Intrusion Detection System Using New Ensemble Boosting Approach

Snehlata S. Dongre and Kapil K. Wankhade, *Members, IACSIT*

***Abstract*—Security is a big issue for all networks in today's enterprise environment. Hackers and intruders have made many successful attempts to bring down high profile company networks and web services. Intrusion Detection System (IDS) is an important detection that is used as a countermeasure to preserve data integrity and system availability from attacks. The main reason for using data mining classification methods for Intrusion Detection System is due to the enormous volume of existing and newly appearing network data that require processing. Data mining is the best option for dandling such type of data. This paper presents the new idea of applying data mining classification techniques to intrusion detection systems to maximize the effectiveness in identifying attacks, thereby helping the users to construct more secure information systems. This paper uses ensemble boosting approach with adaptive sliding window for intrusion detection. The ensemble method is advantageous over single classifier.**

***Index Terms*—Adaptive sliding window, data mining, ensemble approach, adaptive sliding window, IDS.**

I. Introduction

Securing important data from malicious users has been a long time concern for many both in the industry as well as in research. Nowadays with web applications used to access large databases over a network the need for Intrusion Detection has become a dire necessity. When a Database is first designed, it is designed and architected based on initial requirements obtained from the users of the proposed database. There are few security leaks and the web application is well written for the predicted database transactions. Usually a Database system so designed will not be expected to be very susceptible to intrusion. It is a well known fact that no software can be made completely bug free. Loop holes are usually over looked due to poor testing or oversight as part of the database designer. Also the database

may require some re-definitions of the database access roles based on the changes in the user's tasks. New tables and views may have to be added or old ones removed which causes changes in the data dependencies among tables. Such changes are generally invoked to make the database more feasible and this sometimes drastically affects the security level. A once secure database now becomes a perfect haven for malicious attacks. This is the core problem that we are trying to solve in our paper.

A database which is a part of a network or a host is usually monitored by the database administrator. He defines the various access roles for the users. These users hence have restricted access. With a number of users accessing the database with usually common queries there is a high bottleneck that arises. To prevent slow access speeds, the result sets of some queries are cached in a database cache. This reduces the access time but also opens the door for malicious unauthorized accesses. Usually large enterprises have a lot of sensitive data and hence in most cases such caches are poorly used. Malicious activity also arises when access roles are changed or the permissions for a user are changed. Another way to intrude into the database is by performing an unauthorized sequence of transactions. For example, a delete operation on a data item cannot occur without reading the item first.

Intrusion Detection Systems (IDS) [1]-[4] have been developed to identify any unauthorized attempts or successful attacks on any type of monitored data or resources available as part of a network or host system. Most IDSs detect such malicious activity either at the transaction level or at the operating system (OS) level. It is also shown that transaction level attacks take care of most OS level attacks. But there are many attacks which occur internal to the network such as by a user with lesser privileges accessing data that requires more access rights. Such attacks can be identified by analyzing the transaction logs. A transaction log contains all the transactions made on a database. More on Transaction logs are explained in later sections. By analyzing these logs most malicious activity can be identified. In a typical database accessed over a network there may be as many as one million transactions a day and any kind of computational analysis will prove to be costly and tedious.

There have been a number of approaches to reduce the time using different approaches; the most recent effective strategy being data mining used for intrusion detection [5]-[8]. Data mining is the analysis of data to establish relationships and identify hidden patterns of data which otherwise would go unnoticed. Even so existing approaches require analysis of millions of records. Our approach reduces the time to determine the sensitive data patterns from changing data access roles in the database, thereby identify any malicious activity and allow secure database caching at the network level.

The proposed paper organized as, Section II explains about data mining. Section III discussed about IDS taxonomy. The new ensemble boosting algorithm has explained in Section IV. Experimental results are included in

Manuscript received May 25, 2012; revised June 25, 2012.

S. S. Dongre is with the Department of Computer Science and Engineering, G. H. Raisoni College of Engineering, Nagpur, Maharashtra, INDIA 440019 (e-mail: dongre.sneha@gmail.com).

K. K. Wankhade is with the Department of Information Technolgy, G. H. Raisoni College of Engineering, Nagpur, Maharashtra, INDIA 440019 (e-mail: kaps.wankhade@gmail.com).

Section V with concluding conclusion in Section VI.

II. Data Mining for IDS

Data mining is assisting various applications for required data analysis. Recently, data mining is becoming an important component in intrusion detection system. Different data mining approaches like classification, clustering, association rule, and outlier detection are frequently used to analyze network data to gain intrusion related knowledge

Data mining is an analytic process designed to explore data (usually large amounts of data - typically business or market related) in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. The ultimate goal of data mining is prediction. Predictive data mining is the most common type of data mining and one that has the most direct business applications. The process of data mining consists of three stages: (1) the initial exploration, (2) model building or pattern identification with validation/verification, and (3) deployment (i.e., the application of the model to new data in order to generate predictions). These steps are explained below

Step I- Exploration. This stage usually starts with data preparation which may involve cleaning data, data transformations, selecting subsets of records and - in case of data sets with large numbers of variables ("fields") - performing some preliminary feature selection operations to bring the number of variables to a manageable range (depending on the statistical methods which are being considered). Then, depending on the nature of the analytic problem, this first stage of the process of data mining may involve anywhere between a simple choice of straightforward predictors for a regression model, to elaborate exploratory analyses using a wide variety of graphical and statistical methods in order to identify the most relevant variables and determine the complexity and/or the general nature of models that can be taken into account in the next stage.

Step II- Model building and validation. This stage involves considering various models and choosing the best one based on their predictive performance. This may sound like a simple operation, but in fact, it sometimes involves a very elaborate process. There are a variety of techniques developed to achieve that goal - many of which are based on so-called "competitive evaluation of models," that is, applying different models to the same data set and then comparing their performance to choose the best. These techniques - which are often considered the core of predictive data mining - include: bagging (voting, averaging), boosting, stackin (stacked generalizations), and meta-learning.

Step III- Deployment. That final stage involves using the model selected as best in the previous stage and applying it to new data in order to generate predictions or estimates of the expected outcome.

III. IDS Taxonomy

The goal of IDS [1],[2],[9],[10] is to detect malicious traffic. In order to accomplish this, the IDS monitor all incoming and outgoing traffic. There are several approaches on the implementation of IDS. Among those, two are the most popular: Anomaly detection: This technique is based on the detection of traffic anomalies. The deviation of the monitored traffic from the normal profile is measured. Various different implementations of this technique have been proposed, based on the metrics used for measuring traffic profile deviation. Misuse/Signature detection: This technique looks for patterns and signatures of already known attacks in the network traffic. A constantly updated database is usually used to store the signatures of known attacks. The way this technique deals with intrusion detection resembles the way that anti-virus software operates.

IV. Building Classification Model

This classification model uses boosting ensemble method with Hoeffding tree as base model and adaptive sliding window.

A. Boosting

Boosting is a machine learning meta-algorithm for performing supervised learning. While boosting is not algorithmically constrained, most boosting algorithms consist of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier. When they are added, they are typically weighted in some way that is usually related to the weak learners' accuracy. After a weak learner is added, the data is reweighted: examples that are misclassified gain weight and examples that are classified correctly lose weight. Boosting focuses on the misclassified tuples, it risks overfitting the resulting composite model to such data. Therefore, sometimes the resulting “boosted” model may be less accurate than a single model derived from the same data. Bagging is less susceptible to model overfitting. While both can significantly improve accuracy in comparison to a single model, boosting tends to achieve greater accuracy. There is reason for the improvement in performance that it generates a hypothesis whose error on training set is small by combining many hypotheses whose error may be large. The effect of boosting has to do with variance reduction. However unlike bagging, boosting may also reduce the bias of the learning algorithm.

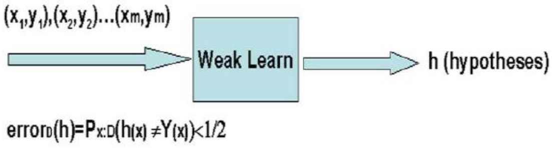

$(x_1,y_1),(x_2,y_2)\ldots(x_m,y_m)$
Weak Learn
h (hypotheses)
$\mathrm{error}_D(h)=P_{x:D}(h(x)\neq Y(x))<1/2$

Fig. 1. Weak learner algorithm

B. Adaptive Sliding Window

A window is maintained those keeps the most recent example and from which older examples are dropped according to some set of rules. The contents of the window can be used to detect change, to obtain updated statistics from the recent examples and rebuild the model(s) after data has changed.

ADWIN [11] is a change detector and estimator that solve in a well-specified way the problem of tracking the average of a stream of bits or real-valued numbers. ADWIN keeps a variable-length window of recently seen items, with the property that the window has the maximal length statistically consistent with the hypothesis “there has been no change in the average value inside the window”.

The idea of ADWIN method is simple: whenever two “large enough” subwindows of *W* exhibit “distinct enough” averages, one can conclude that the corresponding expected values are different, and the older portion of the window is dropped. The meaning of “large enough” and “distinct enough” can be made precise again by using the Hoeffding bound. The test eventually boils down to whether the average of the two subwindows is larger than a variable value ε_{cut} computed as follows,

$$m = \frac{2}{\frac{1}{|W_0|} + \frac{1}{|W_1|}} \tag{1}$$

$$\varepsilon_{cut} = \sqrt{\frac{1}{2m} \ln(4|W|/\delta)} \tag{2}$$

where m is the harmonic mean of $|W_0|$ and $|W_1|$.

The technical result [11] about the performance of ADWIN is :

False positive bound- If μ_t has remained constant within window *W* then the probability that ADWIN shrinks window at most δ.

False negative bound- suppose window *W* divides in to W_0W_1 in which W_1 contains the most recent items and we have $|\mu_{W0} - \mu_{W1}| > 2\varepsilon_{cut}$. Then with probability 1- δ ADWIN shrinks window *W* to W_1, or shorter.

C. Ensemble Boosting Algorithm

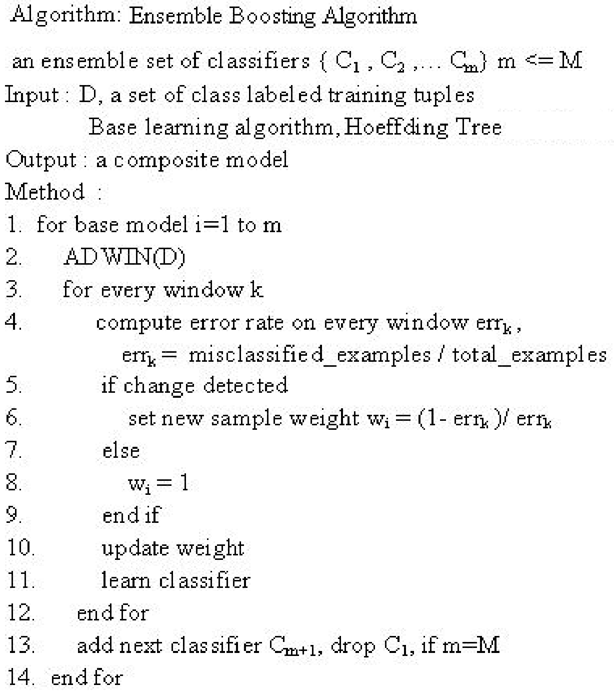
The proposed ensemble boosting method improves performance by using adaptive sliding window, hoeffding tree. Boosting method is used for improving the performance. In this algorithm ADWIN is parameter and assumption free in the sense that it automatically detects and adapts to the current rate of change. Its only parameter is a confidence bound δ. Window is not maintained explicitly but compressed using a variant of the exponential histogram technique [12]. 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.

Adaptive and Efficient Classifier Uses Hoeffding Tree as a base learner. Because of this algorithm woks faster and increases performance. With the help of following equation it assigns dynamic weight.

$$w_k = (1 - err_k) / err_k \tag{3}$$

where, err_k is error rate calculated for every window.

Fig. 2 shows algorithm for ensemble boosting algorithm. Change detection is the main feature of our algorithm, which guarantees that the ensemble can adapt promptly to changes. Change detection is conducted at every window. Our ensemble method achieves adaptability by detecting concept change and discarding old classifier when alarm of change raises.



```
Algorithm: Ensemble Boosting Algorithm

an ensemble set of classifiers { $C_1$ , $C_2$ ,... $C_m$} m <= M
Input : D, a set of class labeled training tuples
        Base learning algorithm, Hoeffding Tree
Output : a composite model
Method :
for base model i=1 to m
    ADWIN(D)
    for every window k
        compute error rate on every window $err_k$ ,
            $err_k$ = misclassified_examples / total_examples
        if change detected
            set new sample weight $w_i = (1 - err_k)/ err_k$
        else
            $w_i = 1$
        end if
        update weight
        learn classifier
    end for
    add next classifier $C_{m+1}$, drop $C_1$, if m=M
end for
```

Fig. 2. Ensemble Boosting Algorithm.

V. Experimental Results

The experiment is carried out using KDDCup’99 Intrusion Detection real time dataset. The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes.

Attacks fall into four main categories:

- DOS: denial-of-service, e.g. syn flood,
- R2L: unauthorized access from a remote machine, e.g. guessing password,
- U2R: unauthorized access to local superuser (root) privileges, e.g., various ``buffer overflow" attacks,
- Probing: surveillance and other probing, e.g., port scanning.

A. Confusion Matrix

One of the methods to evaluate the performance of a classifier is using confusion matrix. A Confusion matrix that summarizes the number of instances predicted correctly or incorrectly by a classification model.

The confusion matrix is more commonly named contingency table which is shown in Table I. For example we have two classes + and - and therefore a 2×2 confusion matrix, the matrix could be arbitrarily large. The number of correctly classified instances is the sum of diagonals in the matrix; all others are incorrectly classified (class “a” gets misclassified as “b” exactly twice, and class “b” gets misclassified as “a” three times). The following terminology is often used when referring to the counts tabulated in a confusion matrix Table II.

TABLE I: CONFUSION MATRIX

		Predicted Class	
		+	-
Actual Class	+	**TP**	**FN**
	-	**FP**	**TN**

- True Positive (TP) [13]: The True Positive means that the number of positive examples correctly detected by the classification model.
- True Negative (TN) [13]: The True Negative means that the number of negative examples correctly detected by the classification model.
- False Negative (FN) [13]: The False Negative means that the number of positive examples incorrectly detected as negative by the classification model.
- False Positive (FP) [13]: The False Positive means that the number of negative examples incorrectly detected as positives by the classification model.

The confusion matrix of proposed method is tabulated in Table II. The Ensemble Boosting algorithm has compared with The KDDCup’99 Winner, eClass0 [14], eClass1 [14], kNN, C4.5, Na ïve Bayes in terms of accuracy is as shown in Table III.

B. Recall and Precision

The Recall and Precision are two widely used metrics employed in applications where successful detection of one of the classes is considered more significant than detection of the other classes. A formal definition of these metrics is given below in equations 4 and 5. The Recall and Precision curve of proposed Ensemble Boosting Algorithm is as shown in Fig. 3.

$$\Pr ecision = TP/(TP+FP) \quad (4)$$

$$\operatorname{Re} call = TP/(TP+FN) \quad (5)$$

TABLE II: CONFUSION MATRIX OF ENSEMBLE BOOSTING ALGORITHM

	Normal	DOS	U2R	R2L	Probe	Total	Accuracy
Normal	96831	133	8	195	90	97257	**99.56**
DOS	285	391092	4	4	38	391423	**99.91**
U2R	26	0	11	6	7	50	**22**
R2L	19	5	0	1016	1	1041	**97.6**
Probe	245	8	0	5	3825	4083	**93.68**
Total	97406	391238	23	1226	3961		
Accuracy	99.41	99.96	47.82	82.87	96.67		

TABLE III: COMPARISON IN TERMS OF ACCURACY

	Normal	DOS	U2R	R2L	Probe
The KDDCup'99	99.5	97.1	13.2	8.4	83.3
eClass0	95.66	92.49	20.16	5.72	76.31
eClass1	99.14	96.53	11.84	9.23	63.13
kNN	99.6	97.3	35	0.6	75
C4.5	97.89	97.08	14.47	1.21	93.52
Naïve Bayes	94.51	97.19	26.92	73.44	90.33
Ensemble Boosting Algorithm	**99.56**	**99.91**	21.15	**90.23**	**93.13**

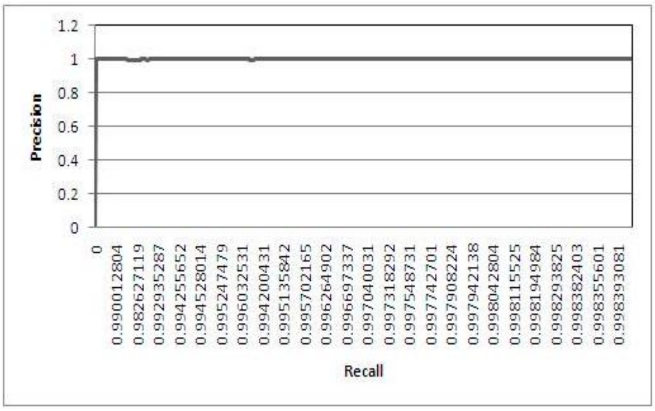


Fig. 3. ROC curve of ensemble boosting algorithm.

VI. CONCLUSION

This paper has presented a new data-mining based approach to deal with intrusion detection. The performance of our boosting ensemble with adaptive sliding window approach on the KDDCup’99 benchmark intrusion detection dataset achieved balanced detection rates for five classes. The experimental results show that the proposed ensemble method outperformed the compared algorithms. On the basis of above results, it is concluded that data mining is one of the best solution for intrusion detection.

REFERENCES

[1] D. E. Denning, “An intrusion detection model,” *IEEE Transactions on Software Engineering, SE-13*, pp. 222-232, 1987.

[2] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, “Network Intrusion Detection,” *IEEE Network*, vol. 8, no. 3, pp 26-41, June 1994.

[3] M. Rowton, “Introduction to network security intrusion detection,” December 2005.

[4] R. Bane and N. Shivsharan, “Network intrusion detection system (NIDS),” in *Proc. of First International Conference on Emerging Trends in Engineering and Technology,* 2008, pp. 1272-1277.

[5] W. Lee, S. J. Stolfo, and K. W. Mok, “Data mining approaches for intrusion detection,” in *Proceedings of the 7th USENIX Security Symposium*, 1998.

[6] S. T. Brugger, “Data mining methods for network intrusion detection,” pp. 1-65, 2004.

[7] Z. Yu, J. Chen, and T. Q. Zhu, “A novel adaptive intrusion detection system based on data mining,” in *Proc. of International Conference on Machine Learning and Cybernetics*, vol. 1, no. 9, 2005, pp. 2390-2395.

[8] M. Panda and M. Patra, “Ensemble rule based classifiers for detecting network intrusions,” pp. 19-22, 2009.

[9] V. Barnett and T. Lewis, *Outliers in statistical data*, John Wiley and Sons, NY, 1994.

[10] C. C. Aggrawal and P. Yu, “Outlier detection for high dimensional data,” in *Proceedings of the ACM SIGMOD Conference*, vol. 30, no. 2, 2001, pp 61-72.

[11] A. Bifet and R. Gavalda, “Learning from time changing data with adaptive windowing,” in *SIAM International Conference on Data Mining*, 2007, pp 443-449.

[12] M. Datar, A. Gionis, P. Indyk, and R. Motwani, “Maintaining stream statistics over sliding windows,” *SIAM Journal on Computing,* 14(1), pp 27-45, 2002.

[13] C. Ferri, P. Flach, and J. Hernandez-Orallo, “Learning decision trees using the area under the ROC curve,” in *Proceeding of the 19th International Conference on Machine Learning, Sydney,* July 2002, pp. 139-146.

[14] P. P. Angelov and X. Zhou, “Evloving fuzzy rule based classifiers from data streams,” *IEEE Transaction on Fuzzy Systems*, vol. 16, no. 6, pp. 1462-1475, 2008.

S. S. Dongre She received B.E. degree in Computer Science and Engineering from Pt. Ravishankar Shukla University, Raipur, India in 2007 and M.Tech. degree in Computer Engineering from University of Pune, Pune, India in 2010.

She is currently working as Assistant Professor the Department of Computer Science and Engineering at G. H. Raisoni College of Engineering, Nagpur, India. Number of publications is in reputed International conferences like IEEE. Her research is on Data Stream Mining, Machine Learning, Decision Support System, ANN and Embedded System. Her book has published on titled Data Streams Mining: Classification and Application, LAP Publication House, Germany, 2010.

Ms. Snehlata S. Dongre is a member of IACSIT, IEEE and ISTE organizations.

K. K. Wankhade He received B. E. degree in Information Technology from Swami Ramanand Teerth Marathwada University, Nanded, India in 2007and M. Tech. degree in Computer Engineering from University of Pune, Pune, India in 2010.

He is currently working as Assistant Professor the Department of Information Technology at G. H. Raisoni College of Engineering, Nagpur, India. Number of publications is in reputed Journal and International conferences like Springer and IEEE. His research is on Data Stream Mining, Machine Learning, Decision Support System, Artificial Neural Network and Embedded System. His book has published on titled Data Streams Mining: Classification and Application, LAP Publication House, Germany, 2010.

Mr. Kapil K. Wankhade is a member of IACSIT and IEEE organizations. He is currently working as a reviewer for Springer’s Evolving System Journal and IJCEE journal.