Journal of Computer Science 8 (7): 1041-1048, 2012
ISSN 1549-3636

Improving the Attack Detection Rate in Network Intrusion Detection using Adaboost Algorithm

[1]Natesan, P., [1]P. Balasubramanie and [2]G. Gowrison
[1]Department of Computer Science and Engineering,
Kongu Engineering College, Perundurai, Erode 638 052, Tamilnadu, India
[2]Department of Electronics and Communication Engineering,
Institute of Road and Transport Technology, Erode 638 316, Tamilnadu, India

Abstract: Problem statement: Nowadays, the Internet plays an important role in communication between people. To ensure a secure communication between two parties, we need a security system to detect the attacks very effectively. Network intrusion detection serves as a major system to work with other security system to protect the computer networks. **Approach:** In this article, an Adaboost algorithm for network intrusion detection system with single weak classifier is proposed. The classifiers such as Bayes Net, Naive Bayes and Decision tree are used as weak classifiers. A benchmark data set is used in these experiments to demonstrate that boosting algorithm can greatly improve the classification accuracy of weak classification algorithms. **Results:** Our approach achieves a higher detection rate with low false alarm rates and is scalable for large data sets, resulting in an effective intrusion detection system. **Conclusion:** The Naive Bayes and Decision Tree Classifiers have comparatively better performance as a weak classifier with Adaboost, it should be considered for the building of IDS.

Key words: Adaboost, weak classifier, detection rate, false alarm rate, computational complexity, Intrusion Detection System (IDS)

INTRODUCTION

The protection of the computer network by applying intrusion detection methodology becomes an important for the network administrator and it is one of the emerging areas in the research of the network security field. The main focus of network intrusion detection techniques is to capture, look into the various header parts and data portion of the packets and classify the attack packets from the normal packets. There are basically two types of intrusion detection systems namely misuse based detection and anomaly based detection. The anomaly based detection system first learns normal user activities and then alerts all user behaviors that deviate from the already learned activities (Barbara and Jajodia, 2002). The main feature of anomaly based detection is the capability of detecting the novel attacks which are different from the already learned attacks. The main drawback of anomaly based detection is that it erroneously classifies the normal user behaviors as attacks, which would result in a higher false positive rate. The misuse based detection uses the certain standard patterns of attacks to detect intrusions by representation of the same pattern of attacks (Freund and Schapire, 1997). Misuse based detection has higher network attack detection rate than anomaly based detection but it is failing to detect novel attacks.

Related work: Proposed a Bayesian classification approach for intrusion detection. It consists of monitoring the user activities inside the network and the use of a Bayesian classification procedure associated with unsupervised machine learning algorithm to evaluate the variation between the present and the already learned behavior. The reported results showed that there was an increase in attack detection rate. Zainal *et al.* (2009) demonstrated the ensemble of different learning algorithms by setting the proper weighting to the individual classifiers used in the classification model. They have also observed that there was an enhancement in the network attack detection and considerable drop on false alarms.

Recently, many researchers constructed hybrid Intrusion Detection System (IDS) to deal with the challenges faced by the intrusion detection system by integrating different machine learning methodologies. Horng *et al.* (2011) were developed a hybrid intelligent IDS by integrating a Hierarchical Clustering and Support

Corresponding Author: Natesan, P., Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode 638 052, Tamilnadu, India

Vector Machines (SVM). Xiang *et al.* (2008) designed IDS by integrating the supervised tree classifiers and unsupervised Bayesian clustering method to detect the network intrusions happening in the network.

Zhang and Zulkernine (2006) designed a novel structure of unsupervised anomaly Network IDS based on the outlier detection technique in the random forests approach. This approach reduced the time complexity and cost of memory to a large extent. The framework built by Sarasamma *et al*. (2005) based on the hierarchical method which improves the attack detection rate and reduces computational cost.

Giacinto *et al.* (2003) approached the intrusion detection problem in a different dimension. Their anomaly IDS was modularized where the protocols and services are modularized which improves the detection results. Gudahe *et al.* (2010) have demonstrated a new ensemble boosted decision tree for intrusion detection system.

Liu *et al.* (2010) have constructed a classifier by using a decision tree as its base learner. The ability of detecting the attacks of this construction was enhanced than SOM algorithms. Hu *et al*. (2008) have proposed an Adaboost based algorithm for network intrusion detection which used decision stump as its base learner. They constructed the decision rules for different categories of features such as categorical and continuous features and also they handled the over-fitting efficiently. The key difference between our proposed work and that of Hu *et al*. (2008) is that they have used decision stump as a weak learner, while we use Bayes Net (BN), Naive Bayes (NB) and Decision Tree (DT)) as weak learners. Hu *et al*. (2008) considered all the attacks as a single category, while our system groups all the attacks based on its characteristics into four categories such as DoS, Probe, R2L and U2R.

MATERIALS AND METHODS

Dataset analysis: Under the sponsorship of Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL), the MIT Lincoln laboratory has established a network and captured the packets of different attack types and distributed the data sets for the evaluation of researches in computer network intrusion detection systems. The KDDCup99 data set is a subset of the DARPA benchmark data set.

KDDCup99 training data set is about four giga bytes of compressed binary TCP dump data from seven weeks of network traffic, processed into about five million connections record each with about 100 bytes (KDDCup99, 1999; Tavallaee *et al*., 2009). The two weeks of test data have about two million sample records. Each KDDCup'99 training connection record contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type. Table 1 and 2 shows the number of samples for each attack category in the training and testing data sets respectively.

The rest of the study is organized as follows. We briefly present the overview of Adaboost algorithm, Bayesian Classifiers and Decision Tree algorithms. In the next part we discussed our proposed work. Experimental analyses are performed and is also given. Finally we conclude the study with suggestions for future work.

Overview of algorithms:

Adaboost algorithm: AdaBoost is an ensemble based machine learning algorithm, which can be combined with many other classification machine learning algorithms in order to improve its classification and attack detection performance. It calls a base learner for a specified amount of iterations in a loop. For each iteration, distribution of weights D_t is calculated and updated that indicates the importance of examples in the data set for the classification. On each iteration of the loop, the weights of each incorrectly classified samples are modified which is based on the distribution of the sample in the data set so that the new classifier will concentrate more on those samples classified as incorrect (Zan *et al.*, 2007; Sabhnani and Serpen, 2003). The pseudo code of Adaboost algorithm is given in Fig. 1.

Bayesian classifiers: Bayesian classification methodology is one of the technique used in the area of data mining for the purpose of classification of samples. Given the probability of distribution of samples in a data set, Bayes classifier can possibly accomplish the best optimal classification accuracy. Bayes Rule is constructed here to find the posterior probability from the prior probability and the likelihood of occurrence, because the latter two is generally easier to be calculated from the specified probability model.

Let X be a sample of a network connection consists of n features and C_i represent a class to be calculated (Khor *et al*., 2010b).

Table 1: Number of samples in the KDDCup'99 training set

	Attacks				
Normal	Probe	Dos	R2L	U2R	Total
97278	4107	391458	1126	52	494021

Table 2: Number of samples in the KDDCup'99 test set

	Attacks				
Normal	Probe	Dos	R2L	U2R	Total
60593	4166	229853	16189	228	311029

Input: Sequence of m training examples
Let the set of training sample data be $\{(x_1,y_1),\ldots,(x_i,y_i),\ldots,(x_m, y_m)\}$ with label $y_i \in \{+1,-1\}$, where x_i denotes i^{th} feature vector and m is the size of the dataset.
Let T be the number of iterations.
Initialize the weights $D_t(i) = 1/m$ for all i.
Repeat for t = 1, 2,…,T the following steps

(1) Call the weak classifier, and provide it with the instances of distribution D_t

(2) Calculate the error of each round of the hypothesis h_i :

$$X \rightarrow Y$$
$$\epsilon_i = Pr_{i\sim D_t}[h_t(x_i) \neq y_i] = \sum_{hi(xi)\neq yi}(D_t(i)) \quad (1)$$

If $\varepsilon_i > 0.5$, then set T = t-1 and abort loop.

(3) Calculate the reweight value by using the equation:

$$\beta_i = \epsilon_i / 1-\epsilon_i \quad (2)$$

(4) Update distribution D_t:

$$D_{t+1}(i) = \begin{cases} \beta i \frac{Dt(i)}{Zt} \\ \frac{Dt(i)}{Zt}(else) \end{cases} (ht(xi) = yi) \quad (3)$$

where, Z_t is a normalization constant.
Output: final hypothesis:

$$h_{fin}(x) = \arg\max_{y\epsilon Y} \sum_{ht(x)=y} \log\frac{1}{\beta} \quad (4)$$

Let us write the error ϵ_i of h_t as $\frac{1}{2}-\gamma_t$. Then γ_t shows how much better of weak learner than random guessing. Freund and Schapire (1997) have proven that the training error ϵ of the final hypothesis is at most:

$$\epsilon = \prod[2\sqrt{\epsilon t(1-\epsilon t)}]$$
$$= \prod_t \sqrt{1-4\gamma 2} \leq \exp(-2\sum_t \gamma 2) \quad (5)$$

From above equation (5), we can conclude that the training error of boosting algorithm drops exponentially fast.

Fig. 1: Adaboost algorithm

The predictable classification results in an observed network connection is decided by finding P (C_i|X), the probability of a class is equal to its likelihood P (X|C_i) times its probability prior to any experimental sample P (C_i), standardized by separating P (Xi) as in (6):

$$P(C_i \mid X) = P(X|C_i)\, P(C_i)/\, P(X) \quad (6)$$

Consider a Naive Bayesian Classification method with n nodes, X_i to X_n. The features and classes are represented by nodes, labeled with X_n and C respectively. An assumption is made in Naïve Bayes Classification where features are conditionally independent from each other. Since P (X) is constant for all classes, only P (X|Ci) needs to be maximized as in (7) (Khor *et al*., 2010a). Hence:

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1|C_i)\, x\, P(x_2 \mid C_i)\, x.\, x\, P(x_n \mid C_i) \quad (7)$$

Naïve Bayes classifier is an accepted classifier appearing in its competitive performance in many research domains such as medical, business and its simplicity in computation that allows researchers to save a lot of computational costs (Khor *et al*., 2010b; Han *et al*., 2005; Friedman *et al*., 1997; Gupta *et al*., 2010; Kayacik *et al*., 2003).

A Bayes Net employs a graphical model to describe the relationship of features. The structure of the graphical model and also a Conditional Probability Table (CPT) of a BayesNet classifier could be built based on a training set.

The graphical model state a factorization of the joint probability distributions, where a value of a node is conditioned on its parent nodes which is given in (8). Hence:

$$P(x_1,x_2,\ldots,x_n) = \prod_{i=1}^{n} (x_i|\text{Parents}(C_i)) \quad (8)$$

A Bayes Net can also be built manually by integrating knowledge of a domain expert. The built process is repetitive process which involves model verification and model revision (Khor *et al*., 2010b).

Decision tree construction: The decision tree is frequently used machine learning technique for constructing classification system. In the decision tree construction, each internal node represents a test for a feature and each branch denotes the conclusion of the test. The leaf node of the tree indicates classes or the division of classes (Xiang *et al*., 2008). The pseudo code for decision tree construction is in Fig. 2.

Proposed work: As per the requirements of a Network Intrusion Detection system, the construction of our proposed system consists of four components of Adaboost algorithm as shown in Fig. 3. Feature extraction, Instance labeling, devise of weak classifiers and the building of the strong classifier.

Process 1-Feature extraction: For each network connection in the data set, the following three key groups of features for detecting intrusions are extracted.

Basic features: This group summarize all the features that can be extracted from a TCP/IP connection. Some of the basic features in the KDDCup99 data sets are protocol_type, service, src_bytes and dst_bytes.

Content features: These features are purely based on the contents in the data portion of the data packet.

Algorithm:

Step 1: Let S be the set of instances, select one instance at random from S and announce that it belongs to some class C_i. P_i is the probability that an arbitrary sample belongs to class Cj, which is estimated by:

$$P_i = \text{freq}(C_i, S)/|S| \quad (9)$$

where, $|S| \in S$. Here, $-\log_2 P_i$ bits of information is conveyed.

Step 2: Let P be the probability distribution, $P=\{P_1, P_2, \ldots, P_n\}$ then the information conveyed by this distribution is defined as:

$$\text{Info}(P) = \sum_{i=1}^{n} -P_i \log_2 P_i \quad (10)$$

Step 3: The weighted average of Info (T_i) is calculated as given below.

$$\text{Info}(X, T) = \sum_{i=1}^{m} -\frac{|T_i|}{|T|} \text{ x Info}(T_i) \quad (11)$$

Here T is a set of samples partitioned into sets $T_1, T_2, \ldots T_m$ based on the non-categorical attribute X.
The Information gain, Gain(X,T) is calculated as:

$$\text{Gain}(X,T) = \text{Info}(T) - \text{Info}(X,T) \quad (12)$$

The attribute with the highest information gain is chosen as the test attribute for the current node. This process continues until all the attributes are compared or when all the instances are all of the same class or there are no remaining attributes on which the instances may be further partitioned.

Fig. 2: Decision tree construction

Traffic features: This group comprises features that are computed with respect to a two-second time window and it is divided into two groups: same host features and same service features. The same host featured inspect only the connections in the past 2 sec that have the same destination host as the current connection. The same service featured inspect only the connections in the past 2 sec that have the same service as the current connection. Some of the traffic features are counted, rerror_rate, rerror_rate and srv_serror_rate.

Process 2-instance labeling: After extracting KDDCup'99 features from each record, the instances are labeled based on the characteristics of traffic as Normal, Dos, Probe, R2L and U2R.

Process 3-selection of weak classifiers: The various weak classifiers identified to use in our proposed system are Naïve Bayes, Bayes Net and Decision Tree. We have used these weak classifiers along with the boosting algorithm to improve the classification accuracy.

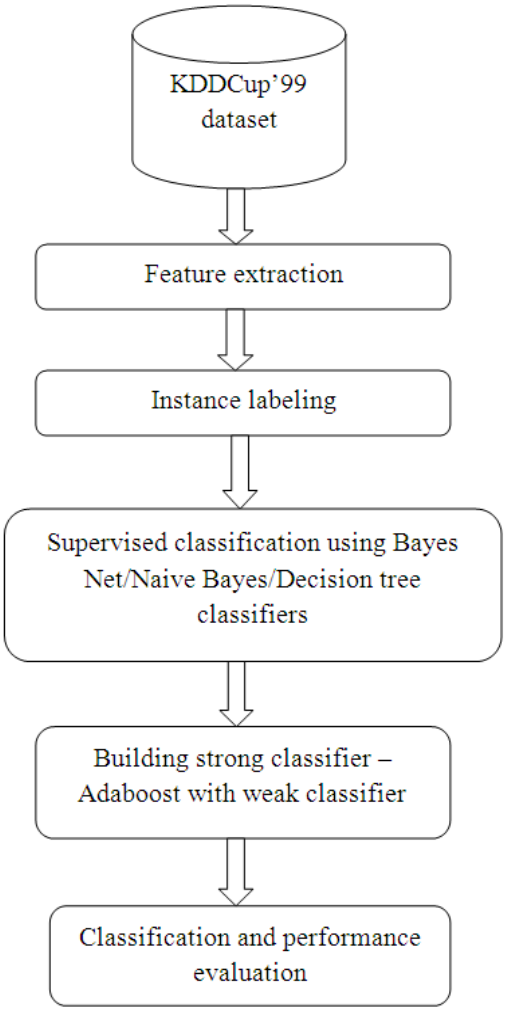

Fig. 3: Framework of our Intrusion detection model

Process 4-building of strong classifier: A strong classifier is constructed by using a mechanism of combining weak classifier and boosting algorithm. The strong classifier results higher attack detection rate than single weak classifier. The Pseudo code of our proposed IDS is shown in Fig. 4.

Experimental analysis: The main focus of our work was to improve the network attack detection rate and to reduce the false alarm rate to a minimum level. The experiment was conducted using the Bayes Net, Naïve Bayes and Decision Tree weak classifiers. Weka 3.6 is a java language based open source data mining software, which comprises a group of machine learning packages for classification of samples, is chosen to implement our algorithm.

Input : Instances in KDDCup'99 dataset

Let the given training sample data be $\{(x_1, y_1), (x2, y2), (x3, y3), \ldots, (x_i, y_i), \ldots, (x_m, y_m)\}$ with label $y_i \in \{Normal, Dos, Probe, R2L, U2R\}$ where x_i denotes i^{th} feature vector and m is the size of the dataset. Initialize the weights $D_t(i) = 1/m$ for all i.

Repeat for t = 1, 2,…,T the steps (1) to (3)

(1) Call the weak classifier, and provide it with the instances of distribution for each category of attacks (D_t)

(2) Calculate the error rate for each category of attacks on each round of the hypothesis:

$$h_i : X \rightarrow Y$$

$$\epsilon_{at} = Pr_{i \sim Dt}[h_t(x_i) \neq y_i] = \sum_{hi(xi \neq yi}(D_t(i)) \quad (13)$$

If $\varepsilon_{at} > 0.5$, then set T = t-1 and abort loop. Here ε_{at} is the error rate of each category of attacks at the t^{th} iteration.

(3) Calculate the reweight value by using the equation:

$$\beta_i = \epsilon_{at} / 1\text{-}\epsilon_{at} \quad (14)$$

Update distribution D_t as given in (2)

(4) Test the model constructed in above using the KDDCup'99 test dataset and evaluate the performance of the system.

Output: Classified instances and false alarms

Fig. 4: Pseudo code of proposed work

RESULTS AND DISCUSSION

In machine learning and data mining algorithms, many different measures are used to evaluate the classification models (Tan *et al*., 2006).

True Positive (TP): Situation in which a signature is fired properly when an attack is detected and an alarm is generated.

False Positive (FP): Situation in which normal traffic causes the signature to raise an alarm.

True Negative (TN): Situation in which normal traffic does not cause the signature to raise an alarm.

False Negative (FN): Situation in which a signature is not fired when an attack is detected.

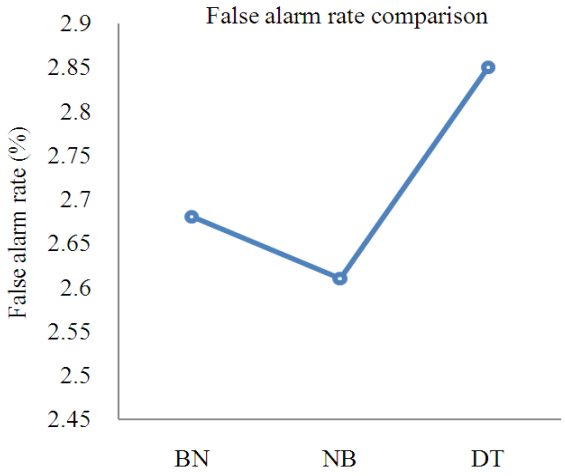

Fig. 5: The false alarm rate of different weak classifiers with Adaboost

Attack Detection Rate (ADR): It is the ratio between the total numbers of attack connections detected by our proposed model to the total number of attacks currently available in the data set.

Attack Detection Rate (ADR) Eq. 15:

$$\frac{\text{Total detected attacks}}{\text{Total attacks}} * 100 \quad (15)$$

False Alarm Rate (FAR): It is the ratio between the total numbers of misclassified instances of the total number of normal connections present in the data set.

False Alarm Rate Eq. 16:

$$\frac{\text{Total misclassfied instances}}{\text{Total normal instances}} * 100 \quad (16)$$

Comparison of performance of weak classifiers:
Detection rate comparison: The detection rates (15) of the various attack categories by using the three weak classifiers in the boosting process are shown in Table 3. It can be noticed that, the detection rate of Dos attack increases to 97.3% and the detection rate of Probe attack increases to 91.4% when the weak classifier decision tree is combined with Adaboost. It can also be seen that the Naive Bayes weak classifier with Adaboost gives the better detection rate in the case of U2R and R2L attack categories.

False Alarm rate comparison: The false alarm rate (16) of Naïve Bayes weak classifier with Adaboost decreases to 2.61%, but it shows an increase in the case of Decision Tree as a weak classifier with the Adaboost algorithm as shown in Fig. 5.

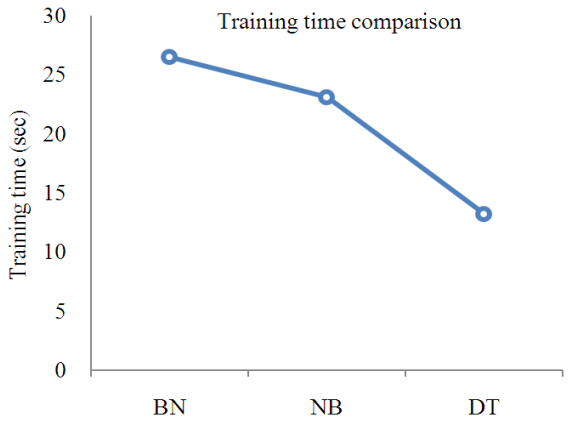

Fig. 6: The training time comparison of different weak classifiers with Adaboost

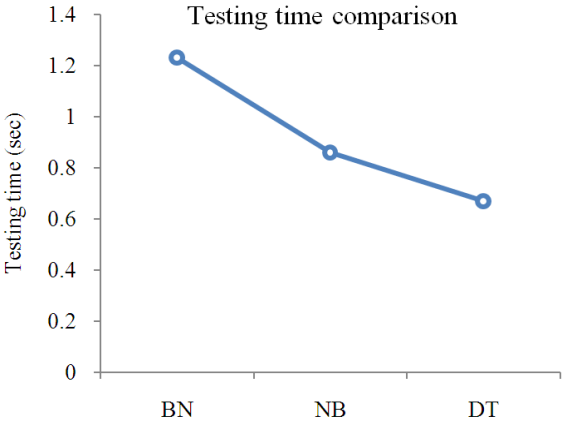

Fig. 7: The testing time comparison of different weak classifiers with Adaboost

Table 3: The attack detection rate of different weak classifiers

	(%) of detection rate		
Attack category	Adaboost with Bayes Net (AB-BN)	Adaboost with Naive Bayes	Adaboost with Decision tree
Dos	95.8	96.7	97.3
Probe	88.5	89.6	91.4
R2L	14.7	19.5	18.4
U2R	49.3	51.2	50.4

Computational time comparison: The training time and the testing time of various weak classifiers with Adaboost are shown in Fig. 6 and 7 respectively. The Naive Bayes and Decision Tree algorithms took more time than Bayes Net Algorithm. It shows a decrease in training time and response time in the case of Naïve Bayes and Decision Tree as a weak classifier with Adaboost algorithm.

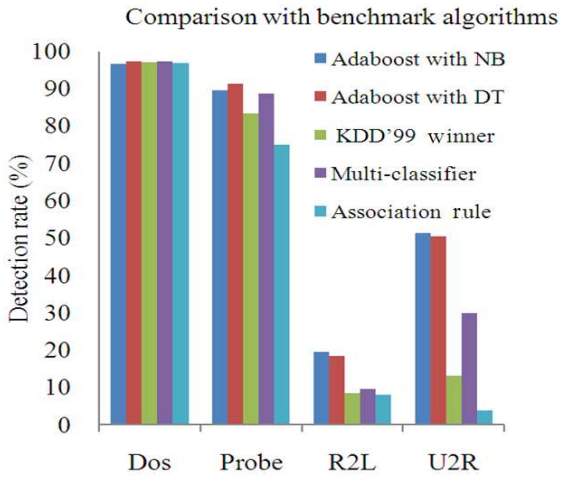

Fig. 8: Comparison with other Algorithms

Table 4: Comparison with other algorithms of attack detection rate

	% of Detection rate			
Name of the method	Dos	Probe	R2L	U2R
Adaboost with NB	96.7	89.6	19.5	51.2
Adaboost with DT	97.3	91.4	18.4	50.4
KDD'99 Winner (Pfahringer, 2000)	97.1	83.3	8.40	13.2
Multi-classifier (Xiang *et al*., 2008)	97.3	88.7	9.60	29.8
Association Rule (Xuren *et al*., 2006)	96.8	74.9	7.90	3.8

Based on the attack detection rates and false alarm rates, the weak classifiers with Adaboost seem to have comparable performances. Decision tree was able to give a high detection rate with low computational time in the case of Dos and Probe attack categories and the Naïve Bayes with Adaboost gave a better detection rate in the case of R2L and U2R attack categories as compared to other weak classifier Bayes Net.

Comparisons of detection rate with different algorithms: The network attack detection rate and false alarm rate of our work are compared with existing work, which are tested on the benchmark KDDCup'99 data set shown in Table 4. Their performances were comparable but the Naïve Bayes classifier with Adaboost and Decision Tree classifier with Adaboost performed well. Since the Naïve Bayes and Decision Tree classifiers have reasonably better performance as a weak classifier with Adaboost, it should be considered for the building of intrusion detection system.

From the Fig. 8, we observe that the Adaboost with Naïve Bayes and Adaboost with Decision Tree perform considerably superior than the earlier reported results including the winner of the KDD'99 cup and Muli-

classifier method. The Adaboost with Decision tree have very high network attack detection of 97.3 percent for Dos and 91.4 percent detection for Probe and the Adaboost with Decision tree have very high network attack detection of 19.5 percent for R2L and 51.2 percent detection for U2R.

CONCLUSION

Conclusion and future work: In this work we have combined the adaboost algorithm with various weak classifiers. The weak classifiers such as Bayes Net, Naive Bayes and Decision tree are used with the Adaboost algorithm to improve the classification accuracy. In this work, we have concentrated on the two problems such as attack detection rate and false alarm rate for building healthy and extensible intrusion detection system. It is important to have a very low false alarm rate for an efficient intrusion detection system. The experiment results illustrate that the Naïve Bayes with Adaboost and Decision Tree with Adaboost algorithm have a very low false alarm rate with a higher attack detection rate. We have focused mainly to obtain better classification through the time and computational complexities are theoretically higher. But practically the time and computational complexities are reduced by processing speed of the computing device.

The areas for future research include the considering the other classifiers to search for the opportunity of improving the classification accuracy and to combine two weak classifiers linearly with Adaboost algorithm. The Adaboost algorithm can be further improved in order to detect the attacks more effectively.

REFERENCES

Barbara, D. and S. Jajodia, 2002. Applications of Data Mining in Computer Security. 1st Edn., Springer, Boston, Mass., ISBN-10: 1402070543, pp: 252.

Freund, Y. and R.E. Schapire, 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci., 55: 119-139. DOI: 10.1006/jcss.1997.1504

Friedman, N., D. Geiger and M. Goldsmidt, 1997. Bayesian network classifiers. Mach. Learn., 29: 131-163. DOI: 10.1023/A:1007465528199

Giacinto, G., F. Roli and L. Didaci, 2003. Fusion of multiple classifiers for intrusion detection in computer networks. Patt. Recog. Lett., 24: 1795-1803. DOI: 10.1016/S0167-8655(03)00004-7

Gudahe, M., P. Prasad and K. Wankhade, 2010. A new data mining based network intrusion detection model. Proceedings of the International Conference on Computer and Communication Technology, Sept. 17-19, IEEE Xplore Press, Allahabad, Uttar Pradesh, pp: 731-735. DOI: 10.1109/ICCCT.2010.5640375

Gupta, K.K., B. Nath and R. Kotagiri, 2010. Layered Approach using conditional random fields for intrusion detection. IEEE Trans. Dependable Secure Comput., 7: 35-49. DOI: 10.1109/TDSC.2008.20

Han, J., M. Kamber and J. Pei, 2005. Data Mining: Concepts and Techniques. 2nd Edn., Morgan Kaufmann, ISBN-10: 1558609016, pp: 800.

Horng, S.J., M.Y. Su, Y.H. Chen, T.W. Kao and R.J. Chen *et al*., 2011. A Novel intrusion detection system based on hierarchical clustering and support vector machines. J. Exp. Syst. Appli., 38: 306-313. DOI: 10.1016/j.eswa.2010.06.066

Hu, W., W. Hu and S. Maybank, 2008. AdaBoost-based algorithm for network intrusion detection. IEEE Trans. Sys., Man Cybernetics, 38: 577-583. DOI: 10.1109/TSMCB.2007.914695

Kayacik, H.G., A.N. Zincir-Heywood and M.I. Heywood, 2003. With the capability of an SOM based intrusion detection system. Proceedings of the International Joint Conference on Neural Networks, Jul. 20-24, IEEE Xplore Press, pp: 1808-1813. DOI: 10.1109/IJCNN.2003.1223682

Khor, K.C., C.Y. Ting and S. Phon-Amnuaisuk, 2010a. A cascaded classifier approach for improving detection rates on rare attack categories in network intrusion detection. Applied Intell., 36: 320-329. DOI: 10.1007/s10489-010-0263-y

Khor, K.C., C.Y. Ting and S. Phon-Amnuaisuk, 2010b. Comparing single and multiple Bayesian classifiers approach for network intrusion detection. Proceedings of the 2nd International Conference on Computer Engineering and Applications, Mar. 19-21, IEEE Xplore Press, Bali Island, pp: 325-329. DOI: 10.1109/ICCEA.2010.214

Liu, Y., N. Li, L. Shi and F. Li, 2010. An intrusion detection method based on decision tree. Proceedings of the International Conference on E-Health Networking, Digital Ecosystems and Technologies, Apr. 17-18, IEEE Xplore Press, Shenzhen, pp: 232-235. DOI: 10.1109/EDT.2010.5496597

Pfahringer, B., 2000. Winning the KDD99 classification cup: Bagged boosting. SIGKDD Exp. Newsletter., 1: 65-66. DOI: 10.1145/846183.846200

Sabhnani, M.R. and G. Serpen, 2003. Application of machine learning algorithms to KDD intrusion detection data set within misuse detection context. Proceedings of the International Conference on Machine Learning: Models, Technologies and Applications, (MLMTA' 03), pp: 209-215.

Sarasamma, S.T., Q.A. Zhu and J. Huff, 2005. Hierarchical kohonenen net for anomaly detection in network security. IEEE Trans. Syst. Man Cybernetics, 35: 302-312. DOI: 10.1109/TSMCB.2005.843274

Tan, P.N., M. Steinbach and V. Kumar, 2006. Introduction to Data Mining. 1st Edn., Pearson Addison Wesley, London, ISBN-10: 0321420527, pp: 769.

Tavallaee, M., E. Bagheri, W. Lu and A.A. Ghorbani, 2009. A detailed analysis of the KDD CUP 99 data set. Proceedings of the 2nd IEEE Symposium on Computational Intelligence for Security and Defense Applications, Jul. 10-10, NRC, Canada, pp: 1-7.

Xiang, C., P.C. Yong and L.S. Meng, 2008. Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. J. Patt. Recogn. Lett., 29: 918-924. DOI: 10.1016/j.patrec.2008.01.008

Xuren, W., H. Famei and X. Rongsheng, 2006. Modeling intrusion detection system by discovering association role in the rough set theory framework. Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, Nov. 28-Dec. 1, IEEE Xplore Press, Sydney, NSW, pp: 24-24. DOI: 10.1109/CIMCA.2006.148

Zainal, A., M.A. Maarof and S.M. Shasuddin, 2009. Ensemble classifiers for network intrusion detection system. J. Inform. Assure. Secu., 4: 217-225.

Zan, X., J. Han, J. Zhang, Q. Zheng and C. Han, 2007. A Boosting approach for intrusion detection. J. Elect., 24: 369-373. DOI: 10.1007/s11767-005-0201-z

Zhang, J. and M. Zulkernine, 2006. Anomaly based network intrusion detection with unsupervised outlier detection. Proceedings of the IEEE International Conference on Communication, Jun. 11-15, IEEE Xplore Press, Istanbul, pp: 2388-2393. DOI: 10.1109/ICC.2006.255127