Performance Analysis of Supervised Approach for Pattern based IDs

V. K. Pachghare
Assistant Professor,
Department of Comp. Engg. & IT
College of Engineering,
Pune-5

Vaibhav K Khatavkar
Assistant Professor,
Department of Comp. Engg. & IT
College of Engineering,
Pune-5

Dr. Parag Kulkarni
Adjunct Professor
Department of Comp.. Engg. & IT
College of Engineering,
Pune-5

ABSTRACT
Aim of an intrusion detection system (IDS) is to distinguish the behavior of network. IDS should upgrade itself so as to cope up with the changing pattern of attacks. Also detection rate should be high since attack rate on the network is very high. In response to this problem, Pattern Based Algorithm is proposed which has high detection rate and low false alarm rate. The work is related to the development of pattern based IDS using supervised approach. The algorithm uses decision stumps as weak classifier. The decision rules are provided for both categorical and continuous features. Weak classifier for continuous features and weak classifier for categorical features are combined to form a strong classifier. The experimentation is performed on KDD CUP 99 dataset and NSL KDD data which is revised KDD CUP 99 data.

Keywords
Pattern, Intrusion detection system, Supervised learning, AdaBoost, Machine Learning

1. INTRODUCTION
There are two main approaches to design IDS: misuse based IDS and anomaly based IDS [1]. Both misuse and anomaly detection approaches are typically presented in terms of distinct training and testing phases.

Modern IDS's are extremely diverse in the techniques they employ to gather and analyze data. The IDS which depends on the audit data usually results in an inflexible detection that is unable to detect an attack if the sequence of events is slightly different from the predefined profile [2, 3]. The techniques used to implement IDS are generally based on Fuzzy Logic (FL), Artificial Neural Networks (ANNs), Probabilistic Reasoning (PR), and Genetic Algorithms (GAs) [4].

Machine learning algorithms have proven to be of great practical value in a variety of application domains. Various machine learning techniques which could be used in intrusion detection systems are: Concept learning, Decision tree, neural networks, Bayesian learning, Genetic algorithms and genetic programming, Instant based learning, Inductive logic programming, Analytical learning, Adaptive and Analytical learning, Inductive and Analytical learning.

Recently, methods from machine learning and pattern recognition have been utilized to detect intrusions. Learning algorithms can be categorized as supervised, unsupervised and semi-supervised. All the three approaches can be used for intrusion detection. Supervised machine learning methods require labeled ground truth data. The objective of supervised learning is to learn to assign correct labels to new unseen examples of the same task. For supervised learning for intrusion detection, there are mainly neural network (NN)-based approaches [5, 6], and support vector machine (SVM) based approaches [7, 8].

In this paper we propose supervised approach for intrusion detection. In the supervised approach we use the labeled data for training and unlabeled data for testing. The rest of the paper is organized as follows. Section 2 describes the related work about intrusion detection system. Section 3 describes our proposed supervised approach. Section 4 describes experiments and results followed by a conclusion in Section 5.

2. RELATED WORK
Recently, methods from machine learning and pattern recognition have been utilized to detect intrusions. Supervised learning and unsupervised learning are both used. For supervised learning for intrusion detection, there are mainly neural network (NN)-based approaches [5, 6], and support vector machine (SVM) based approaches [7, 8].

2.1 Neural Network (NN)
Bonifacio et al. [9] propose an NN for distinguishing between intrusions and normal behaviors. They unify the coding of categorical fields and the coding of character string fields in order to map the network data to an NN. Rapaka et al. [10] use execution numbers of system calls in a host machine as the features of network behaviors to train the NN. Zhang et al. [6] propose an approach for intrusion detection using hierarchical NNs. Han and Cho [11] use evolutionary NNs to detect intrusions. A neural network based IDS typically consist of a single neural network based on either misuse detection or anomaly detection. Neural network with good pattern classification abilities typically used for misuse detection, such as Multilayer Perception, Radial Basis function networks, etc. Neural network with good classification abilities typically used for anomaly detection, such as Self organizing maps (SOM), Competitive learning neural network, etc. SOM is based on competitive learning. The Winner takes all neuron and Forms a topographic map of input patterns i.e. spatial locations of

neurons in the lattice are indicative of statistical features contained in the input patterns.

2.2 Support Vector Machine (SVM)

Mukkamala et al. [12] use SVMs to distinguish between normal network behaviors and intrusions and further identify important features for intrusion detection. Mill and Inoue [13] propose the TreeSVM and ArraySVM for solving the problem of inefficiency of the sequential minimal optimization algorithm for the large set of training data in intrusion detection. Zhang and Shen [28] propose an approach for online training of SVMs for real-time intrusion detection based on an improved text categorization model. SVM are learning machines that plot the training vectors in high-dimensional feature space, labeling each vector by its class. SVMs classify data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in the feature space.

2.3 Boosting

Machine learning studies automatic techniques for learning to make accurate predictions based on past observations. Building a highly accurate prediction rule for various patterns is certainly a difficult task. On the other hand, it is not hard at all to come up with very rough rules of thumb that are only moderately accurate. Boosting, the machine-learning method that is the subject of this report, is based on the observation that finding many rough rules of thumb can be a lot easier than finding a single, highly accurate prediction rule. Boosting is a general method which attempts to boost the accuracy of any given learning algorithm. Boosting has its roots in a theoretical framework for studying machine learning called the PAC learning model, due to Valiant [14]; Kearns and Vazirani [15] gives a good introduction to this model. Schapire [16] developed a much more efficient boosting algorithm which although optimal in a certain sense, nevertheless suffered from certain practical drawbacks.

3. PROPOSED WORK

According to the characteristics of the AdaBoost algorithm and the characteristics of the network intrusion detection problem, the framework of our approach consists of the following four modules: feature extraction, data labeling, design of the weak classifiers, and construction of the strong classifier, as shown in Figure 1. The framework of proposed algorithm is explained in our previous work [17].

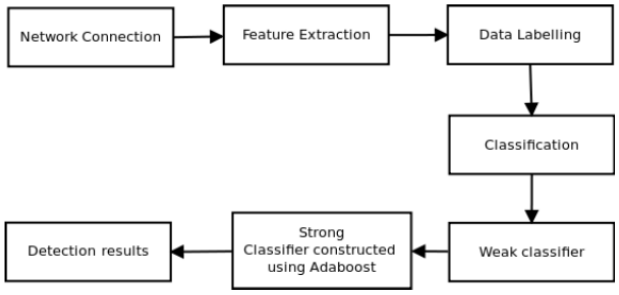

Figure 1: Architecture for Supervised IDS

Weak Classifier Design: A group of weak classifiers has to be prepared as inputs of Adaboost algorithm. They can be linear classifiers, ANNs or other common classifiers. In our algorithm, we select decision stumps as weak classifiers due to its simplicity. For every feature f, its value range could be divided into two non overlapping value subsets C_p^f and C_n^f, and the decision stump on f takes the form as follow:

$$h_f(x) = \begin{cases} +1 & x(f) \in C_p^f \\ -1 & x(f) \in C_n^f \end{cases}$$

where, x(f) indicates the value of x on feature f.

Algorithm: In the AdaBoost algorithm, weak classifiers are selected iteratively from a number of candidate weak classifiers and are combined linearly to form a strong classifier for classifying the network data. In the AdaBoost algorithm, weak classifiers are selected iteratively from a number of candidate weak classifiers and are combined linearly to form a strong classifier for classifying the network data.

Let $H = \{\tilde{h}_f\}$ be the set of constructed weak classifiers. Let the set of training sample data be $\{(x_1, y_1),...(x_i, y_i),....,(x_n, y_n)\}$, where x_i denotes the i^{th} feature vector, $y_i \in \{+1,-1\}$ is the label of the i^{th} feature vector, denoting whether the feature vector represents a normal behavior or not;

Let $\{w_1,..., w_i,..., w_n\}$ be the sample weights that reflect the importance degrees of the samples and, in statistical terms, represents an estimation of the sample distribution.

The supervised algorithm for intrusion detection is described as follows:

1. Initialize Weights as:

$$w_i(1) \quad (i = 1,..., n)$$

$$satisfying \quad \sum_{i=1}^{n} w_i(1) = 1$$

2. Observe the following for *(t = 1… T).*

a) Let ε_j be the sum of the weighted classification errors for the weak classifier hj

$$\varepsilon_j = \sum_{i=1}^{n} w_i(t) I\left[y_i \neq h_j(x_i)\right] \quad (1)$$

Where,

$$I_{[\gamma]} = \begin{cases} 1, & \gamma = true \\ 0, & \gamma = false \end{cases} \quad (2)$$

Choose, from constructed weak classifiers, the weak classifier h(t) that minimizes the sum of the weighted classification errors

$$h(t) = \arg \min_{hj \in H} \varepsilon_j \quad (3)$$

b) Calculate the sum of the weighted classification errors $\varepsilon(t)$ for the chosen weak classifier $h(t)$.

c) Let

$$\alpha(t) = \frac{1}{2} \log\left(\frac{1-\varepsilon(t)}{\varepsilon(t)}\right) \quad (4)$$

d) Update the weights by

$$w_i(t+1) = \left(\frac{w_i(t)\exp(-\alpha(t)y_i h(t)(x_i))}{Z(t)} \right) \quad (5)$$

where $Z(t)$ is a normalization factor

$$Z(t) = \sum_{k=i}^{n} \exp(-\alpha(t)y_i h(t)(x_k)) \quad (6)$$

3. The strong classifier is defined by

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha(t)h(t)(x) \right) \quad (7)$$

We explain two points:

- By combining the decision stumps for both categorical and continuous features into a strong classifier, the relations between categorical and continuous features are handled naturally, without any forced conversions between continuous and categorical features.
- The decision stumps minimize the sum of the false-classification rates for normal and attack samples. It is guaranteed that the misclassification rates for the selected weak classifiers are lower than 50% this ensures the convergence of the algorithm,

4. EXPERIMENTATION AND RESULTS

Our experimentation have two phases, training phase followed by the testing phase. In training phase, first the training data is labeled either normal or attack. Then the categorial and continuous features are extracted. Optimal decision stump is prepared from the extracted features. Labels of training data are predicted based on the optimal decision stump. The confusion matrix is constructed on the training data. The confusion matrix is used to get detection rate , false rates, etc.

KDDCUP 99 [18] is the mostly widely used data set for the evaluation of network-based anomaly based intrusion detection. First we utilized the KDD CUP 1999 data set [18] for our experiments. There are four general types of attacks appeared in the data set: DOS (denial of service), U2R (user to root), R2L (remote to local) and PROBE. In each of the four, there are many low level types of attacks. Detailed descriptions about the four general types can be found in [18].

The number of samples of various types in the training data set is listed in Table 1 Confusion matrix for KDD 10 percent training data using our proposed algorithm is shown in Table 2

The supervised algorithm gives 99.7 % detection rate and 0.06 false positive rate which is better than other supervised approaches [17].

Table 1: Number of Samples in Training Data Set

Normal	**Attack**				**Total**
	DOS	***U2R***	***R2L***	***PROBE***	
	391458	52	1126	4107	
97278	396743				494021

Table 2: Performance of proposed supervised algorithm in Training Data Set

	Normal	DOS	R2L	U2R	PROBE	%
Normal	97218	19	9	0	32	99.93
DOS	20	391413	3	4	18	99.98
R2L	15	0	1102	4	5	98.04
U2R	5	0	0	45	2	88.46
PROBE	40	11	9	0	4047	98.53
%	99.92	99.99	98.22	85.18	98.58	99.95

There are some deficiencies in KDDCUP 99 which cause evaluation results to be unreliable. There are redundant records in KDDCUP'99 data set which will bais the classifier towards more frequent records while training and testing. The size of KDDCUP 99 training and testing data is large which makes it difficult to run experiments on the complete set. Many researchers randomly select a small portion from KDDCUP 99 data set for evaluation of their system. Thus evaluation results will be inconsistent and difficult to compare. [19] gives statistical analysis of KDDCUP 99 data set.

The new data set, namely, NSLKDD [20] is available publicly. The NSL-KDD data set can be used for evaluation of different IDSs on complete dataset without randomly selecting a small portion from the dataset. The NSL-KDD dataset consists of data with various categories like Normal, DOS, U2R, R2L and PROBE. The attacks have subcategories. The number of samples is given in Table 3 now we perform our experiments with NSL-KDD dataset. Table 4 shows the confusion matrix for NSL-KDD training dataset.

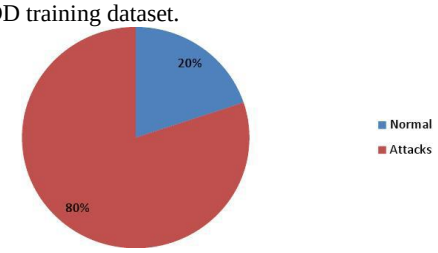

Figure 2: KDD CUP 99 Train Data

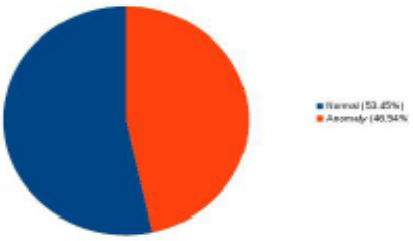

Figure 3: NSL-KDD Train Data

Table 3: Number of samples in NSL-KDD training data

Normal	Attack				Total
	DOS	U2R	R2L	PROBE	
	45912	54	993	11654	
67360	58613				125973

Table 4: Confusion Matrix for NSL-KDD training data

	Normal	DOS	R2L	U2R	PROBE	%
Normal	67283	19	9	0	32	99.88
DOS	21	45882	2	4	18	99.93
R2L	14	0	973	4	4	97.98
U2R	2	0	0	46	4	85.18
PROBE	40	11	9	0	11596	99.50
%	99.88	99.93	97.98	82.60	99.50	

Table 5: Detection Results on both the Data Sets

Test data used	FPR(%)	DR(%)
KDD CUP 99	0.06	99.7
NSL-KDD	0.089	99.84

The detection rate for supervised algorithm on NSL-KDD dataset is 99.84 % and false positive rate is 0.089 % The performance of supervised algorithm on KDD CUP 10 % and NSL-KDD dataset is given as in Table 5 The detection rate for NSL-KDD dataset is greater than KDD CUP 10 % dataset. The false positive rate is also greater for NSL-KDD dataset than KDD 10% dataset.

5. CONCLUSION

In the last twenty years, Intrusion Detection Systems have slowly evolved from host and operating system specific application to distributed systems that involve a wide array of operating system. The challenges that lie ahead for the next generation of Intrusion Detection Systems are many. Traditional Intrusion Systems have not adapted adequately to new networking paradigms like wireless and mobile networks. Factors like noise in the audit data, constantly changing traffic profiles and the large amount of network traffic make it difficult to build a normal traffic profile of a network for the purpose intrusion detection. A perennial problem that prevents widespread deployment of IDS is their inability to suppress false alarms. Therefore, the primary and probably the most important challenge that needs to be met is the development of effective strategies to reduce the high rate of false alarms. The experimental results show that the proposed algorithms have very low false alarm rate for training and testing. It can be seen from the results that the detection for supervised approach using NSL-KDD is better than KDD CUP 99. The proposed algorithm classifies data more correctly in NSL-KDD dataset than on KDD dataset.

6. REFERENCES

[1] Denning D, An Intrusion-Detection Model, IEEE Transactions on Software Engineering, Vol. SE- 13, No 2, Feb 1987.

[2] V K. Pachghare, Dr. Parag Kulkarni, and Deven Nikam, Overview of Intrusion Detection Systems, International Journal of Computer Science and Engineering Systems, Vol. 3, No. 3, 265-268, 2009.

[3] S. Mukkamala and A H. Sung, A comparative study of techniques for intrusion detection, in Proc. Int. Conf. Tools Artif. Intell., 2003, pp. 570-577

[4] V K Pachghare and Parag Kulkarni , Performance Analysis of Pattern Based Network Security, 2nd International Conference on Computer Technology and Development (ICCTD 2010) IEEE, pg 277 281

[5] Y.-H. Liu, D.-X. Tian, and A.-M. Wang, "Annids: Intrusion detection system based on articial neural network", in Proc. Int. Conf. Mach. Learn. Cybern., Nov. 2003, vol. 3, pp. 1337-1342.

[6] C. Zhang, J. Jiang, and M. Kamel, "Intrusion detection using hierarchical neural networks", Pattern Recognit. Lett., vol. 26, no. 6, pp. 779-791, May 2005.

[7] P. Hong and R. E. Schapire, "An intrusion detection method based on rough set and SVM algorithm", in Proc. Int. Conf. Commun., Circuits Syst., Jun. 2004, vol. 2, pp. 1127-1130.

[8] Z. Zhang and H. Shen, "Online training of SVMs for real-time intrusion detection", in Proc. Int. Conf. Adv. Inf. Netw. Appl., 2004, vol. 1, pp. 568-573.

[9] J. M. Bonifacio, Jr., A. M. Cansian, A. C. P. L. F. De Carvalho, and E. S. Moreira, Neural networks applied in intrusion detection systems, in Proc. IEEE Int. Joint Conf. Neural Netw., 1998, vol. 1, pp. 205-210.

[10] A. Rapaka, A. Novokhodko, and D. Wunsch, Intrusion detection using radial basis function network on sequences of system calls, in Proc. Int. Joint Conf. Neural Netw., 2003, vol. 3, pp. 1820-1825.

[11] S. J. Han and S. B. Cho, Evolutionary neural networks for anomaly detection based on the behavior of a program, IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 36, no. 3, pp. 559-570, Jun. 2006.

[12] S. Mukkamala, G. Janoski, and A. H. Sung, Intrusion detection using neural networks and support vector machines, in Proc. Int. Joint Conf. Neural Netw., 2002, vol. 2, pp. 1702-1707

[13] J. Mill and A. Inoue, Support vector classi_ers and network intrusion detection, in Proc. Int. Conf. Fuzzy Syst., 2004, vol. 1, pp. 407-410.

[14] L. G. Valiant, A theory of the learnable, Communication of the ACM, 27(11):1134 1142, November 1984.

[15] Michel J. Kearns and Umesh V. Vazirani, An Introduction to Computational Learning Theory, MIT Press, 1994.

[16] Freund and R. E. Schapire, A Decision-Theoretic Generalization Of Online Learn- ing And An Application To Boosting, J. Comput. Syst. Sci., vol. 55, no. 1, pp. 119-139, Aug. 1997.

[17] Vivek A Patole, V K Pachghare and Parag Kulkarni, AdaBoost Algorithm to Build Pattern Based Network Security, International Journal of Information Processing, 5(1), 57-63, 2011

[18] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Ocotber 2007.

[19] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, A Detailed Analysis of the KDD CUP 99 Data Set, Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

[20] NSL-KDD data set for network-based intrusion detection systems. Available on : http://iscx.ca/NSL-KDD/