of the IEEE Canadian Conf. on Electrical and Computer Engineering, Montreal, Quebec, Canada.*

Guo, C., Zhou, Y.J., Ping, Y., Luo, S.S., Lai, Y.P. and Zhang, Z.K. (2013). Efficient intrusion detection using representative instances, *Computers & Security.* **39**: 225-267.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. (2009). The WEKA data mining software: An update, *ACM SIGKDD Explorations Newsletter*: **11(1)**: 10–18.

Hassan, A., Haidar, S., Malek, S., Lyad, K. and Masri, Z.A. (2006), A Hybrid Honeypot Framework for Improving Intrusion Detection Systems in Protecting Organizational Networks, *Computers and Security.* **25(4)**: 274-288.

Ho, T. (1995). Random decision forests, *In: Proc. of the Third Int. Conf. on Document Analysis and Recognition, ICDAR '95, IEEE Computer Society, Washington, DC, USA*. **1**: 278-282.

Ho, T. (1998). The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence.* **20**: 832–844.

Howlett, R.J. and Lakhmi, C.J. (2001). *Radial Basis Function Networks 2: New Advances in Design*, Springer-Verlang, Berlin Heidelberg.

Jackowski, K. and Wozniak, M. (2009). Algorithm of designing compound recognition system on the basis of combining classifiers with simultaneous splitting

feature space into competence areas, *Pattern Analysis and Applications*. **12(4)**: 415–425.

Jackowski, K., Krawczyk, B. and Woniak, M. (2012). Cost-sensitive splitting and selection method for medical decision support system, In: Yin, H., Costa, J.A., Barreto, G. (Eds.), *Springer, Berlin Heidelberg*, *Intelligent Data Engineering and Automated Learning – IDEAL 2012, Lecture Notes in Computer Science*. **7435**: 850–857.

Jacobs, R. (1995). Methods for combining experts' probability assessments, *Neural Computation*. **7(5)**: 867–888.

Jacobs, R., Jordan, M., Nowlan, S. and Hinton, G. (1991). Adaptive mixtures of local experts, *Neural Computation*. **3**: 79–87.

Jawhar, M.M.T. and Mehrotra, M. (2010). Design Network Intrusion Detection System using hybrid Fuzzy-Neural Network, *Int. Journal of Computer Science and Security*. **4(3)**: 285-294.

John, G.H. and Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers, *In: Proc. of the 11th Conf. on uncertainty in artificial intelligence, Morgan Kaufmann, San Mateo*: 338-345.

Kayacik, H.G., Heywood, A.N.Z. and Heywood, M.I. (2005). Selecting features for Intrusion Detection: A Feature relevance analysis on KDD 99 Intrusion Detection Datasets, *In: Proc. of Int. 3rd annual Conf. on Privacy, Security and Trust.*

KDD Cup 1999: Computer Network Intrusion Detection, Available at http://www.sigkdd.org/kdd-cup-1999-computer-network-intrusion-detection

Khan, A., Khan, S. (2008). Two level anomaly detection classifier, *In: Proc. Int. Conf. on Computer and Electrical Engineering, Phuket*: 65-69.

Khardon, R. and Wachman, G. (2007). Noise tolerant variants of the perceptron algorithm, *The Journal of Machine Learning Research*. **8**: 227-248.

Kim, D.S., Nguyen, H.N. and Park, J.S. (2005). Genetic algorithm to improve SVM based network intrusion detection system, *In: Proc. of the 19th Int. Conf. on Advanced Information Networking and Applications*. **2**: 155–158.

Kim, G., Lee, S. and Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection, *Expert Systems with Applications*. **41(4)**: 1690–1700.

Kim, H. and Koehler, G.J. (1995). Theory and practice of decision tree induction. *Omega*, **23(6)**: 637–652.

Kittler, J. and Alkoot, F. (2003). Sum versus vote fusion in multiple classifier systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **25(1)**: 110–115.

Koc, L., Mazzuchi, T.A. and Sarkani, S. (2012). A network intrusion detection system based on a Hidden Naive Bayes multiclass classifier, *Expert Systems with Applications*. **39(18)**: 13492–13500.

Kohavi, R. (1995). The power of decision tables, *In: Proc. of the 8th European Conf. on Machine Learning*: 174-189.

Kohavi, R. (1996). Scaling up the accuracy of Naive Bayes Classifier: a Decision-Tree Hybrid, *In: Proc. of the 2nd Int. Conf. on knowledge discovery and data mining*: 202-207.

Krawczyk, B. and Wozniak, M. (2011). Privacy preserving models of k-NN algorithm, In: Burduk, R., Kurzynski, M., Wozniak, M., Zolnierek, A. (Eds.), Springer, Berlin/Heidelberg, Computer Recognition Systems 4, *Advances in Intelligent and Soft Computing*. **95**: 207–217.

Krogh, A. and Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning, *Advances in Neural Information Processing Systems*. **7**: 231–238.

Kuncheva, L. (2000). Clustering-and-selection model for classifier combination, *In: Proc. of the Fourth Int. Conf. on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*. **1**: 185–188.

Kuncheva, L., Whitaker, C., Shipp, C. and Duin, R. (2003). Limits on the majority vote accuracy in classifier fusion, *Pattern Analysis and Applications*. **6**: 22–31.

Landwehr, C. E., Bull, A. R., McDermott, J. P. and Choi, W. S. (1994). A taxonomy of computer program security flaws, *ACM Computer Survey*. **26(3)**: 211–254.

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A. and Srivastava, J. (2003). A comparative study of anomaly detection schemes in network intrusion detection, *In: Proc. 3rd SAIM Int. Conf. on Data Mining*: 25-36.

Lee, J. H., Sohn, S. G., Chang, B. H. and Chung, T. M. (2009). PKG-VUL: Security vulnerability evaluation and patch framework for package-based systems, *Electronics and Telecommunications Research Institute (ETRI) Journal*. **31(5)**: 554–564.

Lee, Z.J., Su, S.F. and Lee, C.Y. (2003). Efficiently Solving General Weapon-Target Assignment Problem by Genetic Algorithms with Greedy Eugenics, *IEEE Transactions on Systems, Man, and Cybernetics, Part B.* **33(1)**: 113-121.

Liao, H.J., Lin, C.H.R., Lin, Y.C. and Tung, K.Y. (2013). Intrusion detection system: A comprehensive review, *Journal of Network and Computer Applications*. **36**: 16–24.

Lin, L., Wang, X. and Liu, B. (2002). Combining multiple classifiers based on statistical method for handwritten Chinese character recognition, *In: Proc. of Int. Conf. on Machine Learning and Cybernetics*. **1**: 252–255.

Lin, S.W., Lee, Z.J., Chen, S.C. and Tseng, T.Y. (2008). Parameter determination of support vector machines and feature selection using simulated annealing approach, *Applied Soft Computing*. **8(4)**: 1505–1512.

Lin, S.W., Lee, Z.J., Chen, S.C. and Tseng, T.Y. (2008b). Parameter Determination of Support Vector Machines and Feature Selection using Simulated Annealing Approach, *Applied Soft Computing*. **8(4)**: 1505-1512.

Lin, S.W., Lee, Z.J., Ying, K.C. and Lee, C.Y. (2009). Applying Hybrid Meta-Heuristics for Capacitated Vehicle Routing Problem, *Expert Systems with Applications*. **36(2)**: 1505-1512.

Lin, S.W., Tseng, T.Y., Chou, S.Y. and Chen, S.C. (2008a). A Simulated-Annealing Based Approach for Simultaneous Parameter Optimization and Feature Selection of Back-Propagation Networks, *Expert Systems with Applications*. **34(2)**: 1491-1499.

Lin, S.W., Ying, K.C., Chen, S.C. and Lee, Z.J. (2008c). Particle Swarm Optimization for Parameter Determination and Feature Selection of Support Vector Machines, *Expert Systems with Applications*. **35(4)**: 1817-1824.

Lin, S.W., Ying, K.C., Lee, C.Y. and Lee, Z.J. (2012). An Intelligent Algorithm with Feature Selection and Decision Rules Applied to Anomaly Intrusion Detection, *Applied Soft Computing*. **12(10)**: 3285-3290.

Lindell, Y. and Pinkas, B. (2008). Secure multiparty computation for privacy-preserving data mining, *IACR Cryptology ePrint Archive*. **197**.

Lippmann, R.P., Fried, D.J., Graf, I., Haines, J.W., Kendall, K.R., McClung, D., Weber, D., Webster, S.E., Wyschogrod, D., Cunningham, R.K. and Zissman, M.A. (2000). Evaluating intrusion detection systems: The 1998 darpa off-line

intrusion detection evaluation, *DARPA Information Survivability Conference and Exposition.* **2**: 12-26.

Liu, H. and Setiono, S. (1995). Chi2: Feature Selection and Discretization of Numeric Attributes, *In: Proc. of the 7th IEEE Int. Conf. on Tools with artificial intelligence (TAI '95)*: 88-90.

Louvieris, P., Clewley, N. and Leiu, X. (2013). Effect-based feature identification for network intrusion detection, *Neurocomputing.* **121**: 265-273.

Luo, B. and Xia, J. (2014). A novel intrusion detection system based on feature generation with visualization strategy, *Expert System with Applications.* **41**: 4139-4147.

Magalhaes, R. M. (2003). *Host-Based IDS vs Network-Based IDS (Part 1)*, Windows Security News letter.

Maimon, O. and Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook*, Second Edition, Springer New York Dordrecht Heidelberg London.

Manevitz, L.M. and Yousef, M. (2001). One-class svms for document classification, *Journal of Machine Learning Research.* **2**: 139-154.

Manikopoulos, C. and Papavassiliou, S. (2002). Network intrusion and fault detection: A statistical anomaly approach, *IEEE Communications Magazine.* **40(10)**: 76–82.

Marono, N.S., Betanzos, A.A. and Estevez, R.M.C. (2009). A wrapper method for feature selection in multiple classes datasets, Bio-Inspired Systems: Computational and Ambient Intelligence. *Lecture Notes in Computer Science.* **5517**: 456–463.

McHugh, J. (2000). Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln laboratory, *ACM Transactions on Information and System Security.* **3(4)**: 262–294.

Mohammed, M.N. and Sulaiman, N. (2012). Intrusion Detection System Based on SVM for WLAN. *Procedia Technology.* **1**: 313–317.

Mukherjee, S. and Sharma, N. (2012). Intrusion detection using Naive Bayes classifier with feature reduction, *Procedia Technology.* **4**: 119–128.

Mukkamala, S., Janoski, G. and Sung, A. (2002). Intrusion detection using neural networks and support vector machines, *In: Proc. of the IEEE Int. joint Conf. on neural networks*: 1702–1707.

Mukkamala, S., Sung, A.H. and Abraham, A. (2003). Intrusion Detection Using Ensemble of Soft Computing Paradigms, *In: Proc. the 3rd Int. Conf. on Intelligent Systems Design and Applications, Intelligent Systems Design and Applications, Advances in Soft Computing.* **23**: 239-248.

Mukkamala, S., Sung, A.H., Abraham, A. and Ramos, V. (2004). Intrusion detection systems using adaptive regression splines, *In: Proc. of 6th Int. Conf. on Enterprise information systems, ICEIS'04*: 26–33.

Nanni, L. (2006). Letters: Experimental comparison of one-class classifiers for online signature verification, *Neurocomputing.* **69(7–9)**: 869–873.

Nettleton, D.F., Puig, A.O. and Fornells, A. (2010). A study of the effect of different types of noise on the precision of supervised learning techniques, *Artificial Intelligence Review.* **33(4)**: 275-306.

Neumann, P. G. and Porras, P. A. (1999). Experience with emerald to date, *In: Proc. 1st USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara*: 73-80.

Novikov, D., Yampolskiy, R.V. and Reznik, L. (2006). Artificial Intelligence Approaches for Intrusion Detection, *In: Proc. IEEE Int. Conf. on Systems, Applications and Technology, Long Island, New York*.

Noz, G.M.M., Lobato, D.H. and Suarez, A. (2009). An analysis of ensemble pruning techniques based on ordered aggregation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **31(2)**: 245–259.

Olusola, A.A., Oladele, A.S. and Abosede, D.O. (2010). Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features, *In: Proc. of the World Congress on Engineering and Computer Science, San Francisco, USA*. **1**: 162-168.

Opitz, D.W. and Shavlik, J.W. (1996). Generating accurate and diverse members of a neural network ensemble, In: Proc. Advances in Neural Information Processing Systems: 535–541.

Pai, P.F. and Hong, W.C. (2005). Support Vector Machines with Simulated Annealing Algorithms in Electricity Load Forecasting, *Energy Conversion and Management.* **46(17)**: 2669-2688.

Pajares, G., Guijarro, M., Herrera, P.J. and Ribeiro, A. (2004). On Combining Support Vector Machines and Simulated Annealing in Stereovision Matching, *IEEE Transactions on Systems, Man, and Cybernetics, Part B, Cybernetics.* **34(4)**: 1646-1657.

Pan, Z.S., Chen, S.C., Hu, G.B. and Zhang, D.Q. (2003). Hybrid Neural Network and C4.5 for Misuse Detection", *In: Proc. of the Second Int. Conf. on Machine Learning and Cybernetics*: 2463-2467.

Panda, M. and Patra, M. R. (2007). Network intrusion detection using nave Bayes, Int. Journal of Computer Science and Network Security. 7(12): 258-263.

Panda, M., Abraham, A., Patra, M.R. (2012). A hybrid intelligent approach for network intrusion detection, *In: Proc. of Int. Conf. on Communication technology and system design*: 1-9.

Partalas, I., Tsoumakas, G. and Vlahavas, I. (2009). Pruning an ensemble of classifiers via reinforcement learning, *Neurocomputing*. **72(7–9)**: 1900–1909.

Partridge, D. and Krzanowski, W. (1997). Software diversity: practical statistics for its measurement and exploitation, *Information and Software Technology*. **39(10)**: 707–717.

Pavlo, A., Paulson, E., Rasin, A., Abadi, D., DeWitt, D., Madden, S. and Stonebraker, M. (2009). A comparison of approaches to large-scale data analysis, *In: Proc. of the 2009 ACM SIGMOD Int. Conf. on Management of Data, SIGMOD '09, ACM, New York, NY, USA*: 165–178.

Pearl, J. (1985). Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning (UCLA Technical Report CSD-850017), *In: Proc. of the 7th Conf. of the Cognitive science society, University of California, Irvine, CA*: 329–334.

Peddabachigaria, P., Abrahamb, A., Grosanc, C. and Thomas, J. (2007). Modeling Intrusion Detection System Using Hybrid Intelligent Systems, *Journal of Network and Computer Applications*. **30(1)**: 114-132.

Peng, J., Feng, C., Rozenblit, J. (2006). A hybrid intrusion detection and visualization system, *In Proc. of the 13th Annual IEEE Int. Symposium and Workshop on Engineering of Computer Based Systems (ECBS'06)*: 505–506.

Peng, Y., Huang, Q., Jiang, P. and Jiang, J. (2005). Cost-sensitive ensemble of support vector machines for effective detection of microcalcification in breast cancer diagnosis, In: Wang, L., Jin, Y. (Eds.), Springer, Berlin/ Heidelberg,

Fuzzy Systems and Knowledge Discovery, Lecture Notes in Computer Science. **3614**: 483–493.

Perdisci, R., Gu, G. and Lee, W. (2006). Using an ensemble of one-class SVM classifiers to harden payload-based anomaly detection systems, *In: Proc. of the 6th Int. Conf. on data mining*: 488–498.

Platt, J.C. (1998). Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, *In: Proc. of Int. Conf. on Advance in kernel methods - Support vector learning.*

Premaratne, U., Ling, C., Samarabandu, J. and Sidhu, T. (2009). Possibilistic decision trees for intrusion detection in IEC61850 automated substations, *In: Proc. of the Int. Conf. on Industrial and Information Systems*: 204–209.

Qi, Z.J., Feng, F., Xin, Y.K. and Heng, L.Y. (2013). Dynamic entropy based DoS attack detection method, *Computers and Electrical Engineering.* **39**: 2243–2251.

Quinlan, J.R. (1979). *Discovering rules by induction from large collections of examples*, In D. Michie (Ed.), Expert Systems in the Micro Electronic Age, Edinburgh University Press, Edinburgh.

Quinlan, J.R. (1986). Introduction of decision trees*, Machine Learning.* **1**: 81–106.

Quinlan, J.R. (1987). Decision trees as probabilistic classifiers, *In: Proc. of the 4th Int. Workshop Machine Learning*: 31–37.

Quinlan, J.R. (1993). *C 4.5: programs for machine learning*, San Mateo: Morgan Kaufmann Publishers.

Quinlan, J.R. (1996). Learning decision tree classifier, *ACM Computing Surveys (CSUR).* **28(1)**: 71–72.

Rao, N. (2004). A generic sensor fusion problem: classification and function estimation, In: Roli, F., Kittler, J., Windeatt, T. (Eds.), Springer, *Multiple Classifier Systems, Lecture Notes in Computer Science*. **3077**: 16–30.

Raza, S., Wallgren, L. and Voigt, T. (2013). SVELTE: Real-time intrusion detection in the Internet of Things, *Ad Hoc Networks*. **11**: 2661–2674.

Rivest, R. (1987). Learning decision lists, *Machine Learning*. **2(3)**: 229–246.

Rokach, L. and Maimon, O. (2005). Feature set decomposition for decision trees, *Intelligent Data Analysis*. **9(2)**: 131–158.

Ruta, D. and Gabrys, B. (2005). Classifier selection for majority voting, *Information Fusion*. **6(1)**: 63–81.

Sabhnani, M. and Serpen, G. (2003). Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context, *In: Proc. Int. Conf. on Machine Learning: Models, Technologies, and Applications, Las Vegas, Nevada, USA*: 209–215.

Sachs, A., Thiel, C. and Schwenker, F. (2006). One-class support-vector machines for the classification of bioacoustic time series, *In: Proc. of the ICGST Int. Journal on Artificial Intelligence and Machine Learning*. **6(4)**: 29-34.

Sammany, M., Sharawi, M., Beltagy, E. M. and Saroit, I. (2007). Artificial Neural Networks Architecture for Intrusion Detection Systems and Classification of Attacks, *In: Proc. 5th Int. Conf. INFO, Cairo University*.

Schneier, B. (2000). *Secrets and Lies: Digital Security in a Networked World*, John Wiley & Sons, Inc., Indianapolis, Indiana.

Scholkopf, B., Platt, J.C., Taylor, J.S., Smola, A.J. and Williamson, R.C. (2001). Estimating the support of a high-dimensional distribution, *Neural Computation*. **13(7)**: 1443–1471.

Shihab, K. (2006). A Backpropagation neural network for computer network security, *Journal of Computer Science*. **2(9)**: 710-715.

Shin, H.J., Eom, D.H. and Kim, S.S. (2005). One-class support vector machines: An application in machine fault detection and classification, *Computers and Industrial Engineering*. **48(2)**: 395–408.

Shipp, C. and Kuncheva, L. (2002). Relationships between combination methods and measures of diversity in combining classifiers, *Information Fusion*. **3(2)**: 135–148.

Shlien, S. (1990). Multiple binary decision tree classifiers, *Pattern Recognition*. **23(7)**: 757–763.

Shon, T. and Moon, J. (2007). A Hybrid Machine Learning Approach to Network Anomaly Detection, *Information Sciences*. **177**: 3799-3821.

Shwarz, S.S., Singer, Y. and Srebro, N. (2007). Pegasos: Piramal estimated subgradient solver for SVM*, In: Proc. of the 24th Int. Conf. on machine learning*: 807-814.

Silva, L.S., Santos, A.C., Silvas, J.D.S. and Montes, A. (2004). A Neural Network Application for Attack Detection in Computer Networks, *In: Proc. Int. Joint Conf. on Neural Networks, Budapest*. **2**: 1569-1574.

Sindhu, S.S.S., Geetha, S. and Kannan, A. (2012). Decision tree based light weight intrusion detection using a wrapper approach, *Expert System with Applications*. **39(1)**: 129–141.

Skurichina, M. and Duin, R. (2002). Bagging, boosting and the random subspace method for linear classifiers, *Pattern Analysis and Applications*. **5(2)**: 121–135.

Smetek, M. and Trawinski, B. (2011). Selection of heterogeneous fuzzy model ensembles using self-adaptive genetic algorithms, *New Generation Computing*. **29**: 309–327.

Song, J., Takakura, H., Okabe, Y. and Kwon, Y. (2009). Unsupervised anomaly detection based on clustering and multiple one-class SVM. *IEICE Transactions on Communications.* **E92-B (6)**: 1982–1990.

Souza, D.J.T., Matwin, S. and Japkowicz, N. (2006). Parallelizing feature selection, *Algorithmica*. **45(3)**: 433–456.

Stolfo, S.J., Fan, W., Lee, W., Prodromidis, A. and Chan, P.K. (2000). Cost based modeling for fraud and intrusion detection: Results from the jam project, *In: Proc. of DARPA Information survivability Conf. and exposition, DISCEX*. **2**: 130-144.

Sung, A.H. and Mukkamala, S. (2003). Feature selection for Intrusion Detection using Neural Networks and Support Vector Machines, *Journal of the Transportation Research Board of National Academies.* **1822**: 33-39.

Tao, D., Tang, X., Li, X. and Wu, X. (2006). Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval, *IEEE Transactions on Pattern Analysis Machine Intelligence*. **28(7)**: 1088–1099.

Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A.A. (2009). A detailed analysis of the KDD Cup datasets, *In: Proc. 2nd IEEE Symposium on computational intelligence in security and defense applications*: 53-58.

Tax, D. and Duin, R. (2002). Using two-class classifiers for multiclass classification, *In: Proc. of the 16th Int. Conf. on Pattern Recognition*. **2**: 124 –127.

Tax, D. and Duin, R.P.W. (2001). Combining one-class classifiers, *In: Proc. of the Second Int. Workshop on Multiple Classifier Systems, Springer-Verlag, London, UK*: 299–308.

Termenon, M. and Grana, M. (2012). A two stage sequential ensemble applied to the classification of alzheimer's disease based on MRI features, *Neural Processing Letters*. **35(1)**: 1–12.

Thomas, C., Sharma, V. and Balakrishnan, N. (2008). Usefulness of DARPA dataset for intrusion detection system evaluation, *SPIE Defense and Security Symposium, Int. Society for Optics and Photonics*.

Ting, K., Wells, J., Tan, S., Teng, S. and Webb, G. (2011). Feature-subspace aggregating: ensembles for stable and unstable learners, *Machine Learning*. **82**: 375–397.

Tremblay, G., Sabourin, R. and Maupin, P. (2004). Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm, *In: Proc. of the Pattern Recognition, 17th Int. Conf. on (ICPR'04), IEEE Computer Society, Washington, DC, USA*. **1**: 208-211.

Ueda, N. and Nakano, R. (1996). Generalization error of ensemble estimators, *In: Proc. of IEEE Int. Conf. on Neural Networks, Washington, USA*: 90–95.

Unnthorsson, R. Data sets for model selection in one-class v-svms using rbf kernels. http://www.hi.is/~runson/svm/.

Unnthorsson, R., Runarsson, T.P. and Jonsson, M.T. (2003). Model selection in one-class v-svms using RBF kernels, *In: Proc. of the 16th Int. Congress on Condition Monitoring and Diagnostic Management*.

VanErp, M., Vuurpijl, L., Schomaker, L. (2002). An overview and comparison of voting methods for pattern recognition, *In: Proc. of the Eighth Int. Workshop on Frontiers in Handwriting Recognition*: 195–200.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Walkowiak, K., Sztajer, S. and Wozniak, M. (2011). Decentralized distributed computing system for privacy-preserving combined classifiers – modeling and optimization, In: Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B. (Eds.), Springer, Berlin/Heidelberg, *Computational Science and Its Applications – ICCSA 2011, Lecture Notes in Computer Science*. **6782**: 512–525.

Wang, G., Hao, J., Ma, J. and Huang, L. (2010). A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering, *Expert Systems with Applications*. **37**: 6225–6232.

Wang, H., Fan, W., Yu, P. and Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers, *In: Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, ACM, New York, NY, USA:* 226–235.

Wettschereck, D. and Dietterich, T. (1992). Improving the performance of radial basis function networks by learning center locations. *Advances in Neural Information Processing Systems 4, Denver, CO: Morgan Kaufmann*. **4**: 1133–1140.

Wilk, T. and Wozniak, M. (2012). Soft computing methods applied to combination of oneclass classifiers, *Neurocomputing*. **75**: 185–193.

Woloszynski, T., Kurzynski, M., Podsiadlo, P. and Stachowiak, G. (2012). A measure of competence based on random classification for dynamic ensemble selection, *Information Fusion*. **13(3)**: 207–213.

Woods, K., Kegelmeyer, W.P. and Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **19(4)**: 405–410.

Wozniak, M. and Krawczyk, B. (2012). Combined classifier based on feature space partitioning, *Int. Journal of Applied Mathematics and Computer Sciences*. **22(4)**: 855–866.

Wozniak, M. and Zmyslony, M. (2010). Combining classifiers using trained fuser – analytical and experimental results, *Neural Network World*. **13(7)**: 925–934.

Wu, S.X. and Banzhaf, W. (2010). The use of computational intelligence in intrusion detection system: a review, *Applied Soft Computing*. **10(1)**: 1-35.

Wu, T., Lin, C. and Weng, R. (2004). Probability estimates for multi-class classification by pairwise coupling, *Journal of Machine Learning Research*. **5**: 975–1005.

Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, A., Motoda, Y., McLachlan, G.J., Ng, A., Liu, B. and Yu, P.S. (2008). Top 10 algorithms in data mining, *Knowledge and Information System*, **14(1)**: 1–37.

Yang, J. and Olafsson, S. (2006). Optimization-based feature selection with adaptive instance sampling, *Computer & Operation Research.* **33(11)**: 3088–3106.

Yousef, A., Goran, K., Slavko, G. and Zoran, J. (2014). Flow-based anomaly intrusion detection system using two neural network stages, *Computer Science and Information Systems*. **11(2)**: 601–622.

Zaman, S. and Karray, F. (2009). Features selection for intrusion detection systems based on support vector machines, *In: Proc. 6th IEEE Conf. on Consumer Communications and Networking Conf., Las Vegas*: 1-8.

Zanero, S. (2007). Flaws and frauds in the evaluation of IDS/IPS technologies, *In: Proc. of Forum of Incident Response and Security Teams (FIRST-2007)*: 1-18.

Zhang, J. and Zulkernine, M. (2006a). A hybrid network intrusion detection technique using random forests, *In: Proc. of the 1st Int. Conf. on availability, reliability and security*: 262–269.

Zhang, X., Gu, C. and Lin, J. (2006b). Support vector machines for anomaly detection, *In: Proc. of the 6th World Congress on Intelligent Control and Automation*: 2594–2598.

Zhang, Y. and Jin, X. (2006). An automatic construction and organization strategy for ensemble learning on data streams, *ACM SIGMOD Record Newsletter*. **35(3)**: 28–33.

Zheng, Z. and Padmanabhan, B. (2007). Constructing ensembles from data envelopment analysis, *INFORMS Journal on Computing*. **19(4)**: 486–496.

Zhou, W., Jia, W., Wen, S., Xiang, Y. and Zhou, W. (2013). Detection and defense of application-layer DDoS attacks in backbone web traffic, *Future Generation Computer Systems*.

Zhou, Z.H., Wu, J. and Tang, W. (2002). Ensembling neural networks: many could be better than all, *Artificial Intelligence*. **137(1-2)**: 239–263.