IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-4, Issue-11 E-ISSN: 2347-2693

Intrusion Detection System Using Hybrid Classification Technique

Rajesh Wankhede[1*], Vikrant Chole[2]

[1] Department of Computer Science and Engineering, GHRAET, Nagpur
[2] Department of Computer Science and Engineering, GHRAET, Nagpur
e-mail: wrajesh8@gmail.com, vikrant.chole@raisoni.net

Available online at: www.ijcseonline.org

Received: Oct/09/2016 Revised: Oct/16/2016 Accepted: Nov/12/2016 Published: Nov/30/2016

***Abstract*—** Cyber Security is one of the key elements of any system. Breaching of cyber security can lead to loss of confidential and private data. To prevent the attacks on network an Intrusion Detection System Using Hybrid Classification Technique is proposed. This IDS uses a decision tree algorithm to classify the known attack types in the dataset and SVM is used to classify the normal data from the dataset, there by detecting the unknown attacks. Dataset used is the NSL-KDD Dataset.

Keywords- AdTree, SVM, NSL-KDD, IDS

I. INTRODUCTION

Data security and attack prevention are the two areas where a Intrusion detection system is always useful. Also computerized systems are increasing daily and so is the use of network based activities. Therefore Intrusion detection systems are used widely. Much Intrusion detection system mainly uses various anomaly detection techniques [1][2] and signature detection techniques. Many IDS have also been proposed based on various techniques such as neural networks[3], PCA[4], Gentic-fuzzy[5] rule, and decision trees[6]. IDS can also be classified on the basis of location where they are installed. Based on their installed location they can perform differently and detect attacks accordingly. These are classified as network based IDS and host based IDS.

A host based system is an IDS which monitors and analyses the system from within and also the network packets on its network interface. IDS are also network[7] based. One can see a Host based IDS as an agent that monitors whether anything or anyone, whether internal or external, has circumvented the system's security policy. A Host based IDS analyses the traffic to and from the specific computer on which the intrusion detection software is installed. A host-based system has the capability to observe the crucial system files and any attempt to overwrite these files. A network based IDS is used to analyse the network traffic to protect the system from network based threats. A network based IDS will analyse the inbound packets and scan for any suspicious patterns.

Also the Signature based IDS monitor's packets in the network and compares them to the known signatures which are pre-configured and pre-identified based on attack behaviour of previously known attacks. Alternatively the anomaly based IDS monitors the normal network traffic such as bandwidth range, types of protocols, and devices used to connect and dispatches an alert to the system manager on exposure of an anomalous behaviour. The signature based IDS detects attacks on the known attack signature type. This type of system can detect known attacks with low error rate, but it fails to detect the newly created attacks that do not have similar behaviour to the pattern available in the database. In contrast Anomaly based IDS can be suitable in recognizing the new attack pattern, but here the error rate is high. Thus, in order to solve the above two limitations, a intrusion detection method with hybrid technique which combines misuse detection method and anomaly detection method has been proposed.

II. RELATED WORK

Yonav Freund et.al, [8] proposes an alternating decision tree with boosting. This learning algorithm combines boosting and decision trees. In their paper they compared the alternating decision tree with the C5.0 algorithm. On smaller datasets Alternating Decision tree (ADtree) rapidly fits the data and ADtree reaches a very minuscule error after 50 iterations while the error of the stump boost remains large even after 200 iterations. This is a case in which large capacity of ADtree gives it an advantage. Comparing to the size of classifiers in all but three cases the classifiers generated by the ADtree are much smaller than those generated by C5.0 by boosting. The error performance of this algorithm is relative to that of C5.0 with boosting.

Tavallaee et.al, [9] presented a paper on KDD CUP 99 Data Set and after the analysis of the entire KDD dataset it showed that there were two important issues in the data set which affected the performance of evaluated systems, and thus results in a very poor evaluation of anomaly detection approaches. To overcome the issues, NSL-KDD was proposed, which contains selected records of the KDD data set. Although, the proposed data set suffers from some problems and may not be a perfect representative of existing networks, due to the lack of public data sets for network-based IDSs, they believe that the dataset still can be used as an effective benchmark to help researchers compare different intrusion detection methods.

Hong Kuan Sok et.al[10] presents a paper on using the ADTree algorithm for feature reduction. ADTree also gives good classification performance. In addition, its comprehensible decision rules endows the user to discover the features that heads towards better classification. This knowledge base facilitates to design a smaller dimension of support vectors for suitable classifier. The experiment supports the idea of using this algorithm as both knowledge discovery tool and classification. The classification task has been simplified and the speed increased drastically due to the reduced operations required to implement the classification.

III. PROPOSED METHEDOLOGY

A Hybrid approach is proposed for IDS, The idea behind this approach is to provide mechanisms for improving the detection precise. To introduce the proposed method, it is necessary to give a brief review of SVMs, AdTree and NSl-KDD dataset which is derived from KDD cup[11] dataset.

An Intrusion Detection System is built by using Alternating Decision Tree and SVM[12]. Dataset is gathered first. The dataset used is the NSL-KDD dataset for intrusion detection. For training purpose we have used 20%Train file downloaded from the same server. Here using SQL Server Management Studio the database was created.

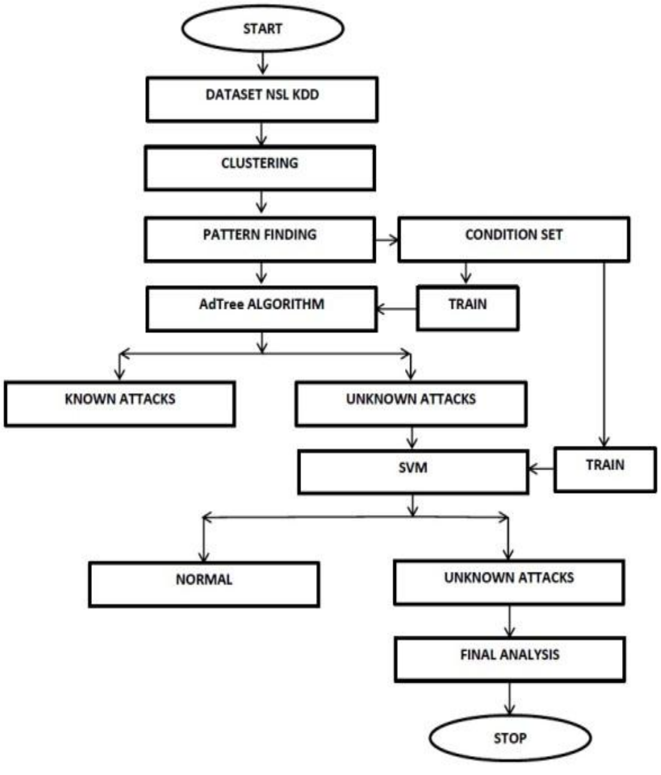
Module 1 consists of the Association mining. After the creation of the database cluster of each attack class were made. This was done by using a sql query, for eg. insert into [dbo].[Attack_back] select * from MainTbl where attack_type=‘back’. Main objective for clustering was to group the data to find the patterns. After clustering association rule is applied. In step one to find patterns count of the features is maintained. In step3 feature wise association is done i.e one feature is associated with rest of the features, to find pairs. After finding pair threshold is calculated. After calculating the threshold, In step3_support average threshold value is selected for the paired features and the rest are discarded. Based on step3_support, in pair3 association is done again on the paired features and in pair3_support average threshold value is calculated.

Threshold=(Max count of the feature – Min count of the feature)/2

After pair3 support the patterns are generated and stored in the database.

Module 2 consists of the Alternating Decision Tree. Here the decision tree is trained with patterns that are generated after mining. The decision tree is trained to detect the KNOWN attacks based on the patterns generated. Since the train dataset contained limited attack classes compared to the testing dataset therefore few of the attacks are ignored by the decision tree. The decision tree is trained to classify only the KNOWN attacks even the normal data is ignored by the decision tree. Based on the features and comparing each with the patterns generated the known attacks are classified. Adtree was chosen as the decision tree after comparing the various decision trees and classification methods such as neural networks, k-means, genetic algorithm etc. of which decision tree were found to be best for classification. This is presented in the review paper.

A support vector machine is used to classify the normal data. SVM is prepared to receive the data from the decision tree. The data sent for classification to the decision tree is in the form of single class also known as UNKNOWN attack. This unknown attack class consist of normal data as well as the attacks that were not classified by the decision tree. Therefore a one-class SVM is used to classify this data. Since the SVM is trained for normal data, thus it classifies normal data in one class as NORMAL and rest is classified as UNKNOWN.



System Architecture

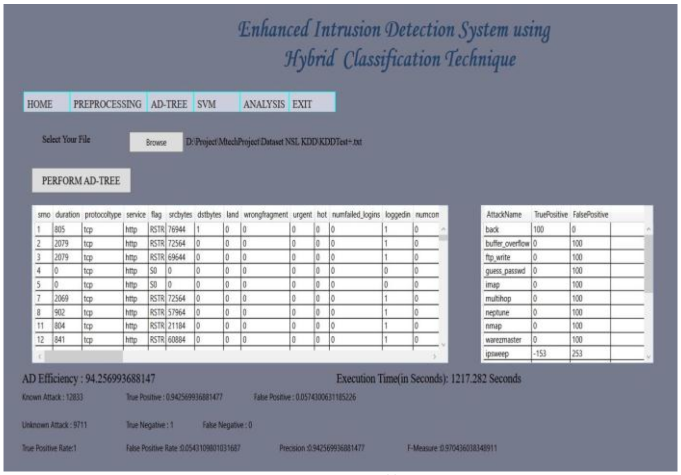
IV. IMPLEMENTATION DETAILS

The working environment for cloud computing where the proposed system is implemented is done using C# language on Microsoft Visual Studio 2012 and SQL Server Management Studio.

*Corresponding Author:
C. T. Lin
e-mail: ct.lin@hotmail.com , Tel.: +00-12345-54321

V. RESULT

The results are shown below. The result screen shows the known attack and unknown attacks. The known attacks in the case for Adtree are the ones that were present in the training dataset. As for the SVM the result shows normal and unknown attacks.


Enhanced Intrusion Detection System using
Hybrid Classification Technique
HOME
PREPROCESSING
AD-TREE
SVM
ANALYSIS
EXIT
Browse
PERFORM AD-TREE

Select Your File D:\Project\MtechProject\Dataset NSL KDD\KDDTest+.txt

srno	duration	protocoltype	service	flag	srcbytes	dstbytes	land	wrongfragment	urgent	hot	numfailed_logins	loggedin	numcon
1	805	tcp	http	RSTR	76944	1	0	0	0	0	0	1	0
2	2079	tcp	http	RSTR	72564	0	0	0	0	0	0	1	0
3	2079	tcp	http	RSTR	69644	0	0	0	0	0	0	1	0
4	0	tcp	http	S0	0	0	0	0	0	0	0	0	0
5	0	tcp	http	S0	0	0	0	0	0	0	0	0	0
7	2069	tcp	http	RSTR	72564	0	0	0	0	0	0	1	0
8	902	tcp	http	RSTR	57964	0	0	0	0	0	0	1	0
11	804	tcp	http	RSTR	21184	0	0	0	0	0	0	1	0
12	841	tcp	http	RSTR	60884	0	0	0	0	0	0	1	0

AttackName	TruePositive	FalsePositive
back	100	0
buffer_overflow	0	100
ftp_write	0	100
guess_passwd	0	100
imap	0	100
multihop	0	100
neptune	0	100
nmap	0	100
warezmaster	0	100
ipsweep	-153	253

AD Efficiency : 94.256993688147 Execution Time(in Seconds): 1217.282 Seconds

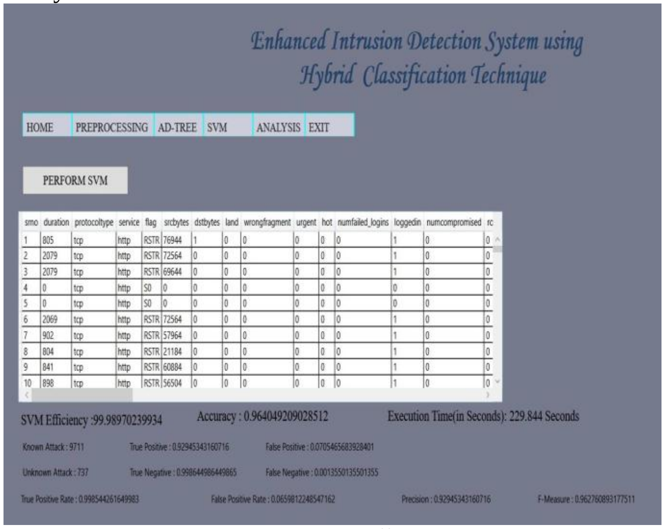
Known Attack : 12833 True Positive : 0.942569936881477 False Positive : 0.0574300631185226

Unknown Attack : 9711 True Negative : 1 False Negative : 0

True Positive Rate:1 False Positive Rate :0.0543109801031687 Precision :0.942569936881477 F-Measure :0.970436038348911

Figure 5.1: Adtree Efficiency

The result screen also shows efficiency both Adtree and SVM as well. The screen also shows the time taken to classify the data, true positive cases, false positive cases true negative cases and false negative cases. It also shows the count of the known and the unknown attacks classified. The precision and the f-measure of the algorithm are also shown. The SVM result screen also shows the overall accuracy of the system.


Enhanced Intrusion Detection System using
Hybrid Classification Technique
HOME
PREPROCESSING
AD-TREE
SVM
ANALYSIS
EXIT
PERFORM SVM

srno	duration	protocoltype	service	flag	srcbytes	dstbytes	land	wrongfragment	urgent	hot	numfailed_logins	loggedin	numcompromised	rc
1	805	tcp	http	RSTR	76944	1	0	0	0	0	0	1	0	0
2	2079	tcp	http	RSTR	72564	0	0	0	0	0	0	1	0	0
3	2079	tcp	http	RSTR	69644	0	0	0	0	0	0	1	0	0
4	0	tcp	http	S0	0	0	0	0	0	0	0	0	0	0
5	0	tcp	http	S0	0	0	0	0	0	0	0	0	0	0
6	2069	tcp	http	RSTR	72564	0	0	0	0	0	0	1	0	0
7	902	tcp	http	RSTR	57964	0	0	0	0	0	0	1	0	0
8	804	tcp	http	RSTR	21184	0	0	0	0	0	0	1	0	0
9	841	tcp	http	RSTR	60884	0	0	0	0	0	0	1	0	0
10	898	tcp	http	RSTR	56504	0	0	0	0	0	0	1	0	0

SVM Efficiency :99.98970239934 Accuracy : 0.964049209028512 Execution Time(in Seconds): 229.844 Seconds

Known Attack : 9711 True Positive : 0.92945343160716 False Positive : 0.0705465683928401

Unknown Attack : 737 True Negative : 0.998644986449865 False Negative : 0.0013550135501355

True Positive Rate : 0.998544261649983 False Positive Rate : 0.0659812248547162 Precision : 0.92945343160716 F-Measure : 0.962760893177511

Figure 5.2: SVM Efficiency

Confusion Matrix: A confusion matrix is a table that is often used to define the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

Confusion Matrix		Predicted Class	
		Negative	Positive
Actual Class	Negative	TN	FP
	Positive	FN	TP

Table 5.1: Confusion Matrix

Formulae:

Total no. of KNOWN attacks = A
Total no. of UNKNOWN attacks =B
Total no. of KNOWN attacks detected as UNKNOWN = C
Total no. of UNKNOWN attacks detected as KNOWN =D
Total no. of KNOWN attacks detected as KNOWN = E
Total no. of UNKNOWN attacks detected UNKNOWN =F

Formulae:

TP = E/(C+D)
TN = F/(F+D)
FP = C/(C+E)
FN = C/(F+D)
True Positive Rate: TP/(TP+FN)
False Positive Rate: 1-Specificity
Specificity= TN/ (TN+FP)
Precision = TP/ (TP+FP)
F-measure = 2 * (Precision * TPR) / (Precision + TPR)

VI. CONCLUSION

Considering the nature of the system which uses a hybrid mechanism to detect and classify the known (signature based) and unknown attacks efficiently and effectively, and thus reducing the false alarm rate. The system achieved efficiency of 94.256% for AdTree and 99.9897% for SVM. The overall classification accuracy achieved 0.964049.

VII. FUTURE SCOPE

Intrusion Detection System is favourable method for detecting the attacks, and as many attacks are real time, therefore a system with real time detection is essential. Additional data mining techniques should be checked to get more fruitful result in term of detection rate and accuracy, reducing the false alarm rate further. In future isolation of each type of attack can be done and also several more techniques may be applied to improve the accuracy further.

REFERENCES

[1] Rajesh Wankhede and Vikrant Chole (2016), Intrusion Detection System using Classification technique, International Journal of Computer Applications (0975 – 8887) Volume 139 – No.11, pp. 25-28.

[2] Gisung Kim and Seungmin Lee (2014), A Novel Hybrid Intrusion Detection Method Integrating Anomaly Detection With Misuse Detection, ELSEVIER, Expert Systems with Applications vol. 41 pp. 1690 – 1700.

[3] Zhi-Song Pan, Song-Can Chen, Gen-Bao Hu, DaoQiang Zhang, (2010), ―Hybrid Neural Network and C4.5 for Misuse Detection ‖, Proceedings of the second International conference

on Machine Learning and Cybernetics, November, pp. 2463 – 2467.

[4] H.F. Eid, A. Darwish A. H. Ella and A. Abraham, ―Principle components analysis and Support Vector Machine based Intrusion Detection System,‖ 2010, 10th International Conference on Intelligent Systems Design and Applications (ISDA), 2010.

[5] Tsang, C. H., Kwong, S., & Wang, H.,‖ Genetic-fuzzy rule reordering in mining approach and evaluation of feature selection techniques for anomaly intrusion detection‖, Pattern Recognition,40 (9), pp. 2373–2391, 2007. .

[6] Juan Wang, Qiren Yang, Dasen Ren, ―An intrusion detection algorithm based on decision tree technology‖, In the Proc. of IEEE Asia-Pacific Conference on Information Processing, 2009.

[7] M. Revathi, T.Ramesh - Network Intrusion Detection Sysytem using reduced dimentioality Indian Journal of Computer Science and Engineering (IJCSE), Vol. 2 No. 1, pp . 61-67.

[8] Yonav Freund et.al, ―The Alternating Decision Tree Algorithm‖, ICML '99 Proceedings of the Sixteenth International Conference on Machine Learning, pp 124-133.

[9] Tavallaee M, Bagheri E, Lu W, Ghorbani A. ―A detailed analysis of the KDD CUP 99 data set‖, IEEE Symposium on Computational intelligence for security and defense applications, 2009,pp 1-6.

[10] Hong Kuan Sok et.al, ―Using the ADTree for Feature Reduction through Knowledge Discovery‖ Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International ,pp1040 – 1044.

[11] Mrutyunjaya Panda and Manas Ranjan Patra, ―A Comparative Study Of Data Mining Algorithms For Network Intrusion Detection‖, First International Conference on Emerging Trends in Engineering and Technology, pp 504-507, IEEE, 2008.

[12] Shi-Jinn Horng and Ming-Yang Su (2011), ―Novel Intrusion Detection System Based On Hierarchical Clustering and Support Vector Machines‖, ELSEVIER, Expert Systems with Applications. pp. 38 306-313.