AN APPROACH USING ENSEMBLE CORE VECTOR MACHINES FOR NETWORK IDS

[1]P. Chandra Sekhar Reddy, [2]Regonda Nagaraju, [3]S.Srikanth Reddy

ABSTRACT--*Past many activities carried by common man were manual. Today's stay connected world there's huge usage of e-services which made man's activities online, along with these services security concerns also increased. Many researchers proposed efficient Intrusion Detection systems are in practice; but still hackers manage to attack the systems to intrude. This paper proposes an efficient intrusion detection system using Ensemble Core Vector Machine approach, where algorithms work on the basis of Minimum Enclosing Ball concept. It detects the attacks like: R2L, U2R, Probe and DoS attack. For each type of attack, a CVM classifier is modeled. KDD Cup'99 datasets are used for training and testing the classifiers. This approach uses Chi-square test for selecting the relevant features for each attack and a weighted function is applied to these features for the dimensionality reduction. The test results verify that this model achieves high efficiency in all the four attacks with less computation time compared to the existing approaches.*

Keywords-- *Intrusion Detection System, core vector machine, minimum enclosing ball, attacks, chi-square test*

I. INTRODUCTION

The rapid development and popularization of the Internet services have brought many problems i.e., cyber attacks, to protect against it, Cyber security involves a set of technologies which protects computers, networks and data from attackers. It may include firewall, antivirus software and IDSs. IDS is a software which can detect intrusions like unauthorized traffic, logins, data duplications, destructions and abnormal behaviours. Many Intrusion Detection mechanisms are set in use.

Two types of IDSs are discussed:

1. Signature-based misuse detection based on known behavior.
2. Anomaly-based detection based on abnormal behaviour.

Drawbacks of current IDS include their inability to prevent attacks by themselves, requires an experienced engineer for administering frequent occurrences of false alarms. For some extent to get rid of these difficulties dataming techniques are applied. Here in most cases accurate detection is attained at the cost of more computation time. But nowadays automatic detection of attacks with minimum false alarms in considerable time is essential. This paper proposes an intrusion detection system implemented using a data mining based classifier called Core Vector Machines (CVM). CVM is an advanced version of Support Vector Machines. It is based on the concept of

[1] *Assistant Professor, St. Martin's Engineering College, Secunderabad, Telangana, India*

[2] *Associate Professor, Department of IT, St. Martin's Engineering College, Secunderabad, Telangana, India*

[3] *Assistant Professor, Department of IT, St. Martin's Engineering College, Secunderabad, Telangana, India*

Received: 22 Feb 2020 | Revised: 13 Mar 2020 | Accepted: 05 Apr 2020

minimum enclosing ball. It can produce less false positives and has a low computation overhead compared to SVM. KDDCup'99 dataset is being used to train and test the classifier.

II. RELATED WORKS

Intrusion detection systems can be based on either known attacks (Signature-based misuse detection) or on abnormal behaviour (Anomaly based detection). Data mining techniques are applied for building both these types of IDSs. In [1], a comparison of core vector machine and ensemble classifiers is done and CVM is selected as the best one in the field of intrusion detection. Principal Component Analysis is used here as the feature selection mechanism. The drawback of this approach is that PCA cannot be applied to a single record and thus cannot be used in real time systems. The various data mining classification techniques used in intrusion detection are discussed in [2]. From this paper, it is clear that the performance of classifiers will be different for different types of attacks. It also discuss about the different publicly available data sets for the training and testing of classifier models. In [3], a hierarchical concept using CVMs is proposed. It shows a higher performance for attacks like R2L and U2R compared to other classifiers. In [5] a bagging ensemble of decision tree is used for network intrusion detection. It shows about 81-99% accuracy, but it takes more time for training and testing compared to CVM. SVM is being used for intrusion detection in [6]. It proposes a hybrid approach of filter and wrapper models for selecting important features. [7] shows that applying AdaBoost improves the detection rate of Naive Bayesian network while keeping false positives in a low rate. [10][11][12] describe the concept behind Core Vector Machines and their features. The paper [15] gives an overall idea about intrusion detection systems and the applicability of data mining in that field.

III. PROPOSED METHOD

A data mining based classifier rarely shows all the desired features like high detection rate, low false positive rate and less computation time in an acceptable way. In the proposed method, Core Vector Machine is used as the classifier which can solve this problem to some extent. In this method, Chi-square test is used for feature selection. The architecture for the proposed method is shown in the Figure 1. Both training and testing is done using the KDD Cup'99 dataset. It has 41 features and these features are broadly classified into three categories: basic, content and traffic features. The basic and content features are being used here for training and testing. That is, 21 features out of the 41 are used.

3.1 Training

Training consists of mainly two phases:

1) Pre-processing

2) CVM modeling

In pre-processing, Chi-square test is applied on the 21 features or attributes inputted. Chi-square test gives the correlation coefficient for each of the 21 attributes. Correlation coefficient of an attribute shows how much the output (here attack label) changes for a small change in the value of that attribute. It means that, if the correlation

Received: 22 Feb 2020 | Revised: 13 Mar 2020 | Accepted: 05 Apr 2020

coefficient of an attribute is large, it influences the final output greatly. A small change in the value of that attribute results in a significant change in the final output. Ten features are selected from the incoming 21 features based on the result of the chi-square test. These are the features with the ten highest correlation coefficients.

The next step in pre-processing is the application of a weighted function to these ten attributes. A weight is assigned to each attribute proportional to its correlation coefficient. That is, highest weight is assigned to the attribute with the highest correlation coefficient and so on. Then the ten features are mapped on to two values representing the x and y coordinates of a point in 2D space. x is calculated as the weighted sum of five features and y is calculated as the weighted sum of the rest. Now two values (x & y) are obtained for a connection which represents all its major functions that will help us to build the classifier model. A simple appropriate mathematical function is also applied to the input data set in order to separate the normal connections and attacks as far as possible. For example, in the case of DoS attacks, root of sum of squares of x and y can be used as the mathematical function to separate attack connections from normal connections.

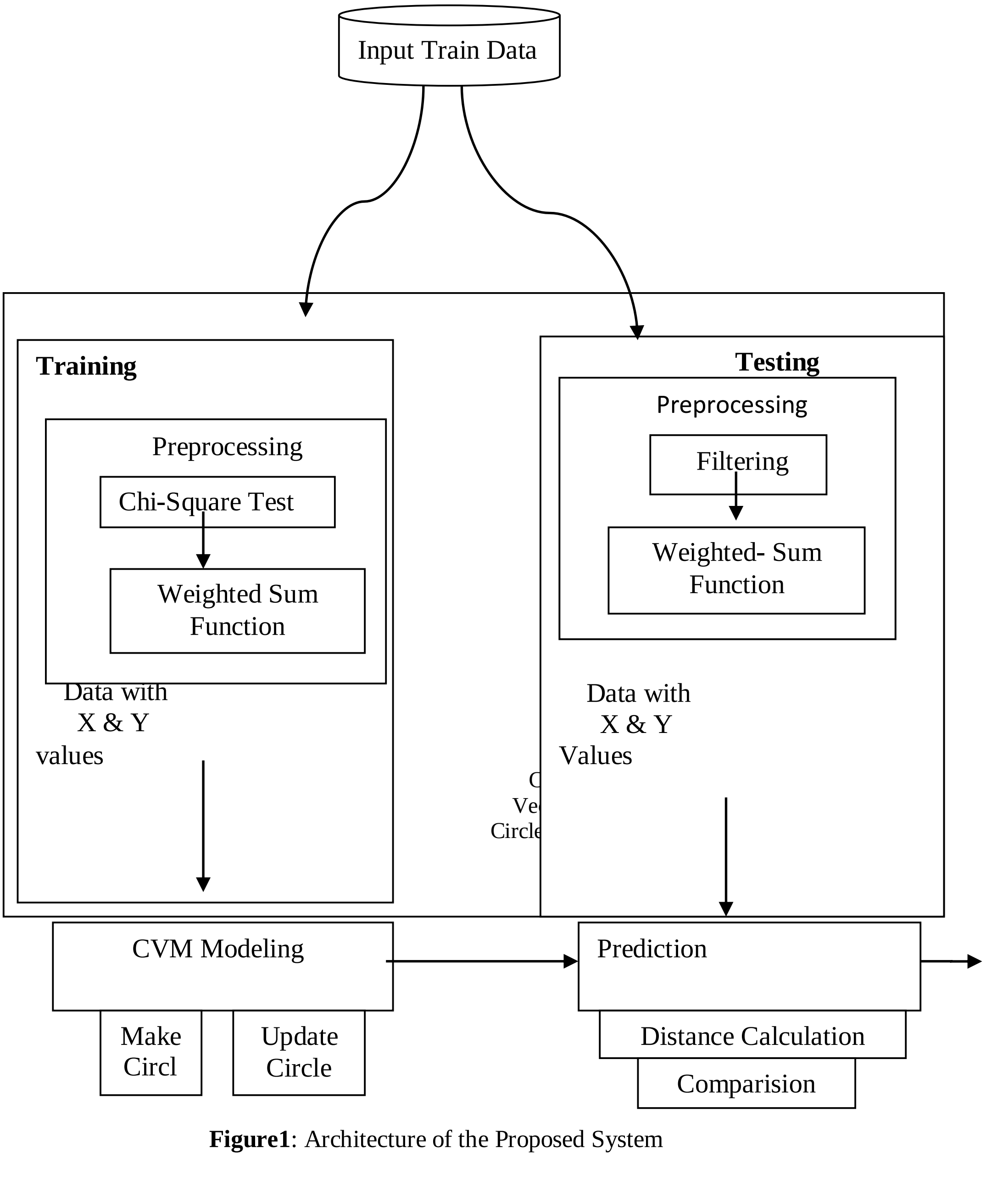

Figure1: Architecture of the Proposed System

The second phase of training involves CVM modeling. Core Vector Machine is actually based on the simple concept of Minimum Enclosing Ball (MEB). Since the multidimensional feature space here is mapped on to x and y coordinates as dimensionality reduction for simplicity, the concept of minimum enclosing circle is used here. In the proposed approach, a core vector classifier is modeled for the attack connections. The concept of the core vector classifier for IDS is described as the following:

1. At first, some initial attack points from the labelled input dataset are selected as the coreset.
2. A minimum circle is found such that it encloses all the points in the coreset.
3. Now increment the radius by a factor of ε if a similar point (here, attack point) is left outside.
4. Repeat this until no attack point in the training dataset is left outside.

There are mainly two operations in CVM modelling. First one is creating the circle (Make circle in Figure 1) and the second is the updating of radius (Update circle). In 'Make circle', first the coreset is selected. Then the first three points from the coreset are selected and their circumscribed circle is found. Iterating through the rest of the corepoints, the diameter of the circle is adjusted to the distance between the farthest corepoints. Radius is set to half of the diameter. In 'Update circle', the updating of radius by the ε factor happens. It iterates through the rest of the training set. Each time an attack point is seen, the radius is incremented by a factor of ε = 1+1e-10. This process continues until there are no more attack points to be included in the circle. After updating, the final training output obtained is the Core Vector Circle (X,Y,R).

3.2. Testing

Testing consists of two phases:

- Pre-processing
- Prediction

The pre-processing phase in testing involves filtering of attributes and a weighted function. The input data is filtered to get those ten major features selected according to the chi-square test results done during training. The weighted sum function is the same as that in the training phase. Weights of attributes are set proportional to the correlation coefficients of attributes obtained through the chi-square test. x and y values are found using the weighted sum function. Again, the same mathematical function is used for separating the attacks from normal connections as far as possible. In both training and testing, for each type of attack data this mathematical function can be different.

Prediction is the second phase of testing. It involves mainly two operations: distance calculation and comparison. The center (x,y) and radius of the core vector circle is obtained as the output of the training phase. Each time an input is given, pre-processing is done to map its features on to two values, x and y. Then the distance to that point (x, y) from the center of the core vector circle is calculated. This distance is then compared to the radius of the circle. If the distance d <= radius, it means that the point is inside the circle created. That is, it is an attack. Else if distance to center d > radius, it means that the point is outside the circle, which in turn shows that the point represents a normal connection.

In the KDD Cup'99 dataset, the major type of attacks discussed are DoS (Denial of Service), Probe attack, U2R (User to Root) attack and R2L (Remote to Local) attack. For each type of these attacks, a core vector circle is formed. When an input occurs, it is tested across all these four core vector circles (compared distances to their centers with their radii). Then their outputs are combined using some weighted voting function and the final output is predicted (that is, attack or not).

3.3. Feature selection

In most of the previous methods for intrusion detection using data mining algorithms, Principal Component Analysis (PCA) is used for feature selection. In PCA, the resulting principal components obtained are the combinations of the attribute values. PCA results in high accuracy results. The problem with PCA is that it should be applied to the whole data set each time the principal component value has to be found. It cannot be applied on to a single connection record. The principal components are found based on the extent of variance in the attribute values. So the principal component values found for a record in different datasets will be different.

In the proposed method chi-square test is used for feature selection. Chi-square test is performed on the training data set and the major attributes which correlates with the output label are found. The output of the chi-square test is the correlation coefficients of all the input attributes. Based on these coefficients, the most important features can be found at the training phase itself. Now, when a single record arrives, it is easy to filter out those main features. There is no need to recompute on the whole data set.

IV. IMPLEMENTATION

The proposed model is trained and tested on the KDD Cup'99 data set available in the UCI Repository. It has about 41 features which basically fall under three main categories: basic features, content features and traffic features. For this classification, basic and content features are being used. There are 21 features plus one output label. The features selected for classification are described in Table1.

The major attack types discussed in KDD dataset include DoS (Denial of Service) attack, Probe attack, R2L (Remote to Local) attack and U2R (User to Root) attack. There are sub categories for each attack and these sub categories are the given as the output labels. The subcategories of DoS attack include back, land, Neptune, smurf, pod and teardrop. Probe attacks are identified as satan, nmap, portsweep and Ipsweep. U2R attacks include loadmodule, buffer_overflow, rootkit and perl. R2L attacks include phf, guess_passwd, warezmaster, imap, multihop, ftp_write, spy, warezclient. At first all the categorical values in the data set are encoded to numbers.

On the 21 features extracted from the KDD dataset, chi-square test is applied to find the ten major attributes. In the proposed method, for each type of attack a classifier is modelled. The ten features found for different attacks may be different. The features selected for different attacks as a result of chi-square test is shown in Table2. These ten features are then converted to x and y coordinates using a weighted function. Then the CVM is modelled in two dimensions as the minimum enclosing circle. The center coordinates and radius of the circle are stored for future operations.

Table 1: Input Features

Feature	Description

Received: 22 Feb 2020 | Revised: 13 Mar 2020 | Accepted: 05 Apr 2020

Duration	Number of seconds taken by the connection
Protocol type	Transport layer protocol like tcp, udp, etc
Service	Network service that created the packet like telnet and ftp
Src_bytes	Number of bytes from source to destination
Dst_bytes	Number of bytes from destination to source
Flag	Status of the connection
Land	Is connection from the same host/port or not
Wrong_fragment	Number of wrong fragments arrived
Urgent	Number of packets flagged as urgent
Hot	Number of “crucial” indicators like entering a system
Num_failed_logins	Number of failed login attempts
Logged_in	If logged in or not
Num_compromised	Number of compromised conditions
Root_shell	If root shell is obtained or not
Su_attempted	If “su” command is executed or not
Num_root	Number of accesses to root
Num_file_creations	Number of files created
Num_shell	Number of shell prompts done
Num_access_files	Number of accesses to success control files
Is_hot_login	Is the login in hot list or not
Is_guest_login	Is the login is as guest or not

The performance metrics used to evaluate the performance of this classifier model are accuracy, detection rate, false-positive rate, training time and testing time.

True Positives (TP): Actual attack connections which are predicted as attacks.

False Positives (FP): Normal connections which are predicted as attacks.

True Negatives (TN): Actual normal connections which are predicted as normal itself

False Negatives (FN**):** Attacks which are predicted as normal

Accuracy: TP+TN/TP+FP+TN+FN

Detection rate: TP/TP+FN

False positive rate: FP/FP+TN

V. RESULTS AND DISCUSSION

In the KDD dataset the records for attacks are more. Also, out of these attack records, DoS records are the most. R2L and U2R attacks are less in number. Table 3 shows the results of the CVM classifiers modeled for the four different types of attacks. The training time is calculated for datasets containing 400 records and testing time is calculated for datasets containing 175 records. The comparison of training time and testing time for different attacks is shown in Figure 2.

Table 2: CVM results for different attack types

Attack	Accuracy	Detection rate	FP rate	Training time (sec)	Testing time (sec)
DoS	0.9905	0.9912	0.4714	0.0026	0.0006
Probe	0.9450	0.9616	0.2	0.0012	0.0012
R2L	0.7641	0.625	0.2131	0.0044	0.0006
U2R	0.9371	0.75	0.0490	0.0025	0.0006

The results of the proposed classifier are compared with other classifiers like Support Vector Machines (SVM) [13], Naive Bayesian classifier (NB) [8], Decision trees (DT) [14], Random Forest (RF) [9] and AdaBoost classifiers (AdaBoost DT). From the comparison it is clear that, CVM is the most suitable classifier for IDS as it takes little training and testing time (Figure 2), shows high detection rate and gives less false positives. Also, CVM detects all the four types of attacks in an acceptable way. Some of the other classifiers are only good at detecting some specific attacks.

From Figure 3 it can be inferred that compared to other classifiers, CVM and Naïve Bayesian classifiers show acceptable performances in detecting most of the attacks. U2R attack is not detected by any of the classifiers other than CVM. Decision tree and Random Forests show the worst accuracy. Figure 4 compares the detection rates of six different classifiers for all the four types of attacks. From the figure, it can be seen that CVM is the only classifier that detects all the four types of attacks.

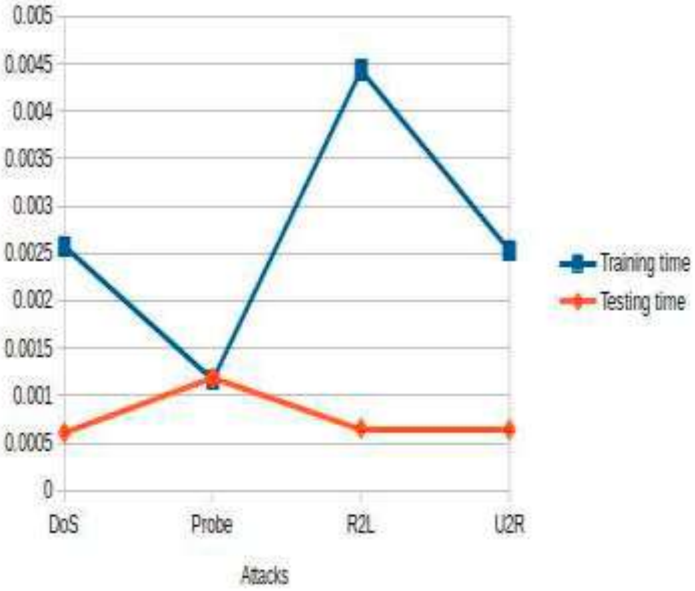

Figure 2: Time comparison for attacks

Received: 22 Feb 2020 | Revised: 13 Mar 2020 | Accepted: 05 Apr 2020

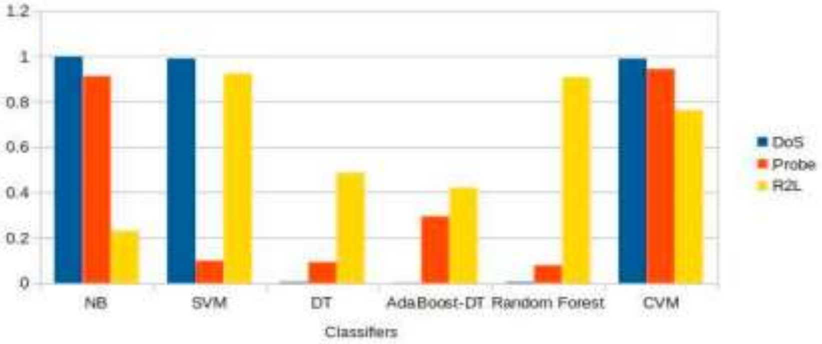

Figure 3: Comparison of accuracy

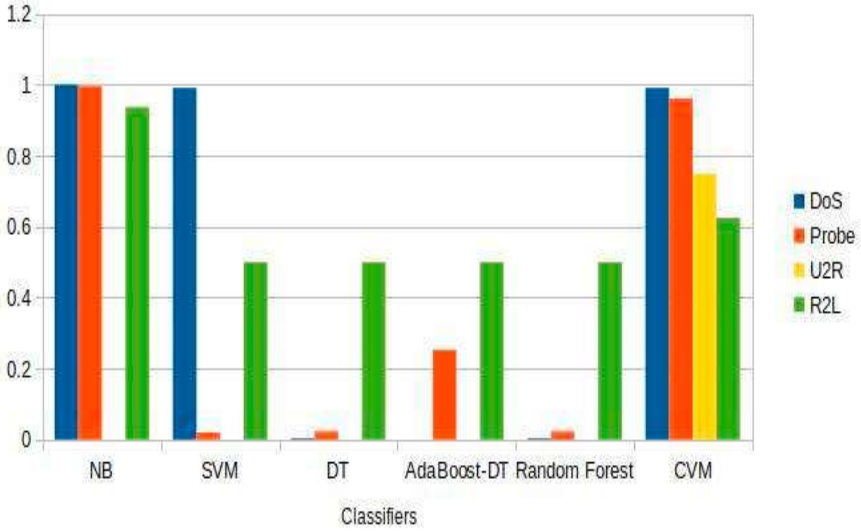

Figure 4: Comparison of detection rate

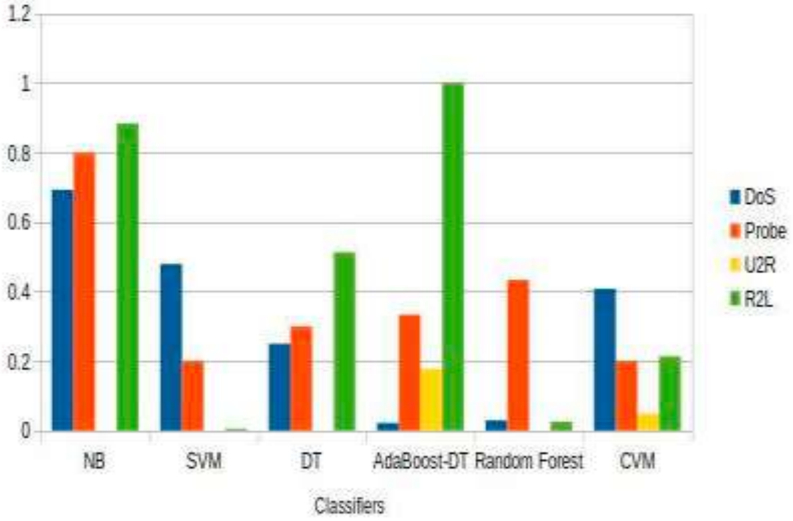

Figure 5: Comparison of false positive rate

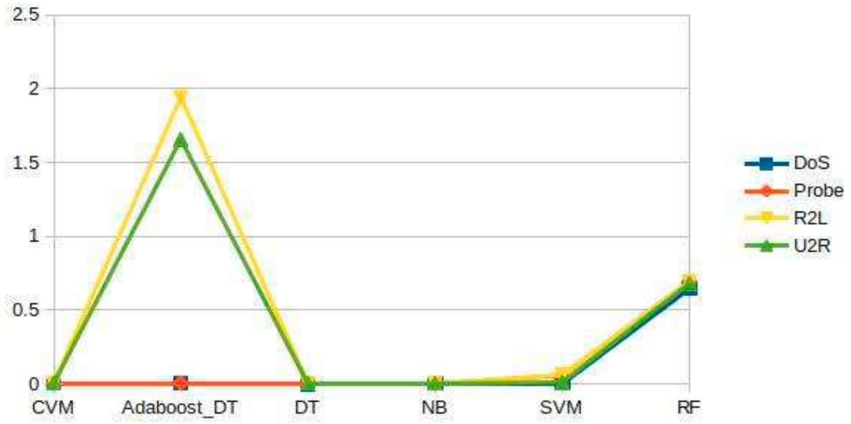

Figure 6: Comparison of training time

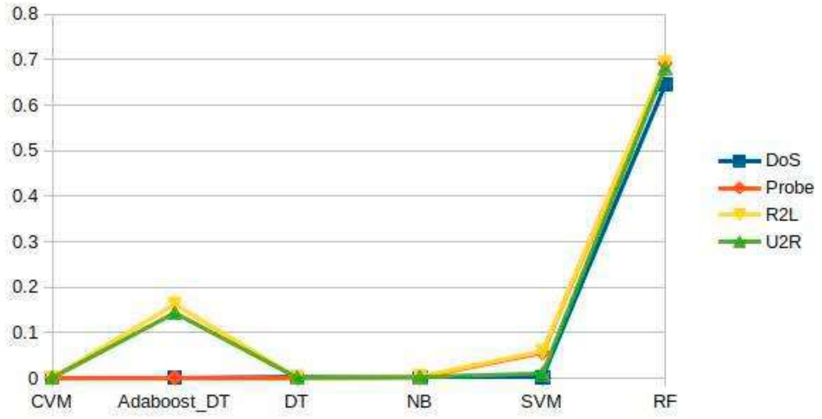

Figure 7: Comparison of testing time

Figure 5 compares the false positive rate of different classifiers. It is clear from the figure that even if there are some other classifiers which shows lower false positive rate than CVM, they show such performance only for some type of attacks. Figure. 6 & 7 compare the training time and testing time taken (in seconds) by different classifiers correspondingly. It can be seen that for ensemble classifiers like AdaBoost classifier the time taken is too high compared to other classifiers.

VI. CONCLUSION

Fast and automatic detection of attacks is essential nowadays. Data mining techniques are applied in the field of intrusion detection to build such systems. Different classifiers like SVM, Naive Bayesian Networks and AdaBoost classifiers are being used for this. But their performances are not optimal. Core Vector Machines are data mining based classifiers which show a somewhat optimal performance. It detects all types of attacks with an acceptable detection rate and false positive rate. It takes less time for training as well as testing compared to other classifiers like SVM which show similar performance. For each type of attack, a CVM model is built and then their results are combined using a weighted function. An ensemble CVM model is created in this way. It shows about 99% detection rate and 27% false positive rate.

REFERENCES

1. P.Amudha, S.Karthik, S.Sivakumari, "Intrusion Detection Based on Core Vector Machine and Ensemble Classification Methods", 2015 International Conference on Soft-Computing and Network Security (ICSNS -2015), Feb. 25 – 27, 2015, Coimbatore, INDIA.
2. Anna L. Buczak, ErhanGuven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection", IEEE Communications Surveys & Tutorials,18(2), Second Quarter 2016.
3. Ye Chen et al, "Hierarchical Core Vector Machines for Network Intrusion Detection", ICONIP 2009, Part II, LNCS 5864,520–529, 2009. Springer-Verlag Berlin Heidelberg 2009.
4. L.Dhanabal, Dr. S.P. Shantharajah, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms", International Journal of Advanced Research in Computer and Communication Engineering, 4(6), June 2015.
5. D.P.Gaikwad, Ravindra C. Thool, "Intrusion Detection System Using Bagging Ensemble Method of Machine Learning", International Conference on Computing Communication Control and Automation, 2015, IEEE Computer Society.
6. Wei Hu and Weiming Hu, "Network-based Intrusion Detection Using Adaboost Algorithm", Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05),IEEE Computer Society.
7. JayshreeJha, LeenaRagha, "Intrusion Detection System using Support Vector Machine", International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868.
8. Wei Li, QingXia Li, "Using Naive Bayes with AdaBoost to Enhance Network Anomaly Intrusion Detection", 2010 Third International Conference on Intelligent Networks and Intelligent Systems.
9. Mrutyunjaya Panda, ManasRanjan Patra, "Network Intrusion Detection Using Naïve Bayes", IJCSNS International Journal Of Computer Science And Network Security, 7(12), December 2007.
10. RifkiePrimartha, BayuAdhi Tama, "Anomaly Detection using Random Forest: A Performance Revisited", 2017 International Conference on Data and Software Engineering (ICoDSE),IEEE.
11. Ivor W. Tsang, James T. Kwok, Pak-Ming Cheung, "Core Vector Machines: Fast SVM Training on Very Large Data Sets", Journal of Machine Learning Research 6 (2005) 363–392.
12. Ivor W. Tsang, AndrasKocsor, James T. Kwok, "Simpler Core Vector Machines with Enclosing Balls", Proceedings of the 24 thInternational Conference on Machine Learning, Corvallis, 911-918, 2007.
13. Ivor W. Tsang, James T. Kwok and Pak-Ming Cheung, " Very large SVM training using core vector machines", Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS), Barbados, January 2005.
14. R.Ravinder Reddy, B.Kavya, Y Ramadevi, "A Survey on SVM Classifiers for Intrusion Detection", International Journal of Computer Applications (0975 – 8887), 98(19), July 2014.

15.]ShailendraSahu, B M Mehtre, “Network Intrusion Detection System Using J48 Decision Tree”, 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI).
16. ZibusisoDewa, Leandros A. Maglaras, “Data Mining and Intrusion Detection Systems”, (IJACSA) International Journal of Advanced Computer Science and Applications, 7(1), 2016.