Intrusion detection using Machine learning Models

Submitted by:

Anupam Misra (2019201082)
Akash Kumar (2019201046)
Aditya Gupta (2019201067)
Aditya Mohan Gupta (2019201047)
Deeksha Sahu (2019201068)
Kritika Singh (2019201075)
Saptarshi Manna (2019201070)

To:

Professor Kannan Srinathan

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

November 2020

Table of Content

ABSTRACT

The advancement in networks has indeed increased the need for designing and executing 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 have been classified into five different types of attacks, i.e. normal, Probe, U2R, R2L, and DOS. The algorithms used in this paper are the 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 outperforms 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.

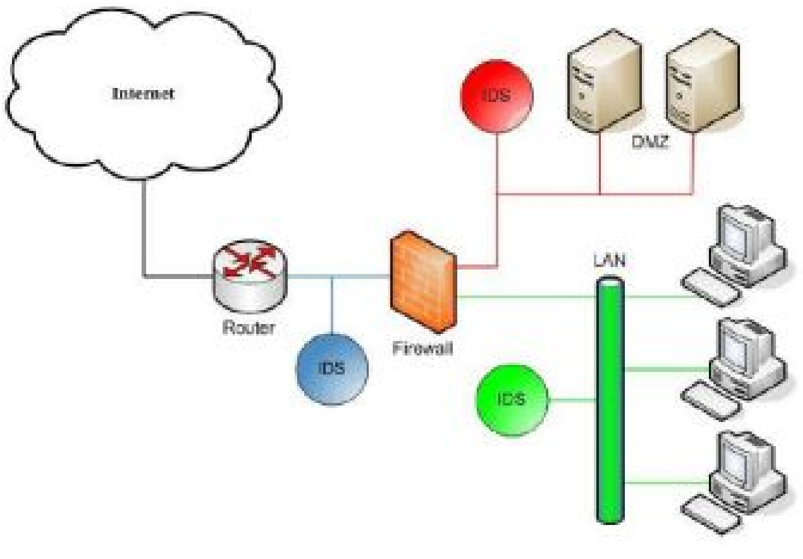


Figure.: A computer network with IDS systems at various locations.

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.

Overview of IDS.

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:

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.

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.

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.

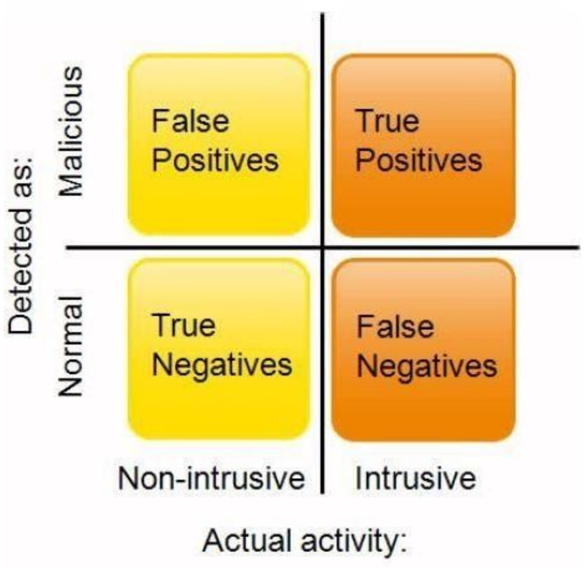


Figure 2.1.: False positives, true positives, true negatives and false negatives.

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 four major classes of attacks are.

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.

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.

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.

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.

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.

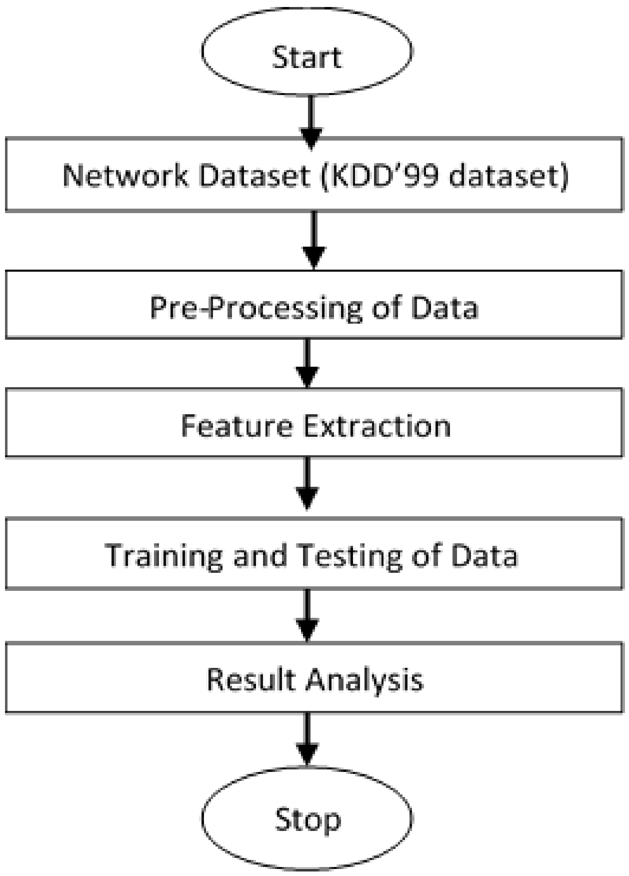
The Snort captures packets from the network and maintain a log of it in the form of Tcpdump file. The data received by the Snort is represented to a user in a specified pattern, the information about the sender and the receiver along with other important information is stored in the header part of the 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> <proto col>…**<Time_to _live>**

Methodology

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.

Procedure used:

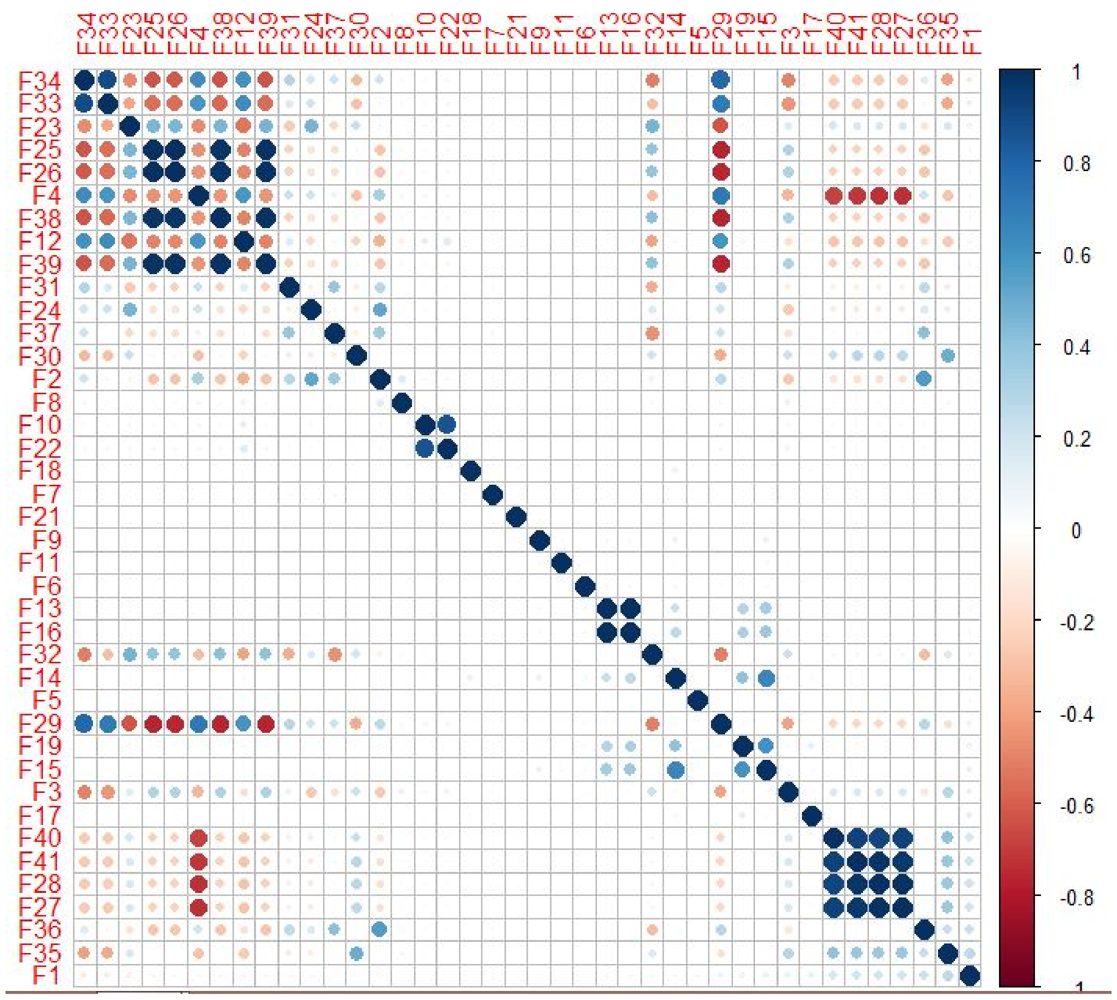


System Configuration

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.

Feature Extraction

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.



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 below:

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 these 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.

Making Snorting 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 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.

Format of snort log file

<action><protocol><source_ip><source_port><direction><destination_ip><destinati
on_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.

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 $_{n \times n}$ 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 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 S_i 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, 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:

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

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

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

The correct accuracy (C) for the classifiers, that is, the rate of all correct predictions is defined in

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

equation (5).

Based on the given equations the accuracy and sensitivity of support vector machine and Random Forest are calculated.

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.

Reference

[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.