Journal of Computer Applications ISSN: 0974 – 1925,
Volume-5, Issue EICA2012-1, February 10, 2012

A New Data Mining Based Network Intrusion Detection Model

Manikandan R
Asst.Professor
Arunai College of Engineering
Tiruvannamalai
E-mail: lathakec.91@gmail.com

Oviya P
Department of IT
Arunai College of Engineering
Tiruvannamalai
E-mail: oviyakec@gmail.com

Hemalatha C
Department of IT
Arunai College of Engineering
Tiruvannamalai
E-mail: maniarunai@gmail.com

Abstract - As information systems are more widely used in the field of internet nowadays the need for secure networks is tremendously increased. New intelligent Intrusion Detection Systems (IDSs) based on sophisticated algorithms rather than current signature base detections are in demand. Due to emerging new attack methods there is often the need to update an installed Intrusion Detection System (IDS). Many of the current Intrusion Detection Systems are constructed by manual coding of expert knowledge, so changes to them are expensive and slow. In data mining based intrusion detection system we should have thorough knowledge about the particular domain in relation to intrusion detection so as to efficiently extract relative rule from huge amounts of records. This paper proposes a new ensemble boosted decision tree approach for intrusion detection system.

Keywords: *Boosted decision tree, Data mining, ensemble approach, Network intrusion detection system.*

I. INTRODUCTION

Being widely used and quickly developed in recent years, network technologies have provided us with new life and shopping experiences, particularly in the fields of e-business, elearning and e-money. But along with network development, there has come a huge increase in network crime. It not only greatly affects our everyday life, which relies heavily on networks and Internet technologies, but also damages computer systems that serve our daily activities, including business, learning, entertainment and so on. Besides of this internal hacking is difficult to detect because firewalls and Intrusion Detection Systems usually only defend against outside attacks.

Intrusion Detection System (IDS) [3] is an important detection used as a countermeasure to preserve data integrity and system availability from attacks. Intrusion Detection Systems (IDS) is a combination of software and hardware that attempts to perform intrusion detection. Intrusion detection is a process of gathering intrusion related knowledge occurring in the process of monitoring the events and analyzing them for sign or intrusion. It raises the alarm when a possible intrusion occurs in the system. The network data source of intrusion detection consists of large amount of textual information, which is difficult to comprehend

and analyze. Many IDS can be described with three fundamental functional components – Information Source, Analysis, and Response. Different sources of information and events based on information are gathered to decide whether intrusion has taken place. This information is gathered at various levels like system, host, application, etc. Based on analysis of this data, we can detect the intrusion based on two common practices – Misuse detection and Anomaly detection. Misuse detection is based on extensive knowledge of patterns associated with known attacks provided by human experts. Pattern matching, data mining, and state transition analysis are some of the approaches for Misuse detection. Anomaly detection is based on profiles that represent normal behavior of users, hosts, networks, and detecting attacks of significant deviation from these profiles. Statistical methods, expert system are some of the methods for intrusion detection based on Anomaly detection.

The main motivation behind using intrusion detection in data mining [5, 10, 12, 13, 15, 18] is automation. Pattern of the normal behavior and pattern of the intrusion can be computed using data mining. To apply data mining techniques in intrusion detection, first, the collected monitoring data needs to be preprocessed and converted to the format suitable for mining processing. Next, the reformatted data will be used to develop a clustering or classification model. The classification model can be rule-based, decision-tree based, association-rule based, Bayesian-network based, or neural network based. Intrusion Detection mechanism based on IDS are not only automated but also provides for a significantly elevated accuracy and efficiency. Unlike manual techniques, Data Mining ensures that no intrusion will be missed while checking real time records on the network. Credibility is important in every system. IDS are now becoming important part of our security system, and its credibility also adds value to the whole system. Data mining techniques can be applied to gain insightful knowledge of intrusion prevention mechanisms. They can help detect new vulnerabilities and intrusions, discover previous unknown patterns of attacker behaviors, and provide decision support for intrusion management.

The proposed paper organized as, Section 2 explains about data mining. Section 3 introduces boosted decision tree. Experiment and result included in Section 4 with concluding conclusion in section 5.

II. DATA MINING

Data mining (DM), also called Knowledge-Discovery and Data Mining, is one of the hot topic in the field of knowledge extraction from database. Data mining is used to automatically learn patterns from large quantities of data. Mining can efficiently discover useful and interesting rules from large collection of data. It is a fairly recent topic in computer science but utilizes many older computational techniques from statistics, information retrieval, machine learning and pattern recognition.

Data mining is disciplines works to finds the major relations between collections of data and enables to discover a new and anomalies behavior. Data mining based intrusion detection techniques generally fall into one of two categories; misuse detection and anomaly detection. In misuse detection, each instance in a data set is labeled as ‘normal’ or ‘intrusion’ and a learning algorithm is trained over the labeled data. These techniques are able to automatically retrain intrusion detection models on different input data that include new types of attacks, as long as they have been labeled appropriately. Data mining are used

in different field such as marketing, financial affairs and business organizations in general and proof it is success. The main approaches of data mining that are used including classification which maps a data item into one of several predefined categories. This approach normally output "classifiers" has ability to classify new data in the future, for example, in the form of decision trees or rules. An ideal application in intrusion detection will be together sufficient "normal" and "abnormal" audit data for a user or a program. The second important approach is clustering which maps data items into groups according to similarity or distance between them.

Anomaly detection techniques thus identify new types of intrusions as deviations from normal usage [7, 8]. In statistics based outlier detection techniques [4] the data points are modeled using a stochastic distribution and points are determined to be outliers depending upon their relationship with this model. However, with increasing dimensionality, it becomes increasingly difficult and in-accurate to estimate the multidimensional distributions of the data points [1]. However, recent outlier detection algorithms that we utilize in this study are based on computing the full dimensional distances of the points from one another [9, 16] as well as on computing the densities of local neighborhoods [6].

Classifier construction is another important research challenge to build efficient IDS. Nowadays, many data mining algorithms have become very popular for classifying intrusion detection datasets such as decision tree, naïve Bayesian classifier, neural network, genetic algorithm, and support vector machine etc. However, the classification accuracy of most existing data mining algorithms needs to be improved, because it is very difficult to detect several new attacks, as the attackers are continuously changing their attack patterns. Anomaly network intrusion detection models are now using to detect new attacks but the false positives are usually very high. The performance of an intrusion detection model depends on its detection rates (DR) and false positives (FP). Ensemble approaches [14, 17] have the advantage that they can be made to adopt the changes in the stream more accurately than single model techniques. Several ensemble approaches have been proposed for classification of evolving data streams

Ensemble classification technique is advantageous over single classification method. It is combination of several base models and it is used for continuous learning. Ensemble classifier has better accuracy over single classification technique. Bagging and boosting are two of the most well-known ensemble learning methods due to their theoretical performance guarantees and strong experimental results. Boosting has attracted much attention in the machine learning community as well as in statistics mainly because of its excellent performance and computational attractiveness for large datasets.

III. OUR APPROACH

This proposed model uses boosted decision tree i.e. hoeffding tree classification techniques to increase performance of the intrusion detection system.

Boosted Decision Tree- The underlying idea of boosting is to combine simple rules to form an ensemble such that the performance of the single ensemble member is improved, i.e. boosted. Let h1, h2, …. hN be a set of hypotheses and consider the composite ensemble hypothesis,

$$f(x) = \sum_{n=1}^{N} \alpha_n \, h_{n(x)} \qquad (1)$$

Here α_n denotes the coefficient with which the ensemble member h_n is combined; both α_n and the learner or hypothesis h_n are to be learned within the boosting procedure.

The boosting algorithm initiates by giving all data training tuples the same weight w0. After a classifier is built, the weight of each tuple is changed according to the classification given by that classifier. Then, a second classifier is built using the reweighted training tuple. The final classification of a intrusion detection is a weighted average of the individual classifications over all classifiers. There are several methods to update the weights and combine the individual classifiers. After the kth decision tree is built, the total misclassification error ε_k of the tree, defined as the sum of the weights of misclassified tuples over the sum of the weights of all tuples, is calculated:

$$\varepsilon_{k=} \sum_{i(miscl)} w_i^k / \sum_i w_i^k \qquad (2)$$

where i loops over all instances in the data sample. Then, the weights of misclassified tuples are increased

$$w^{k+1} \frac{1-\varepsilon_k}{\varepsilon_k} w_i^k \qquad (3)$$

Finally, the new weights are renormalized as,

$$w_i^{k+1} \rightarrow w_i^{k+1} / \sum_i w_i^{k+1} \qquad (4)$$

and the tree k+1 is constructed. Note that, as the algorithm progresses, the predominance of hard-to-classify instances in the training set is increased. The final classification of tuple I is a weighted sum of the classifications over the individual trees. Furthermore, trees with lower misclassification errors "k are given more weight when the final classification is computed.

In decision tree i.e. hoeffding tree, each node contains a test on an attribute, each branch from a node corresponds to a possible outcome of the test and each leaf contains a class prediction. A decision tree is learned by recursively replacing leaves by test nodes, starting at the root. The attribute to test at a node is chosen by comparing all the available attributes and choosing the best one.

For classifying examples in the dataset, the prior and conditional probabilities generated from the dataset are used to make the prediction. This is done by combining the effects of the different attributes values from the example. Suppose the example ej has independent attribute values { ai1, ai2,…, aip}, we know P(aik | cj), for each class cj and attribute ajk and then estimate P(ej|cj) by

$$P(e_i|c_j) = P(c_j) \prod_{k=1 \to p} P(a_{ij}|c_j) \qquad (5)$$

To classify an example in the dataset, the algorithm estimates the likelihood that ei is in each class. The probability that ei is in a class is the product of the conditional probabilities for each attribute value with prior probability for that class. The posterior probability P(cj | ei) is then found for each class and the example classifies with the highest posterior probability for that example. The algorithm will continue this process until all the examples of sub-datasets or sub-subdatasets are correctly classified. When the algorithm correctly classifies all the examples of all sub or sub-sub datasets, then the algorithm terminates and the prior and conditional probabilities for each sub or sub-sub-datasets are preserved for future classification of unseen examples.

In this proposed scheme boosting method improves ensemble performance by using adaptive window and adaptive size hoeffding tree as base learner. Because of this algorithm woks faster and increases performance. It uses dynamic sample weight assignment technique. In this algorithm adaptive sliding window is parameter and assumption free in the sense that it automatically detects and adapts to the current rate of change. Its only parameter is a confidence bound _. Window is not maintained explicitly but compressed using a variant of the exponential histogram technique. It keeps the window of length W using only O (log W) memory & O (log W) processing time per item, rather than the O (W) one expects from a naïve implementation. It is used as *change detector* since it shrinks window if and only if there has been significant change in recent examples, and *estimator* for the current average of the sequence it is reading since, with high probability, older parts of the window with a significantly different average are automatically dropped.

IV. EXPERIMENT AND RESULT

The proposed boosted decision trees algorithm is tested on KDDCup'99 dataset [11] and compared to that of a Naïve Bayes, kNN, eClass0 [2], eClass1 [2] and the Winner (KDDCup'99).

A. Evaluation of Anomaly Detection

There are generally two types of attacks in network intrusion detection: the attacks that involve single connections and the attacks that involve multiple connections (bursts of connections). The standard metrics in Table 1 treat all types of attacks similarly thus failing to provide sufficiently generic and systematic evaluation for the attacks that involve many network connections.

Table I. Confusion Matrix for Evaluation of Intrusion Detection

Confusion Matrix		Predicted Class	
		Normal	Intrusion / Attack
Actual Class	Normal	True Negative	False Positive
	Intrusion / Attack	False Negative	Correctly Detected

Interleaved Test-Then-Train - In this method each individual example can be used to test the model before it is used for training and from this the accuracy can be incrementally updated. The intension behind using this method is that, the model is always being tested on examples it has not seen. The advantage over holdout method being that holdout set is not needed for testing and ensures a smooth plot of accuracy over time as each individual example will become increasingly less significant to the overall average.

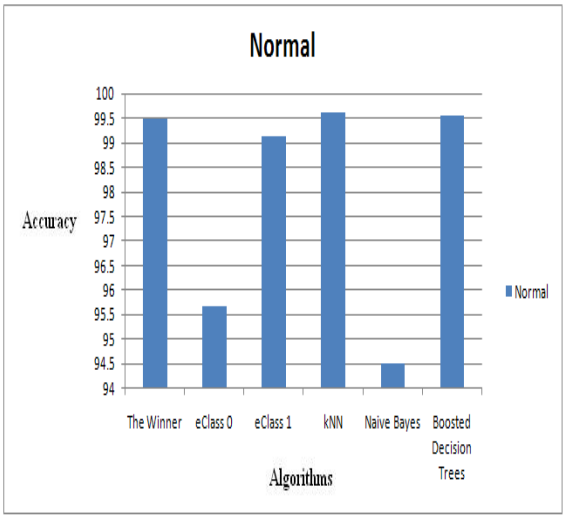
B. Evaluation on KDDCup'99 Data Set

The experiment is carried out on a intrusion detection real data stream which has been used in the Knowledge Discovery and Data Mining (KDD) 1999 Cup competition. In KDD99 dataset the input data flow contains the details of the network connections, such as protocol type, connection duration, login type etc. Each data sample in KDD99 dataset represents attribute value of a class in the network data flow, and each class is labeled either as normal or as an attack with exactly one specific attack type. In total, 41 features have been used in KDD99 dataset and each connection can be categorized into five main classes as one normal class and four main intrusion classes as DOS, U2R, R2L and Probe. There are 22 different types of attacks that are grouped into the four main types of attacks DOS, U2R, R2L and Probe tabulated in Table 2.

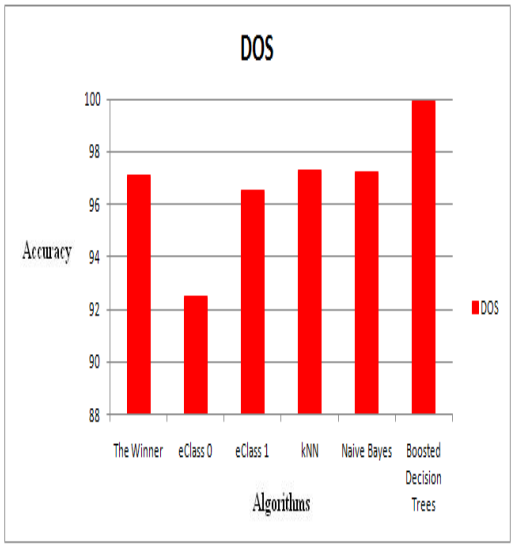
The experimental setting is for the KDD99 Cup, taking 10% of the whole real raw data stream (494021 data samples) and 12 features are selected as per proposed algorithm.Figures 1(a) - 1(e) show graphical comparison of boosted decision trees algorithm with the Winner (KDDCup'99), eClass0, eClass1, kNN, C4.5 and Naïve Bayes in terms of accuracy or detection rate.

Table II. Different Types of Attacks

Main Attack Classes	22 Different Attack Types
DOS- Denial Of Service	Back ,land, Neptune, pod, smurf, teardrop.
U2R-User to Root	Buffer_overflow, loadmodule, perl, rootkit.
R2L-Remote to User	ftp_write, guess_password, imap, multihop, phf, spy.
Probe	Ipsweep ,nmap, portsweep, satan.



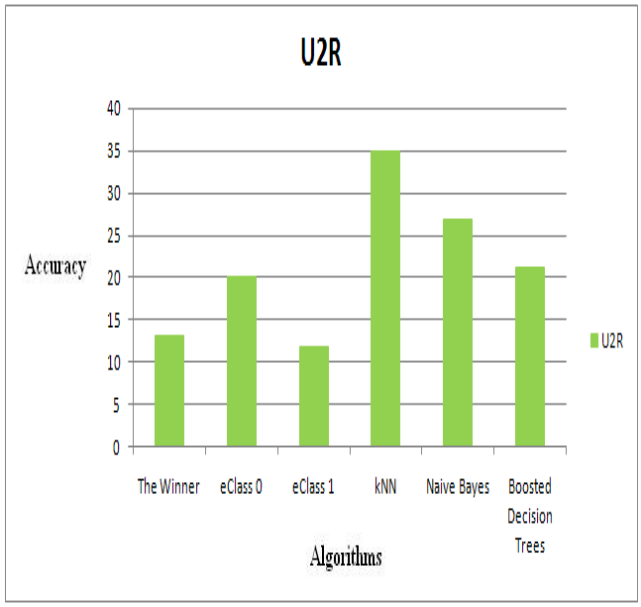
(a) Normal with 41 features



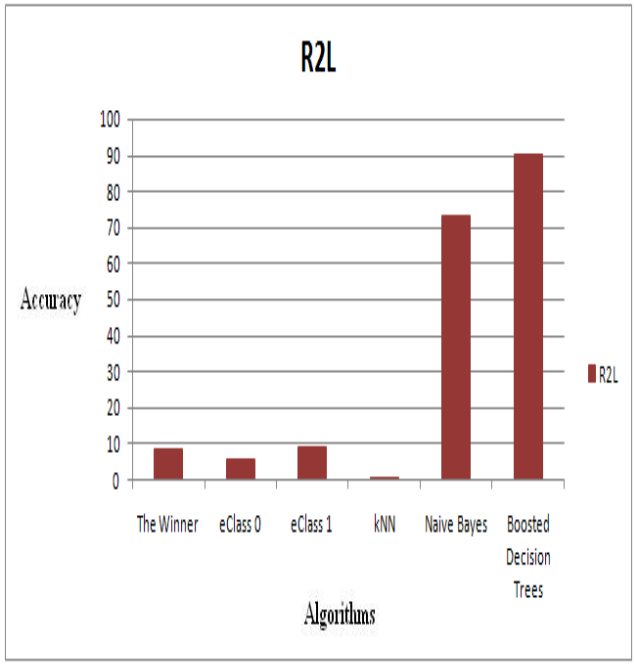
(b) DOS attack with 41 features

V. CONCLUSION

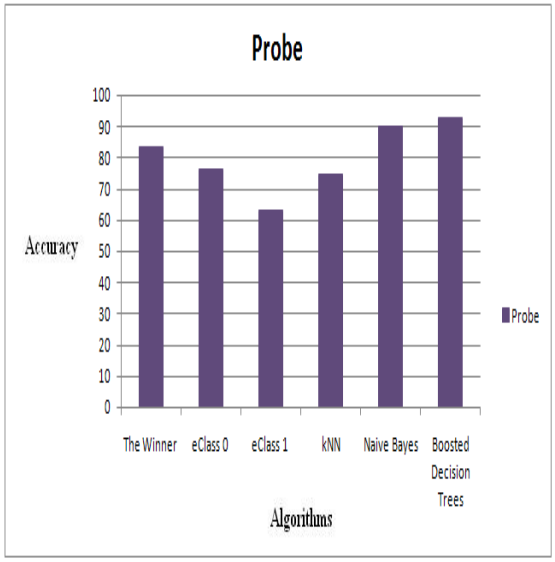
This paper introduced a network intrusion detection model using boosted decision trees: a learning technique that allows combining several decision trees to form a classifier which is obtained from a weighted majority vote of the classifications given by individual trees. The generalization accuracy of boosted decision trees has compared with Naïve Bayes, kNN, eClass0, eClass1 and the Winner (KDDCup'99). Boosted decision trees outperformed the compared algorithms on real world intrusion dataset, KDDCup'99. On the basis of these results, it can be concluded that boosted decision trees may be a competitive alternative to these techniques in intrusion detection system.



(c) U2R attack with 41 features



(a) R2L attack with 41 features



(b) Probe attack with 41 features

Figures 1(a) - 1(e) show graphical comparison of boosted decision treesalgorithm with SVM, kNN and Naïve Bayes with feature selection. 12 features are selected from 41 features.

REFERENCES

1. C. C. Aggrawal, P. Yu, "Outlier Detection for High Dimensional Data", Proceedings of the ACM SIGMOD Conference, 2001.
2. P. P. Angelov, X. Zhou, "Evloving fuzzy rule based classifiers from data streams", IEEE Transaction on Fuzzy Systems, Vol 16, No. 6, pp. 1462- 1475, 2008.
3. R. Bane, N. Shivsharan, "Network intrusion detection system (NIDS)", pp. 1272-1277, 2008.
4. Barnett, T. Lewis, "Outliers in Statistical Data", John Wiley and Sons, NY, 1994.
5. S. T. Brugger, "Data mining methods for network intrusion detection", pp. 1-65, 2004.
6. M. M. Breunig, H. P. Kriegel, R. T. Ng, J. Sander, "LOF: Identifying Density-Based Local Outliers", Proceedings of the ACM SIGMOD Conference , 2000.
7. D. E. Denning, "An Intrusion Detection Model", IEEE Trans-actions onSoftware Engineering, SE-13, pp. 222-232, 1987.
8. H. S. Javitz, A. Valdes, "The NIDES Statistical Component: Description and Justification", Technical Report, Computer Science Laboratory, SRI International, 1993.
9. E Knorr, Ng, R.: Algorithms for Mining Distance-based Outliers in Large Data Sets. Proceedings of the VLDB Conference (1998).
10. W. Lee, S. J. Stolfo, "Data Mining Approaches for Intrusion Detection", Proceedings of the 1998 USENIX Security Symposium, 1998.
11. R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. P. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, M. A. Zissman, "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation. Proceedings DARPA Information Survivability Conference and Exposition (DISCEX) 2000", Vol 2, pp. 12--26, IEEE Computer Society Press, Los Alamitos, CA, 2000.

12. W. Lee, S. J. Stolfo, “Data mining approaches for intrusion detection” Proc. of the 7th USENIX Security Symp.. San Antonio, TX, 1998.
13. W. Lee, S. J. Stolfo, K. W. Mok, “A data mining framework for building intrusion detection models”, Proc. of the 1999 IEEE Symp.on Security and Privacy, pp. 120--132. Oakland, CA, 1999.
14. M. Masud, J. Gao, L. Khan, J. Han, “Classifying evolving data streams for intrusion detection”.
15. M. Panda, M. Patra, “Ensemble rule based classifiers for detecting network intrusions”, pp 19-22, 2009
16. S. Ramaswami, R. Rastogi, K.Shim, “Efficient Algorithms for Mining Outliers from Large Data Sets”, Proceedings of the ACM SIGMOD Conference, 2000.
17. H. Wang, W. Fan, P. Yu, J. Han, “Mining concept-drifting data streams using ensemble classifiers”, In Proceedings of the ACM SIGKDD, pp.226-235, Washington DC, 2003.
18. Z. Yu, J. Chen, T. Q. Zhu, “A novel adaptive intrusion detection system based on data mining”, pp.2390-2395, 2005.