Intrusion Detection System Using Machine Learning Models

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Computer Science and Engineering

Submitted By

Sumit Gangwal

(Roll No. 801332027)

Under the supervision of:

Dr. V. P. Singh

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Intrusion Detection System Using Machine Learning Models*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. V.P.Singh* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Signature:

(Sumit Gangwal)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. V.P. Singh)
Assistant
Professor
Computer Science and Engineering
Department

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic
Affairs)
Thapar University
Patiala

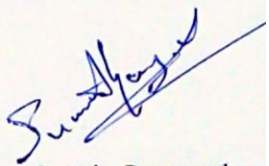
ACKNOWLEDGEMENT

No Volume of words are enough to express my gratitude towards my guide, **Dr. V. P. Singh,** Assistant Professor, Computer Science and Engineering Department, Thapar University, Patiala, who has been very concerned and has aided for all the guidance essential for the thesis report. He has helped me to explore this vast topic in an organized manner and provided me all the ideas on how to work towards a research-oriented venture.

I am also thankful to Dr. Ashutosh Mishra, P.G. Coordinator, for the motivation and inspiration that triggered me for the seminar work.

I would like also like to thank the staff members and my colleagues who were always there in the need of the hour and provided with all the help and facilities, which I required, for the completion of my thesis.

Most importantly, I would like to thank my parents and the almighty God for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at the times when there was no hope.



Sumit Gangwal

801332027

M.E. (CSE)

ABSTRACT

The anomaly Intrusion Detection is one of the major research issues now a days. The advancement in networks has indeed increased the need of designing and executing a more reliable and more accurate network security systems. For this purpose, intrusion detection systems (IDS) are used to monitor the threats encountered on the network, by detecting any change in the normal profile. The idea here is, to use classification algorithms for analyzing KDD'99 datasets, with 41 Attributes (features). Based on these 41 attributes, the KDD'99 Datasets has been classified into five different types of attacks, i.e. normal, Probe, U2R, R2L and DOS. The algorithms used in this paper are support vector machine (SVM) and Random Forest (RF). Apart from using Random Forest for classification, it is also used in feature extraction. These algorithms are used to classify the data among various classes. The simulation results demonstrated that the support vector machine out performs as compared with Random Forest as an anomaly intrusion detection system with high accuracy. The validation of snort rule's dataset, generated by the given attacks, has been performed using support vector machine. The experimentation results have higher accuracy, for the validation of the KDD'99 data set used in the training.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. Background

With the rapid advancement in networks and networking technologies all over the globe for the purpose of information sharing and transferring data from source to destination has also increased the risk of network attacks exponentially. According to Internet World Stats [18] shown in table 1.1, by 31st December 2000, the total number of Internet users in the world was lesser than half a billion (360985492) users and it has increased by 753.03% by 31st December 2014 to more than 3 billion (3079339857) users. Since the number of Internet users is growing so rapidly, the number of intrusions is also increasing concurrently, and the types of attacks are becoming more and more sophisticated. Therefore, this is clearly understandable that there is a great need of paying more attention towards creating better network security systems.

Table 1.1. World internet usage and population statistics

World Regions	Population	Internet Users Dec. 31, 2000	Internet Users Latest Data	Growth 2000-2015
Africa	1,158,353,014	4,514,400	318,633,889	6,958.2 %
Asia	4,032,654,624	114,304,000	1,405,121,036	1,129.3 %
Europe	827,566,464	105,096,093	582,441,059	454.2 %
Middle East	236,137,235	3,284,800	113,609,510	3,358.6 %
North America	357,172,209	108,096,800	310,322,257	187.1 %
Latin America / Caribbean	615,583,127	18,068,919	322,422,164	1,684.4 %
Oceania / Australia	37,157,120	7,620,480	26,789,942	251.6 %
WORLD TOTAL	**7,264,623,793**	**360,985,492**	**3,079,339,857**	**753.0 %**

Any actions that seeks to compromise the availability, confidentiality and integrity of a system is referred as attack [1]. The attacks are focused over the vulnerabilities of the user on the network by defying user access rights and gaining unauthorized access. The Internet has become an important part of our life nowadays. It is used in almost

every area, like social networking, business, entertainment, education, *etc*. For example, businesses are run on the Internet where people can buy and sell stuff and can use different facilities like Internet banking. Confidential information, on the other hand, like for instance: national security, are passed within people and groups via the Internet, and even a slight change of information or a sneak of this information by hacking into the system can cause devastating effects for mankind. Hence, the Network Security has become a serious issue for all the network users around the world. To prevent our systems and data from theft and attacks beforehand, the Intrusion Detection System (IDS) was developed. Intrusion detection system is based on the assumption that the behavior of intruder is different from that of a legal/normal user, intrusion detection system is a set of techniques that are being used for detection of suspicious activities on both, network as well as on host level

In the last few decades, we have entered into a new era of networking, creating and destroying different things on the network either legally or illegally. An intrusion detection system is capable of detecting different types of malicious activities on the network that cannot be detected and prevented by conventional firewalls with their security policies. The implementation of an effective intrusion detection system is not an easy task, because the intrusion techniques applied by the intruders are always changing and becoming more intelligent and advanced. Therefore, an intrusion detection system systems needs to be flexible and self-adjustable according to the changing intrusion methods. Currently, the only way of achieving a flexible and robust IDS is by using soft computing techniques. Figure 1.1 describes an internet model having IDS installed at various positions.

The soft computing approach is a new method adopted for intrusion detection. It can be stated as a method to detect and discover patterns, associations, changes, anomalies, rules and events in the data (*Hand et al.*, 2001). It tries to find and extract the information form the data, in the form of models, which cannot be detected by naked eyes easily. There are various soft computing techniques like- Artificial Neural Networks (ANN) (Cho and Park, 2003; Lippmann and Cunningham, 2000), Support

Vector Machines (SVM) (Abraham, 2001), Neuro-Fuzzy (NF) computing (*Mukkamala et al.*, 2003) statistics (*Anderson et al.*, 1995; *Debar* et al., 1992) *etc.*

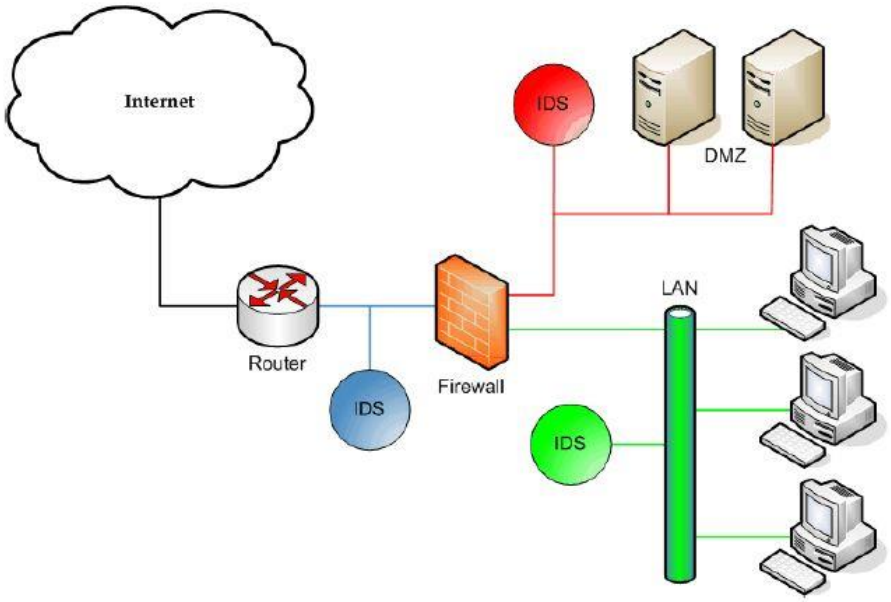


Figure 1.1.: A computer network with IDS systems at various locations.

On the other hand, some learning techniques are combined with other learning techniques, called as hybrid techniques. These soft computing techniques are basically classifiers, which classifies the incoming network data whether it is normal data or an attack. Among all these techniques, support vector machine has been proved as an effective technique in terms of accuracy, less training time, ability to overcome high dimensionality of data. Although, support vector machine has better performance than many other traditional Artificial Intelligence techniques, but to improve the efficiency of intrusion detection system and reduce the rate of occurrence of false positive and false negative, can still be done by using hybrid approach for intrusion detection system. In this thesis, along with support vector machine, random forest technique is also used for training and testing of data. The random forest is also implemented to extract important features from the dataset. Usually feature selection method tries to determine the best subset which has adequate information for intrusion detection from the subsets of features. Apart from random forest, other artificial intelligence and

machine learning techniques used for feature extraction or feature reduction are; Decision Tree, Bayesian Networks, Neural Networks, *etc*.

The selection and reducing the features in the data increases both training and testing speed of the support vector machine and random forest. This method of selection and reduction of features also increases the accuracy and efficiency of the intrusion detection system. In the real-life intrusion detection dataset, there may be some features that are redundant or less important than other features [20]. These redundant or unnecessary features make it harder and time consuming to detect possible intrusion types [21]. After the feature extraction is performed, the dataset is then used to train the support vector machine and random forest. After train of both techniques the testing data is tested and it gives the prediction results about the test data whether the data is Intrusive or not. The accuracy and significance of output of support vector machine and random forest is calculated. The best technique with highest accuracy is determined from the predicted results. These prediction results are also checked and validated using snort, by writing rules according to the predicted result produced by both and generating different types of attack with almost similar values as in the data set. This will help the system to validate our predicted results.

1.2. Thesis Structure

The rest of the thesis is structured as under:

- Chapter 2 introduces the earlier work done in the field of intrusion detection using artificial intelligence, support vector machine and random forest techniques, as well as the concepts and theories regarding the intrusion detection system and brief description about Snort.
- Chapter 3 states and defines the problem statement that describes the need of anomaly intrusion detection system.
- Chapter 4 presents the experimental setup and implementation part used to carry out the research. It deals with all the technologies, tools and datasets used while working on the problem to find the solution.

- Chapter 5 reports the experimental results and a brief discussion and explanation of results.
- Chapter 6 concludes the whole research and provides us with some possible perspectives of the research in the near future.

CHAPTER 2

LITERATURE SURVEY

This chapter deals with the detailed review of the earlier work done related to the artificial intelligence technique in the area of intrusion detection and basic concepts about the soft computing techniques that are used in this research work ,that is, random forest and support vector machine. In the first section, a detailed introduction about the intrusion detection system, its definition and formation are provided. In the second section the brief history and overview of the dataset (KDD'99) is given. In the third section, the basic concepts and theory of support vector machine are discussed. The main focuses on theoretical advantages of support vector machine, which makes it such a good instrument for intrusion detection. In the third section the basic concepts of random forest technique has also been discussed.

2.1. Over view of IDS.

2.1.1. Basic concept of IDS.

The intrusion detection system is a tool for detection of abnormal behaviors in a system. An abnormal pattern in general is, unwanted, malicious and misuse activity occurring within the system. The intrusion detection system can be an implementation of software or a hardware that is used to monitor the networks for intrusions or any deviation from the normal activity. Normally, intrusion detection system is a security surveillance system, such as a firewall system that tries to find and if possible, safeguard and prevent the system from harm. The basic functioning of the intrusion detection system is, to behave as a passive alert system, that is, if the intrusion is found on the network IDS produces an alarm and gives the user the relevant information (IP, ports, packets, *etc*.) which initiated the alarm. The intrusion detection system, which plays in the active mode responds to the observed intrusion by utilizing counter-measures to prevent the system from attack, these types of active intrusion

detection system are called as Intrusion Prevention Systems (IPS). Intrusion detection system is a set of techniques that are being used for detection of suspicious activities on both, network as well as on host level. It is classified into two categories:

2.1.1.1. Misuse or Signature based Intrusion Detection System.

The main idea behind the misuse intrusion detection system is presenting the attack in the form of signatures and patterns, these patterns are then maintained as a database of known attack scenarios and signatures. These signatures are then compared with the data received on the network and if the match is found, it generates an alert to the user and the user can take required steps, like in the antivirus systems. Advantage of Misuse IDS is that it produces very low false positives (an alert which is not actually an intruder activity), are easy to develop, and requires less computational resources.

The problem with this kind of approach is:

- It can detect only those threats that are in its database, else it will bypass any new type of threat. For this, the database is needed to update frequently.
- Misuse based systems can detect previously known attacks only.
- Every new signature of a new attack or its variant needs to be added to the database, therefore the size becomes very large and hence updating it, is too much time consuming.
- It is difficult to determine an attack, as the description of the attack in database is generally a low level description.
- If the signatures are specific in description of attack, the lesser false positive alerts there are, but if the signatures are more specific, it will be easier for the intruder to prepare a slightly different variant of that attack which would not match the signature and hence would be overlooked.

2.1.1.2. Anomaly based Intrusion Detection System.

Anomaly intrusion detection system is established on the statistical analysis, it detects attacks based on irregularities in the pattern with respect to the normal patterns

of data on the network, that is, it is based on the statistical analysis of network data, the flag of all the normal states are set to the normal activity profile of a system and the rest of the states as anomalous activities. The anomaly intrusion detection system is based on data-mining techniques. So anomaly intrusion detection system is capable of detecting new unknown threats. But these machine learning techniques may also suffer from certain disadvantages like:

- The time required in training the system about normal behaviour may be lengthy.
- Behaviour of surveillance environment may differ after certain time period, resulting the retraining of the system.
- If training dataset itself consists of attacks, the intrusion detection system will take malicious behaviour as normal.

To remove these disadvantages and restrictions, an efficient and accurate soft computing technique that detects and classifies the data precisely and accurately and takes lesser training time are considered.

2.1.1.3. False Positives and False Negatives.

To evaluate the IDS's performance and accuracy for detecting intrusion, there are four possible conditions that are considered. In figure 2.1, the vertical axis represents how the network activity is detected by the intrusion detection system and the horizontal axis represents the actual activity on the network. False positives are legal occurrences of data on the network that are incorrectly detected as anomalous. True positives are the occurrences that are correctly marked anomalous. The anomalous occurrences that are not detected by the detector are called as false negatives and hence not marked as anomalous.

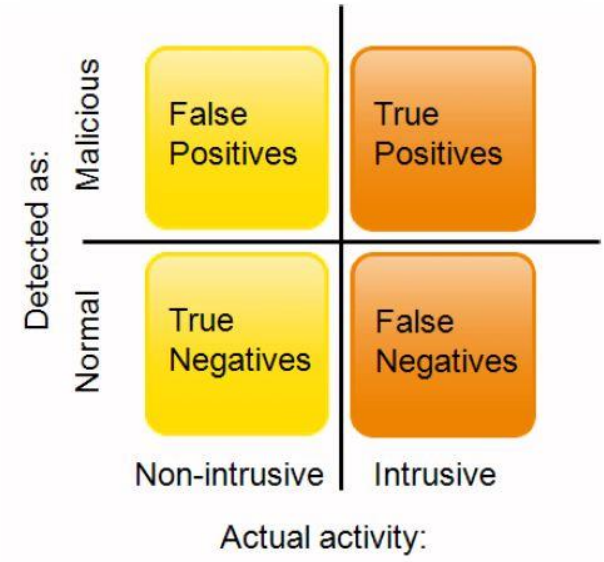


Figure 2.1.: False positives, true positives, true negatives and false negatives.

The true negatives the occurrences that are correctly marked as legal activities. It must be the network operator's responsibility to investigate and check, whether the anomaly or intrusion is false positive or false negative. The most risky situation are the false negatives. It is a possible intrusive activity that passes through the intrusion detection system as a normal activity, it might be harmful for the system.

This problem can be solved by using a soft computing method that can provide accurate results and can handle very large dimensions of data, that is, a method which classify the data very accurately and methodologies that can improve the performance of the classifier and hence the performance of the result. For this purpose KDD'99 dataset is used which is universally accepted as the benchmark of intrusion detection Dataset. The support vector machine and Random Forest are the classifiers whose accuracy and precision are way much more than any other classifier and they both are flexible towards a large amount of data, which produces lower false positive rate and lower false negative rate and it is desirable to know which classifier is capable of producing desired results. Along with SVM, random forest is also used to train and test the dataset. The training and testing outcome of both the classifiers is compared, to find out which technique is better suited for intrusion detection system. The random forest is also used in this research for finding the significance of features and remove

the unnecessary features and data from the dataset, which improves the classification results drastically utilizing lesser training and testing time.

2.2. Data Description

To perform the experiments, the benchmark of International Knowledge Discovery and Data Mining group (KDD) datasets has been utilized for the present research work. In 1998, with sponsorship of Defense Advanced Research Projects Agency (DARPA), the Lincoln Lab of Massachusetts Institute of Technology (MIT), collected the data on the network. That was the first initiative ever taken to encourage the involvement by eliminating privacy and security concerns by supplying the datatypes that are used in the network to the intrusion detection designers to create and generate intrusion detection methodologies, as shown in table 2.1.

These datasets were generated by initializing a simulation of a fictitious network consisting of different "target" machines running various operating systems and services. To spoof IP addresses and generate the traffic between IP addresses different machines were used. At last, to record all network traffic a sniffer was used that stored it in TCP dump format. Since, the data taken for the study is very huge, it takes a lot of time in pre-processing and then training and testing of data. Hence a smaller dataset of KDD'99 has been used for the experimental purpose.

Table 2.1. Features of Kdd'99 dataset [8].

SN.	Feature Name	Date Type	Description
1	duration	Continuous	Time period of connection
2	protocol_type	Nominal	Protocol used in connection
3	service	Nominal	Destination service
4	Flag	Nominal	Status flag of connection
5	src_bytes	Continuous	Bytes sent from the source to destination
6	dst_bytes	Continuous	Bytes sent from the destination to source
7	Land	Nominal	'1' if connection is from/to the same host/port; else '0'
8	wrong_fragment	Continuous	Number of wrong fragment
9	urgent	Continuous	Number of urgent packets
10	Hot	Continuous	Number of hot indicators
11	num_failed_logins	Continuous	Number of failed login attempts
12	logged_in	Nominal	'1' if successfully logged in; else '0'

SN.	Feature Name	Date Type	Description
13	num_compromised	Continuous	Number of compromised conditions
14	root_shell	Nominal	'1' if root shell is obtained; else '0'
15	su_attempted	Nominal	'1' if "su root" command attempted; else '0'
16	num_root	Continuous	Number of root accesses
17	num_file_creations	Continuous	Number of file creation operations
18	num_shell	Continuous	Number of shell prompts
19	num_accessfile	Continuous	Number of operations on access control files
20	num_outbound_cmds	Continuous	Number of outbound commands in an ftp session
21	is_host_login	Nominal	'1' if the login belongs to the hot list; else '0'
22	is_guest_login	Nominal	'1' if the login is a guest login; else '0'
23	Count	Continuous	Number of connections to the same host as the current connection in the past two seconds
24	srv_count	Continuous	Number of connections to the same service as the current connection in the past two seconds (same-service connections)
25	serror_rate	Continuous	% of connections that have "SYN" errors (same-host connections)
26	srv_serror_rate	Continuous	% of connections that have "SYN" errors (same-service connections)
27	rerror_rate	Continuous	% of connections that have "REJ" errors (same-host connections)
28	srv_rerror_rate	Continuous	% of connections that have "SYN" errors (same-service connections)
29	same_srv_rate	Continuous	% of connections to the same service (same-service connections)
30	diff_srv_rate	Continuous	% of connections to different services
31	srv_diff_host_rate	Continuous	% of connections to different hosts (same-service connections)
32	dst_host_count	Continuous	Count of connections having the same destination host
33	dst_host_srv_count	Continuous	% of connections having the same destination host and using the same service
34	dst_host_same_srv_rate	Continuous	% of connections having the same destination host and using the same service

SN.	Feature Name	Date Type	Description
35	dst_host_diff_srv_rate	Continuous	% of different services on the current host
36	dst_host_same_src_port_rate	Continuous	% of connections to the current host having the same source port
37	dst_host_srv_diff_host_rate	Continuous	% of connections to the same service coming from different hosts
38	dst_host_serror_rate	Continuous	% of connections to the current host that have an S0 error
39	dst_host_srv_serror_rate	Continuous	% of connections to the current host and specified service that have an S0 error
40	dst_host_rerror_rate	Continuous	% of connections to the current host that have an RST error
41	dst_host_srv_rerror_rate	Continuous	% of connections to the current host and specified service that have an RST error

Based on these datasets, five major attack classes were obtained.

2.2.1. The four major classes of attacks are.

2.2.1.1. Probing.

In this class, an attacker scans a network or host to gather known vulnerabilities and information about the host computer. An attacker with the map of machines and services offered to them on the network, uses the information to search for exploits. Probe attack abuses the computer's legitimate features, social engineering techniques. It is the most common class of attacks and demands less technical skills and expertise, *e.g.* Probe attacks are: ipsweep, Nmap, portsweep and satan.

2.2.1.2. Denial of service Attacks (DoS).

In this attack class, an attacker prevents the legitimate users from using services on the network, by overwhelming or flooding the system with request and consuming the resources by exploiting system's misconfigurations or by aiming the implementations bugs. DoS attacks can be classified on the basis of the services that an attacker exploits. Types of DoS attacks are: Smurf, Neptune, back, teardrop, pod and land.

2.2.1.3. User to Root/Super-User Attacks.

In this class of attacks, an attacker first hacks into a normal/legitimate user's account on the network and then exploits the vulnerabilities so that he can gain root access of the system. Regular buffer overflows is the most usual exploit in this attack class, and the reasons may be programming mistakes or environment assumptions. Types of User to Root attacks are: buffer_overflow, rootkit, loadmodule, perl.

2.2.1.4. Remote to Local/User Attacks.

It is a class of attacks where an attacker sends packets to a machine on the network, then exploits its vulnerabilities and illegally gains local access as a user. *E.g.* of Remote to local attacks are: ftp_write, spy, imap, warezclint, guess_passwd, warezmaster, ftp_write, multihop, phf.

2.3. Overview of SVM

In this part, brief explanation of the basic concepts of support vector machine is given. The theoretical explanation of the abilities of SVM to handle very high dimensionality of data is also provided. Which is a major advantage of support vector machine and hence a good classifier for experimentation of high dimensional network data.

2.3.1. Basic concept of SVM.

As described in chapter 1, soft computing techniques can be used to detect intrusions using there patterns and its deviation from normal class patterns. Support vector machine was invented by Vapnik in 1979 [4]. The basic SVM or linear SVM deals with two class problems, also known as a binary classification problem. Its ability of generalization is superior to other traditional learning methods. In this research work, number of support vectors are separated using a hyperplane. Figure 2.2 depicts the boundaries that are defined between two classes of training data consists of subsets of support vectors. The S support vector machine is normally used for

supervised learning method. Vapnik proposed the initial idea of support vector machine for the positive and negative samples which can be separated by a unique optimal hyperplane with the largest margin.

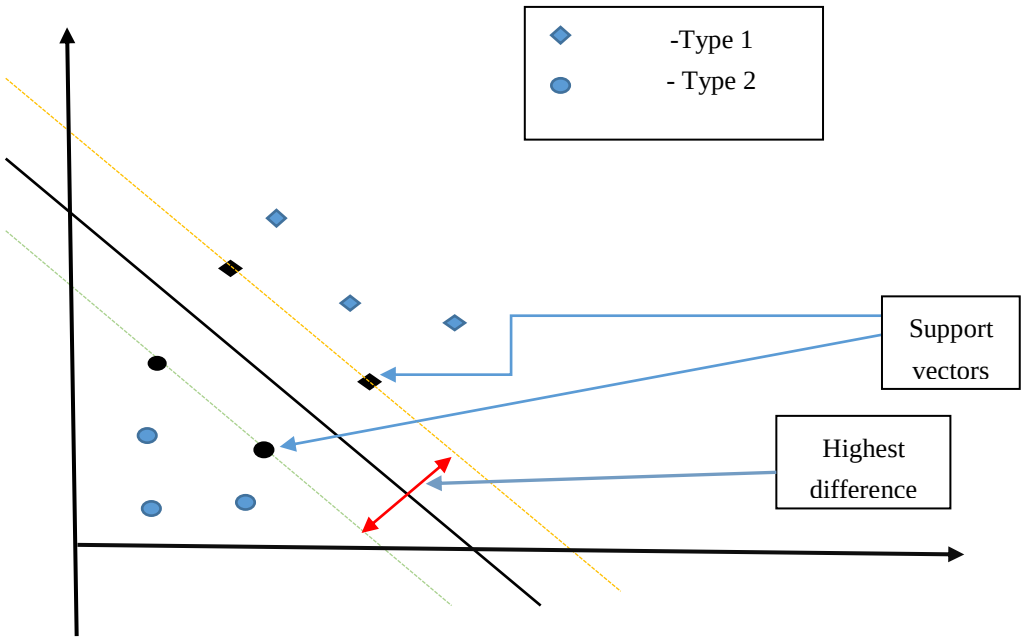


Figure 2.2.: The separation of two classes by hyperplane in linear classification.

The SVM, for binary classification is considered as the one of the most prominent learning algorithms [3, 5]. Reason of using support vector machine is; first, its performance is more accurate than any other classifier. Second, support vector machine is scalable, based on classification function it is insensitive to the number of data points and independent of the dimensionality of the feature space. So it can be used train a larger set of patterns and thus is better algorithm than the neural networks.

L. Khan et al. [3] proposed a clustering method comprising of support vector machine with dynamically growing self-organizing tree (DGSOT) algorithm for Intrusion Detection. In terms of accuracy and lesser training time, they compared the results of SVM and DGSOT with Rocchio Bundling Technique and random selection and concluded that there proposed method outperformed the rocchio bundling technique.

R. C. Chen et al. [2], used support vector Machine and Rough Set Theory to detect intrusions. They used Rough set theory to pre-process the dataset and reduce the dimensions. The features were selected using Rough Set Theory and then the dataset were sent to the SVM model for training and testing. They found that the method was effective to decrease the space and size density of the initial dataset. Their experiments showed improvement in detection of false positive.

Shih-Wei Lin et al. [8], proposed a method implementing support vector machine, simulated annealing and decision tree for anomaly intrusion detection. To improve the accuracy, support vector machine and simulated annealing were used to extract the features from KDD'99 dataset. Decision Tree and simulated annealing obtained decision rules from anomaly intrusion data. Simulated Annealing adjusted the parameters automatically for the decision tree and support vector machine was successfully detected the intrusions.

Yinhui Li et al. [7] used the gradually feature removal method to extract the features and constructed a classifier by combining clustering method, the support vector machine and ant colony algorithm for anomaly intrusion detection system.

2.4. Random Forest.

Breiman in 2001, proposed the random forest algorithm. The Random Forest is an ensemble of unpruned classification trees. It is based on a forest of trees using random inputs. Classification and regression trees are shaped differently in random Forest technique, as it construct tree employing a different bootstrap sample of the data. In normal trees, each node in the tree is split using the best split among all the variables. In case of random forest, a subset of predictors which is randomly chosen at that node is used to split up the node of that tree [10]. This orthodox approach happens to perform exceptionally well as compared with other classifiers, like Neural Networks (NN), discriminant analysis, *etc*. Also it is sturdy against the overfitting of data.

Furthermore, it has just two parameters only, one is the number of trees in the forest and other one is number of variables present in random subset at each node) and usually to their values, it is not very sensitive and hence tidier and user friendly.

Random forests are a popular method for feature ranking and feature extraction, since they are can be applied very easily. In common way, they require very little feature engineering and parameter tuning and in most random forest libraries the mean decrease impurity is exposed. But they have their own tricks and explanations, especially when data interpretation is concerned. With more correlated features in the dataset, the strong features might end up with low scores and the method can be biased towards variables with many categories. As long as this trick of correlated data is kept in mind, there is no problem to try it out on our data.

2.5. Basic concept of Snort

Snort is an open source lightweight Intrusion Detection System and the most popular network sniffing tool, that can log packets traveling through the network.it is one of the most popular Network Intrusion Detection System (NIDS). Since snort is an open source which means that the original source code of the program is accessible to everyone without any charge which means that it allows any number of people around the globe to analyze and contribute to the program building. Snort utilizes GNU general public license. Lawton (2002) discussed the availability and accessibility of open source software codes, which makes it easier for the hackers to solve the problem of defeating the security. But he also clarifies that closed source (commercial) systems can still be compromised. But, if the code is open source and available to all, the security holes are shut down as soon as they can be identified. This proves the advantage of open source in case of computer security.

Snort can be configured to operate in different modes like sniffer mode, Network Intrusion Detection System, packet logger and Intrusion Prevention System. Packet sniffing and logging are the elementary functions of snort. But the major reason snort is used for is intrusion detection. IPS is a new functionality added to the snort that

allows snort to drop the malicious packets or redirects it to another destination. Snort captures packets using libpcap [19] and decodes them and then forwards them to detection engine. Snort uses a set of rules to define whether the data is intrusive or not. Snort signatures or rules are regularly updated on the snort website, snort is flexible, meaning that it can use a previously logged traffic stored in a file on the system in the exact same pattern as it uses the live current traffic. It also supports variety of outcomes like, saving and storing of alerts to the databases or files, or generating a network traffic log of all the incoming traffic which can be processed later by the user.

The Snort captures packets from the network and maintain a log of it in the form of Tcpdump file. The data received by the Snort is represented to a user in a specified pattern, the information about the sender and the receiver along with other important information is stored in the header part of the packet. The purpose of the snort initially is to capture the packet on the network and display it to the user in human readable form.

<time><sequence_no.><source_ip><orientation_symbol><destination_ip><protocol>…**<Time_to _live>**

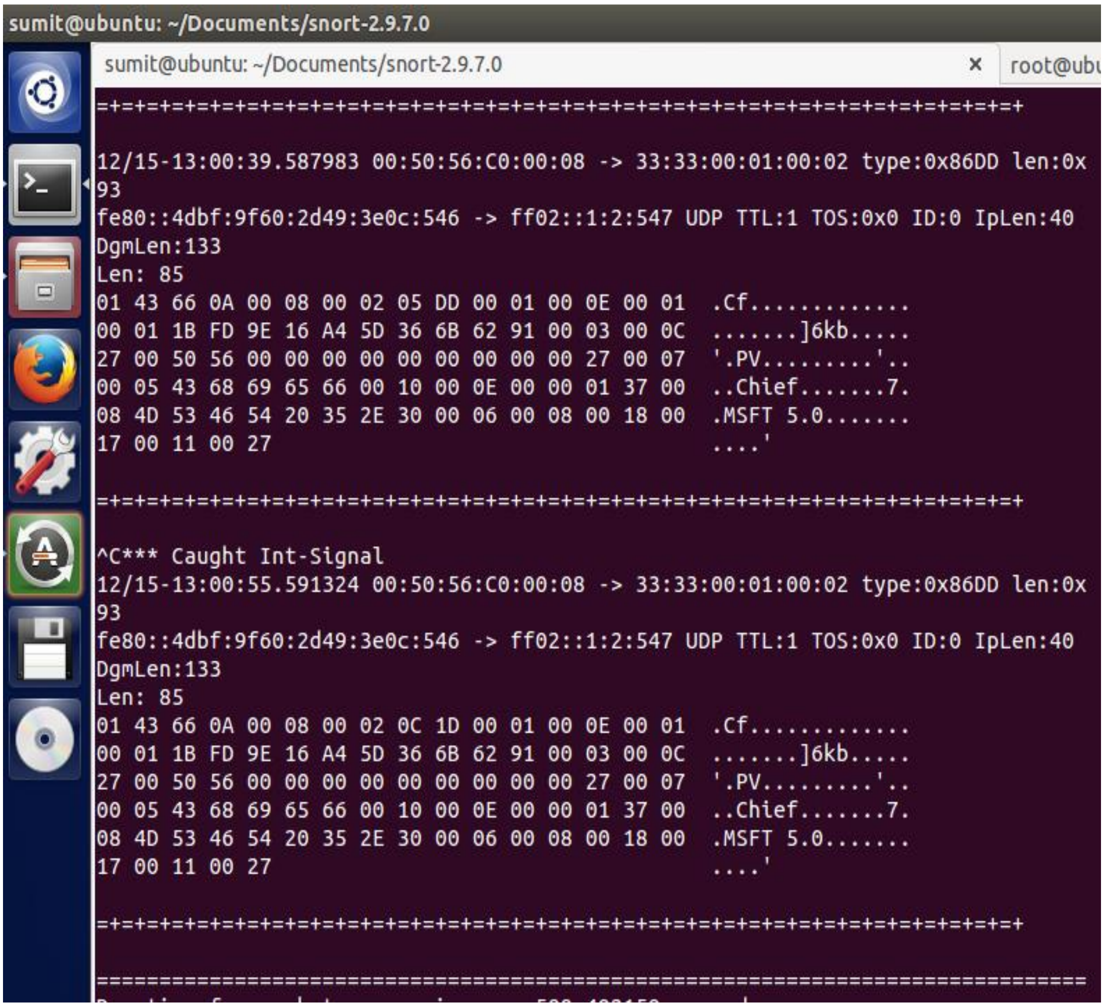


Figure 2.3.: Example of a data packet captured on network by snort.

Figure 2.3 shows the data packet captured on network by snort. The Snort keeps on capturing these types of packets on the network and from this packet data, useful information is extracted called as features (both general and specific) and there values. General features are those features that are common in all, which does not distinguishes one from the other, *e.g.* all humans have two hands and two legs, *etc*. Specific features are those features that helps in distinguishing one information from other, for example, all humans have different pitch of voice, facial features, *etc*.

CHAPTER 3

PROBLEM STATEMENT

In previous Chapters various methods and techniques that were used in the past and techniques that can be used in the near future in the field of intrusion Detection System are described. Since, everyone wants a completely secure and reliable solution to their network security problems. Despite so much research work done on the Intrusion Detection Systems, countless number of antivirus software, there are no reliable solutions yet, unless all the requirements of an intrusion detection system are fulfilled, which are time consuming, cumbersome and updating all the time. This chapter deals with the problems that are faced by the IDS developers for developing an efficient intrusion detection System, its incapability and issues regarding the development and organization of intrusion detection system.

Cybercrime are growing day by day, and so is the spending of funds on the development of effective and robust intrusion detection system. By the year 2017, it is expected that the Global Security Market budget will be $120.1 billion, which was $63.7 billion in the year 2011. The annual cost over global cybercrimes is $100 billion. The statistics show, that over 556 million users become victim of network attacks per year, which means 18 victims per second are getting affected by a potential attack.

The major reasons why the cyber-attacks happen are:

- **Cybercrime**: The crimes that are done using computer networks, like internet fraud, identity theft, *etc*.
- **Hacktivism**: Derived by “hack” and “activism”, it is the use of computers and networks to promote debates, sustain political ends *etc*.
- **Cyber warfare**: It is network based attempts to damage another nation’s computers through viruses or DoS attacks.

- **Cyber espionage**: The use of computer networks to gain illegal access to confidential information, generally held by government and organizations.

Now the question arises is why is that we cannot create an intrusion detection system that can stop such kinds of attacks and losses even after spending so much money on it. The available antivirus systems cannot provide the desire safety, because they are based on misuse intrusion detection system. These antivirus systems cannot cop-up with the new and intelligent methods implemented by intruders unless they are detected. For this problem Anomaly Intrusion Detection Systems were invented which could detect any undesired change in the network data or a deviation from normal standards of data on the network, which means that this feature makes it capable of detecting new intrusion types. But the major problems with the IDS are:

Problem 1: The software Based security systems cannot provide promising security to new types of threats.

Problem 2: The behaviour of monitored environment may change after certain period of time and require retraining of the system.

Problem 3: If the training set itself contains attacks, the system will consider malicious behavior as normal.

Problem 4: False Positives and False Negatives.

The overcome these problems the following Objectives has been carried out for the research work:

Objective 1: The machine learning techniques such as support vector machine (SVM) and random forest (RF) are applied on the intrusion detection system (IDS) for anomaly detection.

Objective 2: The comparison of accuracy and sensitivity of both support vector machine and random forest methods is done.

Objective 3: To verify and validate the results of proposed methodology using Snort.

CHAPTER 4

EXPERIMENTS PERFORMED

The purpose of this thesis is to provide an effective, robust and efficient Intrusion Detection System. In this chapter, the experimental setup and implementation part used to carry out the research is explained in detail. In first section, the system configuration on which the experiments are performed is explain. In the second section, the data sets used in the experimentation, its configuration, correlation, pre-processing and reduction into smaller dataset is explained. In third section, the experiments using the technologies and tools that are used in the research work while working on the problem, to find the solution are explained.

4.1. Methodology Used.

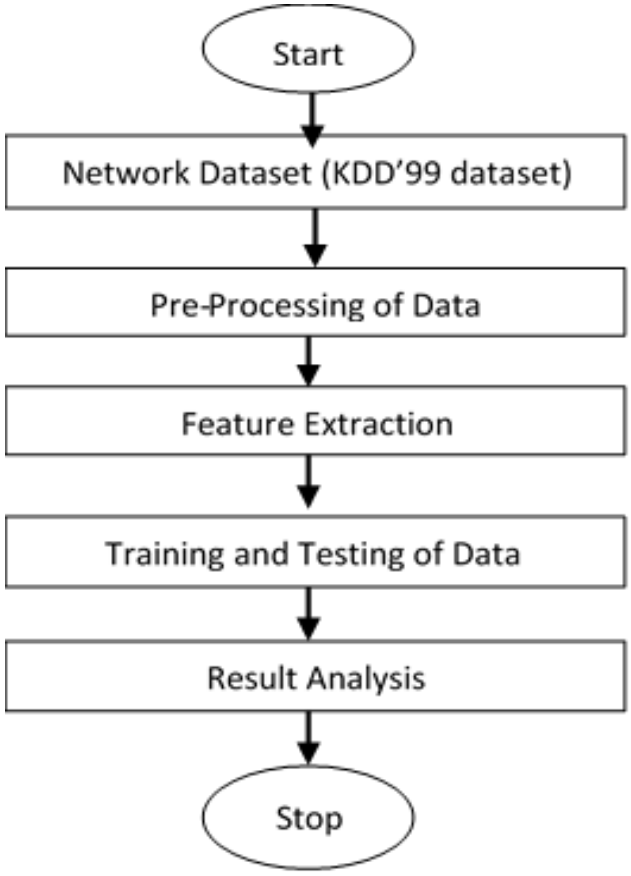


Figure 4.1. Flow chart of the methodology used.

4.2. System Configuration.

The most crucial thing in performing the experimentation is the speed of the Intrusion Detection running on the host computer which is running the actual software implementation and the algorithm. Hence configuration of the system is directly proportional to the outcomes of the experimental results. To evaluate the speed of the model, all the tests are run on the same computer having Intel corei5-3230M, 2.60 Ghz CPU and 4 Gb RAM. Since the software implementation of the model used is based on LIBSVM and Random Forest in R-tool, a descent processing elements that could bear the load of processing of the data on these models is needed.

4.3. Pre-processing of KDD'99 Dataset.

In this section the raw data set of DARPA intrusion detection dataset is downloaded from Internet [15]. Initially the data set contained various duplicate values, redundant data and many missing values. The first task is to clear the data from such values, which is a very slow and tiring process, as all the values were to be removed manually from such big data, this takes a lots of time and patience. After the removal of unnecessary and useless data entries, table 4.1 shows the sample dataset after the cleansing of data.

Table 4.1. Sample dataset.

Class	F1	F2	F3	…	F41
normal	0	21	206	…	0
smurf	0	23	1032	…	0
normal	9908	21	2634	…	1
normal	357	21	334	…	0
normal	0	21	4539	…	0

A total of 125973 records or instances are used. Table 4.2 describes the number of instances of particular type of attack and the total number of instances in the dataset are taken for further testing.

Table 4.2. Network attack types and classification

S. No.	Attack Names	Number of Instances	Attack Classes
1	Smurf	2646	DOS (Denial of Service)
2	neptune	41214	
3	back	956	
4	teardrop	892	
5	pod	201	
6	land	18	
7	satan	3633	Probe
8	ipsweep	3599	
9	nmap	1493	
10	portsweep	2931	
11	warezclint	890	R2L (Remote to Local)
12	guess_passwd	53	
13	warezmaster	20	
14	imap	11	
15	ftp_write	8	
16	multihop	7	
17	phf	4	
18	spy	2	
19	buffer_overflow	30	U2R (User to Root)
20	rootkit	10	
21	loadmodule	9	
22	perl	3	
23	normal	67343	Normal Data
Total		**125973**	

4.4. Feature Extraction using Random Forest Technique.

After the Dataset are generated, a test of correlation among the attributes or features is performed. The correlation model used on the features is Pearson correlation model. Correlation model is a statistical measure that signifies the interdependence of two or more random variables, generating values from +1 and −1 inclusive. The result gave us a matrix of correlated features. Figure 4.2, illustrates the correlation among attributes. The value of correlated features wanders from +1 to -1, where +1 is total positive correlation, 0 is no correlation, and −1 is total negative correlation.

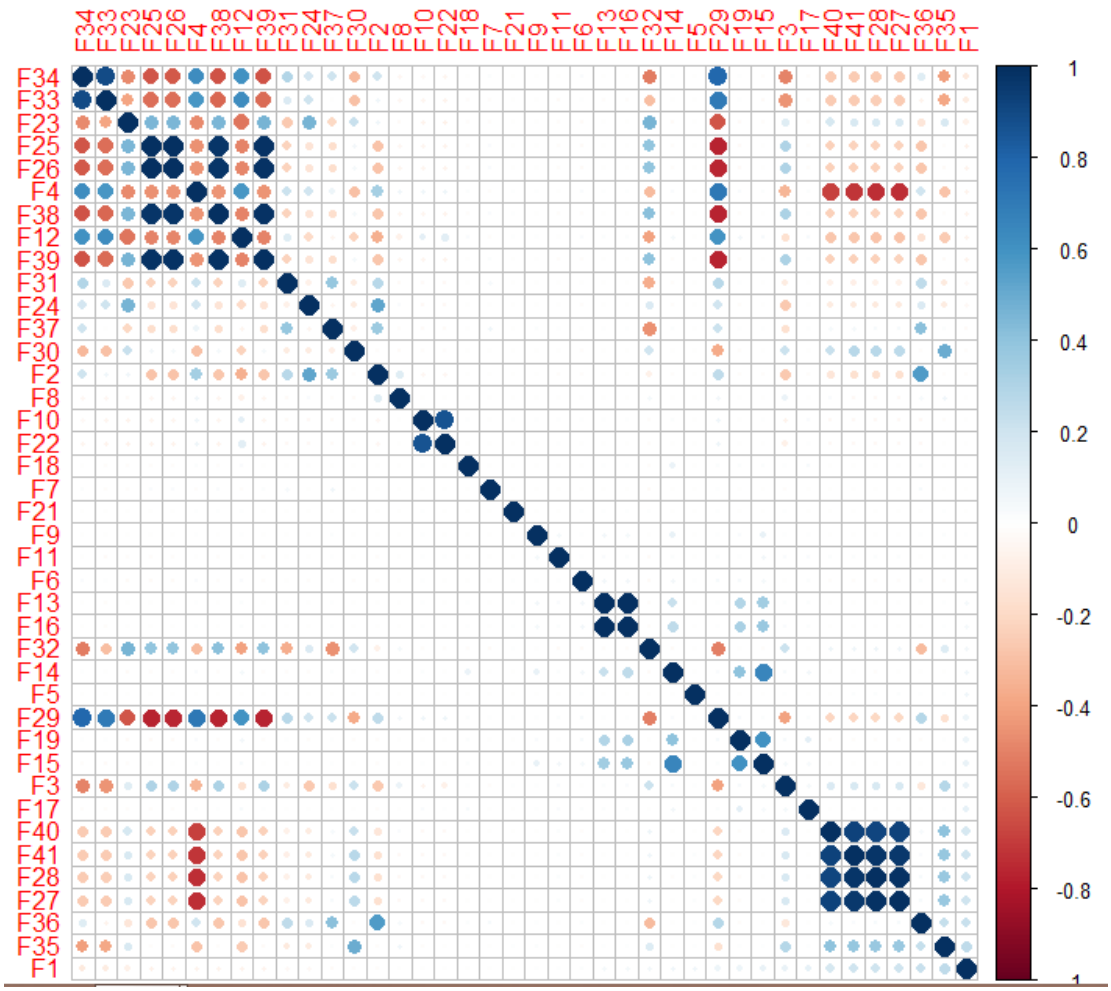


Figure 4.2: Correlation between each feature

The correlation model of the dataset is necessary, to find out those features that are dependent on other features, which does not carry their own significance. Those dependent features are pointed out from the data by using the correlation model. After the correlation test of dataset, another test of feature significance detection is done using random forest technique. The result of feature extraction test produced, the features with their significance weight and the feature ranks as shown in table 4.3.

Table 4.3. Importance of features

Rank	1	2	3	4	5
Feature	F36	F10	F2	F5	F6
Significance	30.45	30.01	29.75	29.57	27.68
Rank	6	7	8	9	10
Feature	F35	F33	F40	F8	F32
Significance	26.31	25.69	25.40	24.86	24.72
Rank	11	12	13	14	15
Feature	F37	F34	F1	F24	F4
Significance	23.90	22.12	21.34	20.20	18.55
Rank	16	17	18	19	20
Feature	F3	F27	F38	F41	F12
Significance	17.73	15.72	15.60	15.31	15.23
Rank	21	22	23	24	25
Feature	F39	F23	F22	F13	F11
Significance	14.83	14.04	13.93	13.43	11.55
Rank	26	27	28	29	30
Feature	F31	F25	F26	F30	F29
Significance	11.24	10.53	10.22	9.39	8.64
Rank	31	32	33	34	35
Feature	F14	F28	F16	F18	F17
Significance	7.84	6.75	6.49	6.48	6.29
Rank	36	37	38	39	40
Feature	F15	F19	F7	F9	F20
Significance	3.80	3.17	3.12	1.75	0.00
Rank	41				
Feature	F21				
Significance	0.00				

Based on this feature significance results the features that are not necessary to be included in the dataset are removed and hence the final dataset on which training and testing is to be performed is generated which contains 30 recommended features. These thirty features are then pre-processed for training the support vector machine and random forest. Since the data is reduced and contains only those features that are needed, the time required in training the support vector machine and random forest will be reduced.

4.5. Procedure for using the Support Vector Machine.

4.5.1. Pre-processing of dataset for training LibSVM.

LibSVM tool of the SVM software has been used in this research work. The LibSVM requires data points to be represented as a vector of real number. To train the SVM, change all the alphabetic character values to numeric values in the dataset. Since the LibSVM does not take alphabet characters as input to maintain a reference table and assign each feature and feature value a numeric code. In this, a simple numeric characters to represent the values. For example, as there are five major attack classes, described as code 1 to 5.

The format of the training and testing file will be same as:

<Class> <feature'1'>:<value> <feature'2'>:<value>… <feature'N'>:<value>

An example of Sample dataset format used for SVM training and testing:

101 1:0 2:21 3:3022 4:4010 5:195 6:313 7:0 8:0 9:0 10:1…30:0

202 1:0 2:21 3:3011 4:4006 5:0 6:0 7:0 8:0 9:0 10:0…30:0

101 1:0 2:21 3:3022 4:4010 5:231 6:2239 7:0 8:0 9:0 10:1…30:0

304 1:0 2:21 3:3022 4:4010 5:302 6:443 7:0 8:0 9:0 10:1…30:0

402 1:0 2:22 3:3010 4:4010 5:44 6:127 7:0 8:0 9:0 10:0…30:0

An example of corresponding reference table of codes is shown in table 4.4.:

Table 4.4. A sample reference table

S. No.	Attack Names	Reference code	Attack Classes
1	Smurf	201	DOS (Denial of Service) {code -20}
2	neptune	202	
3	back	203	
4	teardrop	204	
5	pod	205	
6	land	206	
7	satan	301	Probe {code- 30}
8	ipsweep	302	
9	nmap	303	
10	portsweep	304	

S. No.	Attack Names	Reference code	Attack Classes
11	warezclint	401	R2L (Remote to Local) {code- 40}
12	guess_passwd	402	
13	warezmaster	403	
14	imap	404	
15	ftp_write	405	
16	multihop	406	
17	phf	407	
18	spy	408	
19	buffer_overflow	501	U2R (User to Root) {code- 50}
20	rootkit	502	
21	loadmodule	503	
22	perl	504	
23	normal	101	Normal Data {code-10}

4.5.2. Training LibSVM

After the processing and conversion of complete data, which again is a very tiring task. The data is initialized into the LibSVM. “LibSVM 3.19” a tool of SVM is used, for the purpose of classification of Intrusion Data. LibSVM toolbox is a set of pattern recognition and regression machine learning software developed by National Taiwan University (*LIN et al.*). Save training and testing files into the windows folder inside LIBSVM 3.19. Use functions “svm-train” and “svm-predict”. The tool is operated from the command prompt, to train a training file, svm-train command is used and a model file is automatically generated. svm-train mainly completes the SVM training on training data set and svm-predict is used to predict the classification result on test data set using the model formed by the svm-train, it classifies the data according to the feature values. According to the data the support vector machine creates hyperplanes based on the values and standard deviation. The dataset that has similar values and lie under the same hyperplane are considered in one class and rest according to their respective hyperplanes.

The distribution of dataset for training of LibSVM and Random Forest is shown in figure 4.3.

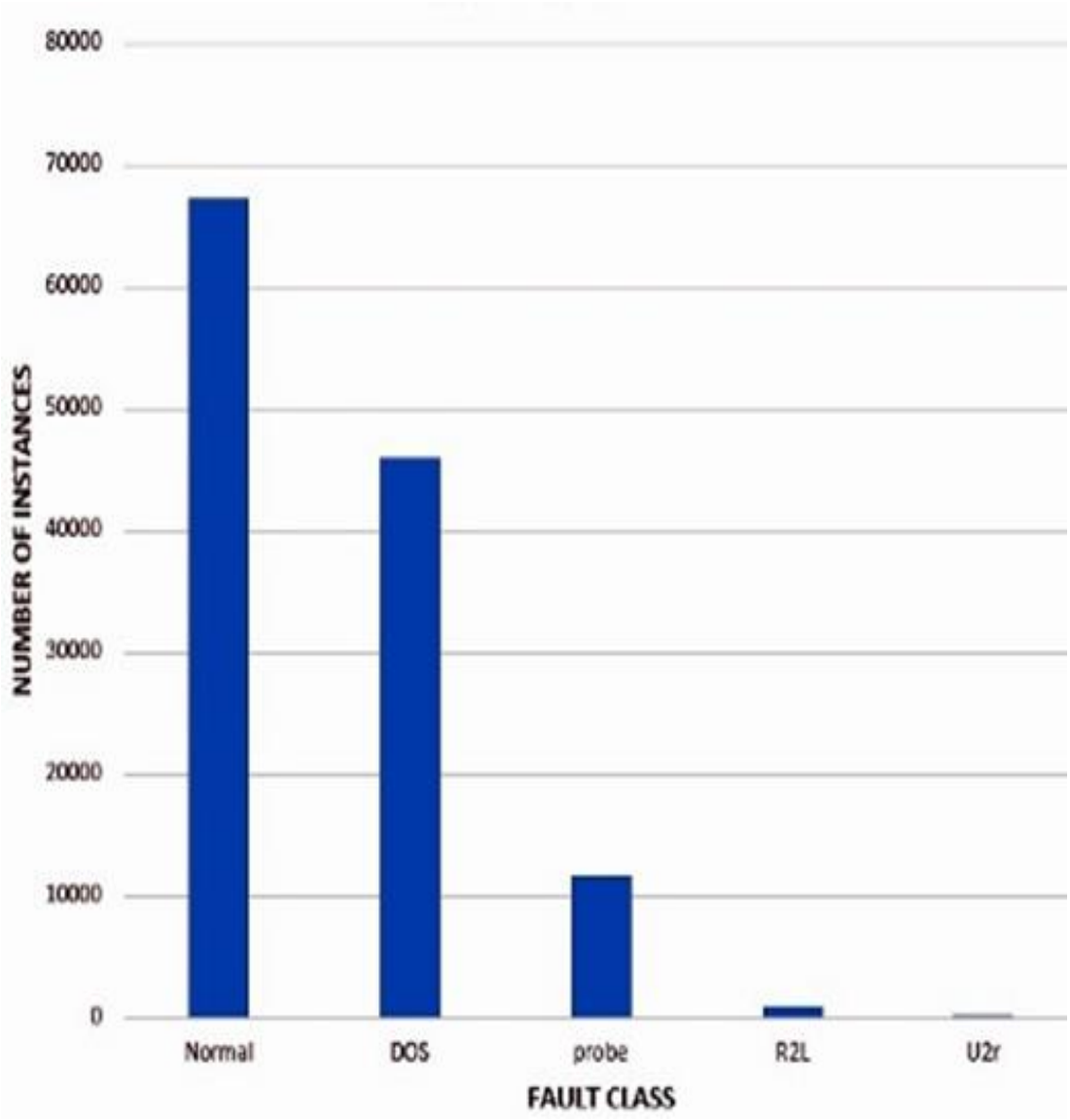


Figure 4.3: The distribution of dataset for training of libsvm and random forest is shown in

4.6. Making SNORT rules:

Packet sniffing and logging are the elementary functions of snort. But the major reason snort is used for is Intrusion Detection. IPS is a new functionality added to the snort that allows snort to drop the malicious packets or redirects it to another destination. Snort captures packets using libpcap [23] and decodes them and then forwards them to detection engine. Snort uses a set of rules to define whether the data is intrusive or not. Snort signatures or rules are regularly updated on the snort website, snort is flexible, meaning that it can use a previously logged traffic stored in a file on the system in

exactly the same way as it uses the live traffic. Snort also supports variety of outputs, such as saving alerts to databases or files, or creating a network traffic log of all the received traffic which can be processed later by the user. Snort has inbuilt rules to detect intrusions on the network. Inbuilt Snort rules are stored in snort.config file. But user can also create his own rules to detect intrusion or remove particular systems to interact with the user. Snort gives users lot of independence by letting him create his own rules in local.rules file or he can also include his rules in the snort.config file. A rule is formally defined as shown:

4.6.1. The format of snort log file

<action><protocol><source_ip><source_port><direction><destination_ip><destination_port><rule_options>

e.g. 1:- alert tcp any any -> any 80

[Create an alert for all tcp traffic where destination port is 80]

e.g. 2:- log tcp any any ->192.168.1.0/24 25

[This rule logs all tcp traffic coming from any source and any port to given network where port no is 25]

The term action tells what response is to be given to the user in case if the condition in the rule matches, when compared against an internet packet. “Alert” is the most common rule action, which means saving alert data to a file or data base for later.

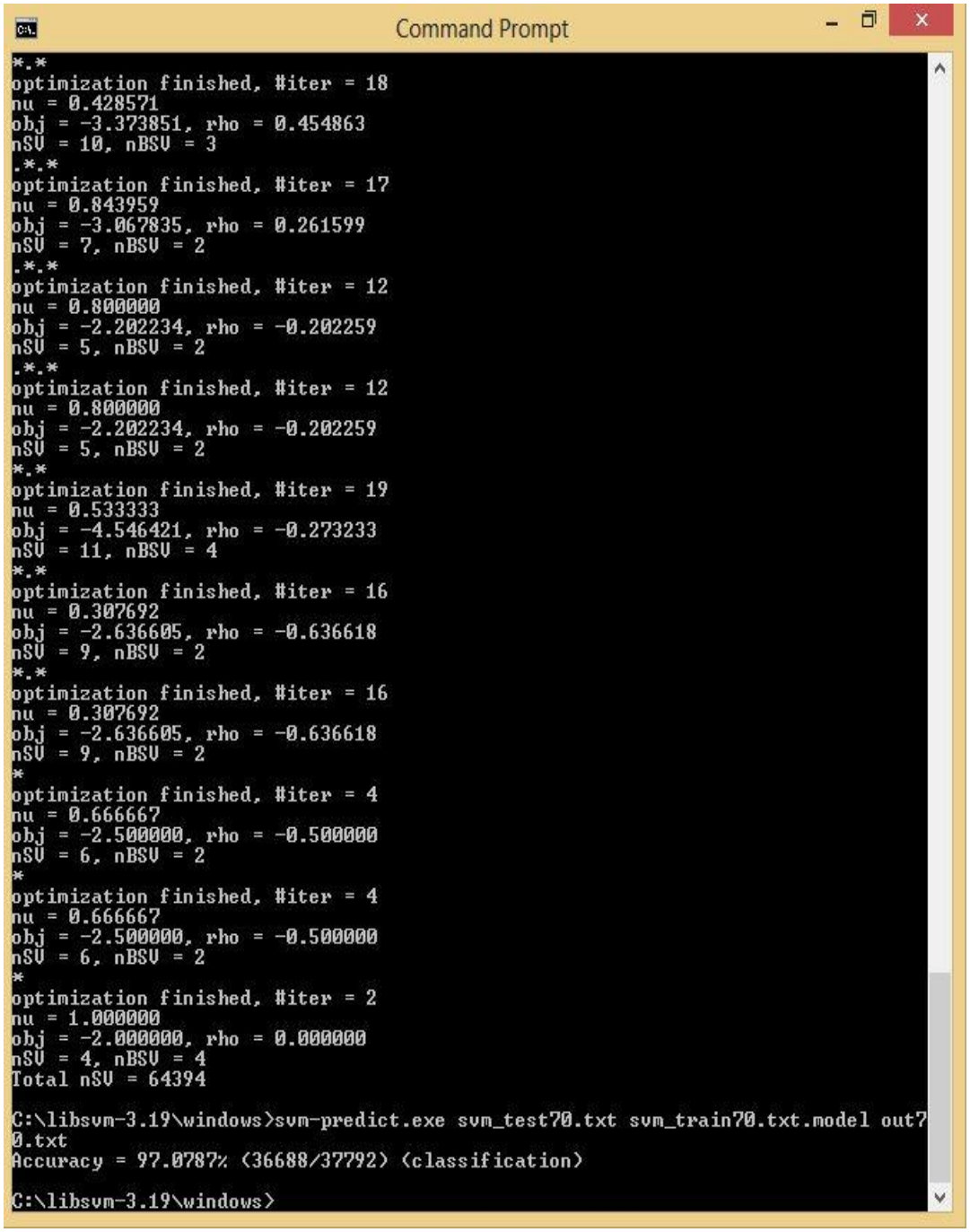


Figure 4.4.: Snap shot showing LibSVM accuracy at 70-30% partition of data

4.7. Model Evaluation

There are many different methods to measure the performance of the prediction, these methods depend upon the dataset they are used upon or on the application. A brief discussion upon the performance of the methods is explained in this section. The formula used for both Random Forest and Support Vector Machine is given as under.

$$CLASS \sim f(F1, F2 \ldots \ldots, F40, F41) \qquad (1)$$

The accuracy and sensitivity *(C, S)* as a measure for determining the performance of both the machine learning models are considered. To determine the accuracy and sensitivity, a confusion matrix is generated, depicting the information about actual and predicted classification done by the classifiers. In this matrix, the diagonal elements represent the number of instances that are true, that is, the diagonal elements are identified as true according to its given respective label. On the other hand, the elements that are off-diagonal are mislabeled or identified wrongly by classifier. In confusion matrix, if the diagonal value is high, the accuracy of the classifier is high too and visa-versa. If *n* is the total number of classes, then the value of C_{ij} of a confusion matrix of size $n \times n$ is number of patterns of class *i* predicted in class *j*.

The accuracy of the classifier can be calculated by the given formula:

$$Accuracy = \frac{\sum_{i=1}^{n} c_{ii}}{\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}} \qquad (2)$$

However, the classification accuracy may produce inaccurate results in cases where there is high variation in the number of instances in the classes. Therefore the accuracy of the classifiers as a pair of *(C, S)* values are calculated, where *C* is the overall Accuracy and *S* is the minimum of all the sensitivities amongst all the classes. The Sensitivity, $S_i$for the class *i* is defined as the total number of correct patterns predicted in class *i* with respect to the total number of patterns in class *i*., the significance of the class *i* can be calculated by the given formula [6]:

$$S_i = \frac{c_{ii}}{\sum_{j=1}^{n} c_{ij}} \qquad (3)$$

The sensitivity (S) of the classifier will be the minimum of all the sensitivities as shown in equation (4) [6].

$$S = \min(S_i; i = 1,2 \ldots, n) \qquad (4)$$

The correct accuracy (C) for the classifiers, that is, the rate of all correct predictions is defined in equation (5) [6].

$$C = \frac{1}{n}\sum_{i=1}^{n} S_i \qquad (5)$$

Based on the given equations the accuracy and sensitivity of support vector machine and Random Forest are calculated. The outcomes of the experiments are described in chapter 5.

CHAPTER 5

TEST RESULTS

In this section, the predicted results of random forest and support vector machine on the testing dataset are analyzed. The dataset are partitioned into 50-50%, 60-40%, 70-30% and 80-20%. The accuracy is calculated using equation (5) and is shown in table 5.1, it is observable that the support vector machine has the highest accuracy and sensitivity, in pair of (93.918, 0.91), (95.414, 0.94), (97.078, 0.95) and (97.997, 0.97) on the training and testing partitions respectively.

Further, 10-fold cross validation is done to measure the robustness of support vector machine. Figure 5.1 and figure 5.2 describe the accuracy and sensitivity for the 10 folds. Cross validation results show consistent performance in terms of accuracy. Therefore, from the results it is evident that support vector machine outperforms the random forest.

Table 5.1. Performance comparison of SVM and RF on different training and testing partitions in terms of accuracy and sensitivity.

TRAINING – TESTING PARTITION	**MODELS**	
	Random Forest (*C*,*S*)	**SVM (*C*,*S*)**
50-50%	(83.989, 0.81)	(93.918, 0.91)
60-40%	(85.440, 0.83)	(95.414, 0.94)
70-30%	(85.190, 0.84)	(97.078, 0.95)
80-20%	(85.921, 0.84)	(97.997, 0.97)

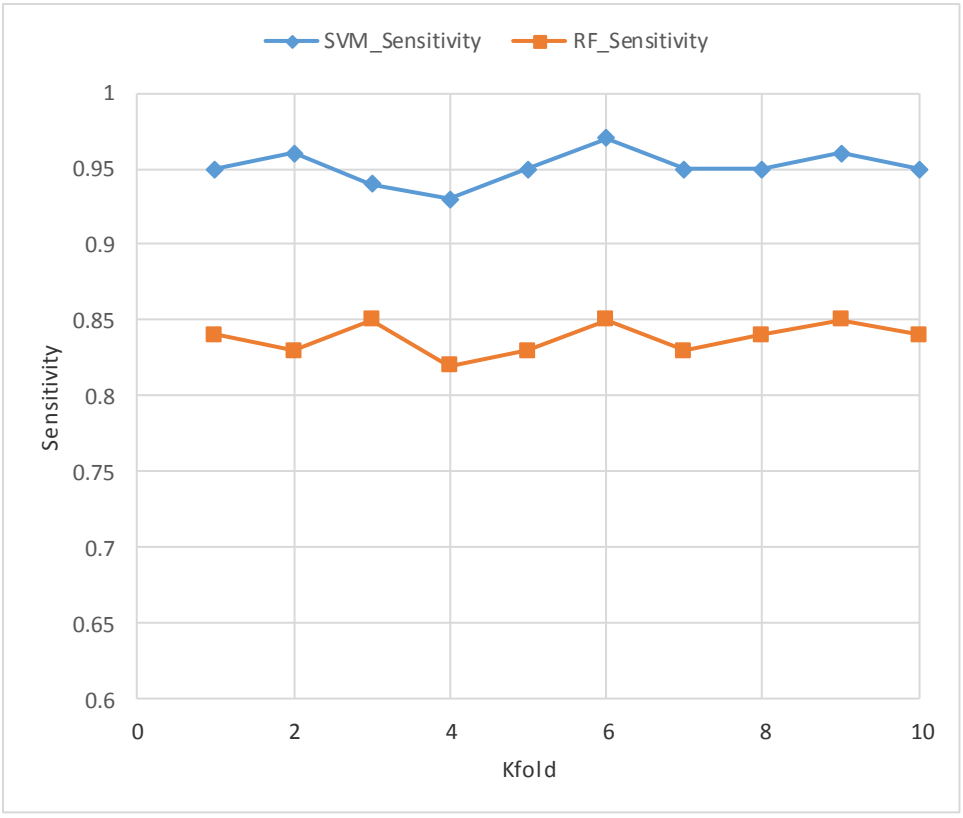


Figure 5.1.: 10-Fold cross validation of accuracy on training-testing dataset(70-30%) to predict attack classes using SVM and RF.

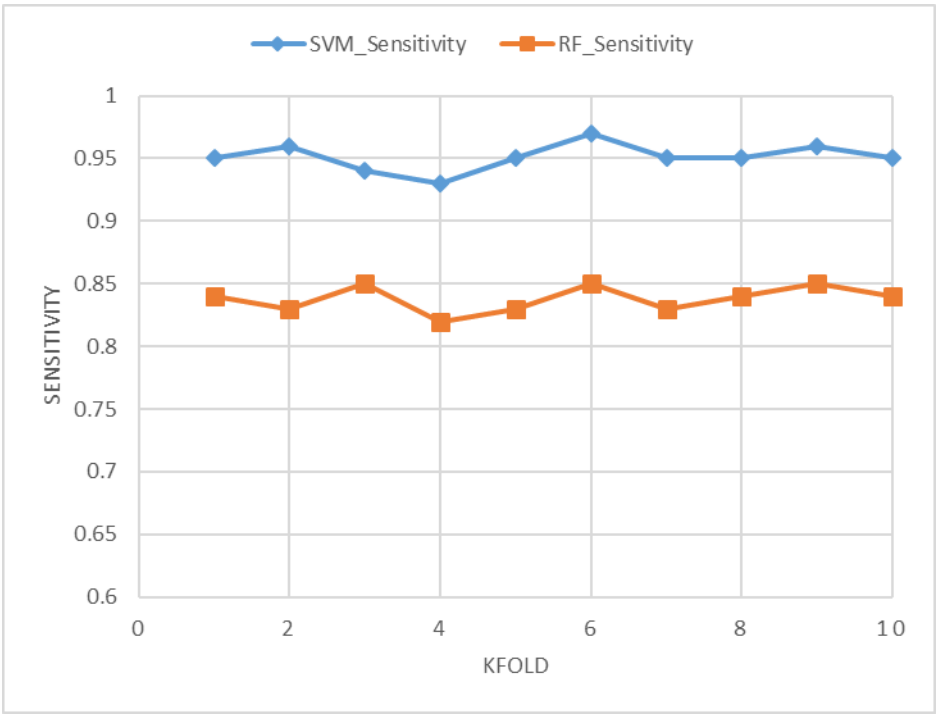


Figure 5.2.: 10-Fold cross validation of sensitivity on training-testing dataset (70-30%) to predict attack classes using SVM and RF.

CHAPTER 6

CONCLUSION

The analysis of two classification models, support vector machine and random forest for anomaly intrusion detection system is done. The performance of these two models has been observed and studied on the basis of their accuracy and precision on the test data. The experiments proved that both the classifiers are capable of handling high dimensional data and still produce accurate results. Since, the accuracy and sensitivity of anomaly based IDS results using support vector machine is approximately 97% and 0.95 respectively, which is much higher than that of random forest's 85% accuracy and 0.84 sensitivity. The results indicates that the ability and accuracy of the support vector machine classifier out performs from that of random forest. The random forest took lesser time in training and testing of the dataset as compared to support vector machine. The accuracy in the results produced using Random Forest is lower as compared to the support vector machine. Due to this reason, the results produced by the support vector machine only, are considered to create rules in the snort. The Snort also verified and validated the results that were produced by the support vector machine.

Future Work

The Experimental results shows the efficiency of both Random Forest and support vector machine, which proves that the machine learning techniques can be successfully applied to the Anomaly Intrusion Detection System. The research work can be extended by implementing various other Soft Computing Techniques in Anomaly Intrusion Detection.

REFERENCES

[1] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms," *J. Netw. Comput. Appl.*, vol. 28, no. 2, pp. 167–182, Apr. 2005.

[2] R. C. Chen, K. F. Cheng and C. F. Hsieh, "Using rough set and support vector machine for network intrusion detection," *International Journal of Network Security & Its Applications*, vol. 1, no. 1, April 2009.

[3] L. Khan, M. Awad, and B.M. Thuraisingham, "A new intrusion detection system using support vector machines and hierarchical clustering," *Int'l J. Very Large Data Bases*, vol. 16, no. 4, pp. 507-521, 2007.

[4] V. N. Vapnik, *The nature of statistical learning theory*. New-York: Springer Verlag, 1995.

[5] C. Tsai, Y. Hsu, C Lin, and W. Lin. "Intrusion detection by machine learning: A review." *Expert Systems with Applications*, vol. 36, no. 10, pp. 11994-12000, 2009.

[6] J. C. F. Caballero, F. J. Martinze, C. Hervas, and P. A. Gutierrez, "Sensitivity vs. accuracy in multiclass problems using memetic Pareto evolutionary neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 750–770, May 2010.

[7] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method*", Expert Systems with Applications*, vol. 39, no. 1, p 424-430, January 2012.

[8] Shih-Wei Lin, Kuo-Ching Ying, Chou-Yuan Lee and Zne-Jung Lee, "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection," *Applied Soft Computing*, vol. 12, pp. 3285-3290, 2012.

[9] Guyon I, Elisseeff A. "An introduction to variable and feature selection", Journal of Machine Learning Research, vol. 3,pp. 1157–1182, 2003.

[10] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Trans. Syst. Man Cybern. C*, vol. 38, no. 5, pp. 649–659, 2008.

[11] J. Zhang and M. Zulkernine, "A hybrid network intrusion detection technique using random forests," in Proc. *1st International Conference on Availability, Reliability and Security*. USA: IEEE CS, pp. 262–269, 2006.

[12] Mukkamala S., Janoski G., and Sung A.H., "Intrusion detection using neural networks and support vector machines," In Proc. *IEEE International Joint Conference on Neural Networks*, pp.1702-1707, 2002, 2002.

[13] Md. Al M. Hasan,, Md. Nasser, B. Pal, and S. Ahmad. "Support vector machine and random forest modeling for intrusion detection system (IDS)," *Journal of Intelligent Learning Systems and Applications*, vol. 6, pp. 45-52, 2014.

[14] D. Novikov, R. Yampolskiy, and L Reznik.. "Anomaly detection based intrusion detection," *Third International Conference on Information Technology: New Generation*, Page(s):420 – 425, April 2006.

[15] Darpa Intrusion Detection datasets [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[16] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. *2009 IEEE Int. Conf. Comput. Intell. Security Defense Appl.*, pp. 53–58.

[17] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[18] Internet World Stats [online]. http://www.internetworldstats.com/stats.htm.

[19] James Riden, "Using the 'snort' intrusion Detection System," Dec. 26, 2005 [Online]https://www.debian-administration.org/artical/318/Using_the_'snort'_Intrusion_Detection_System.

[20] R. Kohavi and G.H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, 97(1-2):273–324, 1997.

[21] W. Lee and S.J. Stolfo, “A framework for constructing features and models for intrusion detection systems”. *Information System Security*, 3(4):227–261, 2000.

LIST OF PUBLICATIONS

1. Sumit Gangwal, V. P. Singh, “Anomaly Intrusion Detection System Using Machine Learning Models,” International Conference on Soft Computing Techniques & Implementations, ICSCTI 2015, IEEE, Faridabad, India, Oct.8-10,2015. [**Communicated**]