Potentiate the Detection-rate of Network Intrusion Detection using Adaboost algorithm

Ankita Chowdhury[1], Rupali Bhanuse[2], Shradha Birajdar[3], Devendra Ghorsad[4]
[1, 2, 3, 4] Department Of Information Technology
[1, 2, 3, 4] AISSMS IOIT, Kennedy Road, Near RTO, Pune-411001

Abstract- *Network intrusion detection aims at differentiate the intrusions on the Internet from normal use of Internet and is an essential part of the information security system. Network consists of nodes whose operation can be controlled by underlying network. KDDCUP'99 is the mostly widely used data set for the evaluation of signature-based IDSs. In this paper, first a conventional online Adaboost process is used where decision stumps are used as weak classifier. In the second algorithm, online Adaboost process is used and online Gaussian mixture models (GMMs) are used as weak classifier. In addition to the algorithm proposed particle swarm optimization (PSO) and support vector machine (SVM) is used. A distributed intrusion detection framework is proposed, in which a local parameterized detection model is constructed in individual node using the online Adaboost algorithm. The global detection model is constructed in each node by combining the local parametric models using a minimum number of samples in the node, which is used to detect intrusions. The algorithm integrates the local detection models global model in each node. This handles the intrusion category found in other nodes, without having to share samples of these intrusion types.*

Keywords- Adaboost, Decision Stumps, Dynamic Distributed System, GMM, KDD'99, Network Intrusion, PSO, SVM.

I. INTRODUCTION

Internet plays a vital role in communication between people. To ensure a secure communication between two parties, we need a security system to detect the attacks very efficiently. Network intrusion detection serves as a major system to work with other security system to provide protection to the computer networks. The main focus of network intrusion detection techniques is to catch, look into the various header parts and data portion of the packets and classify the attack packets from the normal packets. There are mainly 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 differentiate from the already learned activities. The main feature of Anomaly based detection is the capability of detecting the novel attacks which are deviate from the already learned attacks. The main drawback of anomaly based detection is that it wrongly classifies the normal user behaviors as attacks, which would result in a higher false positive rate. The misuse based detection mechanism uses the certain standard patterns of attacks to detect intrusions by representation of the same type of attacks Misuse based detection has higher network attack detection rate than anomaly based detection mechanism but it is failing to detect novel attacks. An intruder is an attacker, person or group of people who initiates the activities during intervention. Attackers can be from within the trust network, person who has the access to use system with normal user rights or someone who uses a hole in some OS to escalate their access level or admin rights. It can be from external side of the system or network that is someone on another network or even in some other country who utilize a weakness, vulnerability in an insecure network service on the system to take unauthorized entry and access of the trust network.

In traditional centralized intrusion detection, in which all the network data are sent to a central site for processing, the raw data communication occupies considerable network bandwidth. There is a computational load in the central site and the privacy of the data obtained from the local nodes cannot be protected. Distributed detection, which shares local intrusion detection model learned in local nodes, can reduce data communications, distribute the computational burden, and protect privacy.

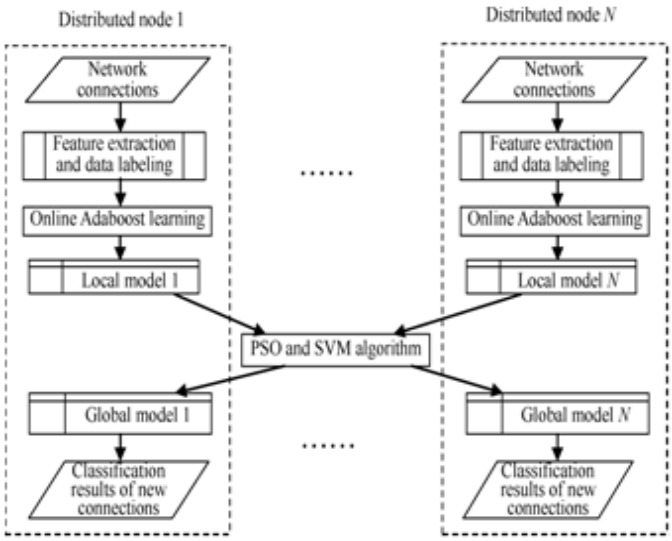
In this paper, we are using KDD'99 as our training data set, and attacks from them as testing data. We use decision stumps as weak classifiers in the first instance. GMM classifier is constructed for each classifier. With this the local models are constructed at each node. After this, PSO and SVM are used for reducing and combining the results. This results in the creation of the global models.

II. OVERVIEW

A. Our Framework

In the distributed intrusion detection framework, every node independently constructs its own local intrusion detection model according to its own data. By integrating all

the local models, at each node, a global model is trained using a small number of the sample in the node, without sharing any of the original training data between nodes. The global model is used to detect intrusions at the node. Fig. 1 gives an overview of our framework that consists of the modules of data preprocessing, local models, and global models.



1. **Data Preprocessing:** For each network connection, there are three groups of characteristics that are commonly used for intrusion detection are extracted: basic characteristic of each transmission control protocol (TCP) connections, content features inside a connection suggested by domain knowledge, and traffic features evaluated using a two-second time window. The extracted attribute values from a network connection form a vector x = (x1, x2, . . . ,xD), where D is the number of character components. There are continuous and categorical features, and the value ranges of the features may dissimilar greatly from each other. The framework for constructing these features can be found in features. A set of data is classified for training purposes. There are number of types of attacks on the Internet. The attack samples are labeled as −1, −2,... relying on the attack type, and the normal samples are all labeled as +1.

2. **Local Models:** The establishment of a local detection model at every node includes the design of weak classifiers and Adaboost-based training. Every individual feature component corresponds to a weak classifier. In this way, the mixed characteristics data for the network connections can be handled naturally, and full use can be made of the information in every feature. The Adaboost training is implemented using only the local training samples at each node. After training, every node contains a parametric model that incorporates the parameters of the weak classifiers and the ensemble weights.

3. **Global Models:** By splitting all the local parametric models, a global model is constructed using the PSO and SVM-based algorithm in every node. The global model in each node fuses the information learned from all the local nodes using a less number of training samples in the node. Feature vectors of new network connections to the node are input to the global classifier, and labeled as either normal or attacks. The results of the global model in the node are used to update the local model in the node and the updated model is then shared by other nodes.

III. RELATED WORK

1. J. B. D. Caberera, B. Ravichandran, and R. K. Mehra
Examines the application of statistical traffic modeling for detecting novel attacks against computer networks. In this paper it is discuss the application of network activity models and application models using the 1998 DARPA Intrusion Detection Evaluation data set. Network activity models monitor the volume of traffic in the network, while application models describe the operation of application protocols.

2. W. Lee, S. J. Stolfo, and K. Mork-
This paper describes a data mining framework for adaptively building Intrusion Detection (ID) models. The central idea is to utilize auditing programs to extract an extensive set of features that describe each network connection or host session, and apply data mining programs to learn rules that accurately capture the behavior of intrusions and normal activities. These rules can then be used for misuse detection and anomaly detection.

3. H. G. Kayacik, A. N. Zincir-heywood, and M. T. Heywood-
An approach to network intrusion detection is investigated, based purely on a hierarchy of Self-Organizing Feature Maps. Our principle interest is to establish just how far such an approach can be taken in practice. To do so, the KDD benchmark dataset from the International Knowledge Discovery and Data Mining Tools Competition is employed.

IV. CHALLENGES IN EXISTING SYSTEMS

1. Network environments and the attacks training data changes rapidly over time, as new types of attack appear. In addition, the size of the training data expands over time and can become very large. Most existing algorithms for training intrusion detectors are offline. The intrusion detector must be reinforced periodically in batch mode in

order to keep up with the changes in the network. This reinforcing is time consuming.

2. There are various types of attributes for network connection data, including both categorical and continuous ones, and the value ranges for different attributes differ greatly—from {0, 1} to describe the normal or error status of a connection, to specify the number of data bytes sent from source to destination. The combination of data with different attributes without loss of information is major to maintain the accuracy of intrusion detectors.
3. In traditional centralized intrusion detection, in which all the network data are sent to a central site for processing, the raw data communications occupy considerable network bandwidth. There is a estimation burden in the central site and the privacy of the data obtained from the local nodes cannot be protected.

V. PROPOSED SYSTEM

The classical Adaboost algorithm transfers the training task in batch mode. A number of weak classifiers are constructed using a training set. Weights, which specify the importance of the training samples, are derived from the classification fallacy of the weak classifiers. The final strong classifier is an ensemble of weak classifiers. The classification error of the final strong classifier intersects to 0. However, the Adaboost algorithm based on offline learning is not suitable for networks. We apply online type of Adaboost to construct the local intrusion detection models. It is proved in that the strong classifier acquired by the online Adaboost converges to the strong classifier acquired by the offline Adaboost as the no. of training samples increases. In the following, we first introduce the weak classifiers for intrusion detection, and then describe the online Adaboost-based intrusion detection algorithms.

Weak Classifiers:

Weak classifier that can be updated online match the requirement of dynamic intrusion detection. We consider two types of weak classifier. The first type consists of decision stumps for classifying attacks and normal behaviors. The second type is online GMMs that model a distribution of values of each feature component for each attack type.

Decision Stumps: A decision stump is a decision tree with a root node and two leaf nodes. A decision stump is constructed for every feature component of the network connection data.

For a categorical feature f, the set of attribute values Cf is distributed into two subsets Cfi and Cf n with no intersection and the decision stump takes the form:

$$h^f(x) = \begin{cases} 1 & x^f \in C^f_n \\ -1 & x^f \in C^f_i \end{cases} \quad (1)$$

Where xf is the attribute value of x on the feature f. The subsets Cfi and Cfn are determined using the training samples: for an attribute value z on a feature f, all the training samples whose attribute values on f are equal to z are found; if the number of attack samples in these samples is more than the number of normal samples, then z is assigned to Cfi, otherwise, z is assigned to Cfn. In this way, the false alarm rate for the training samples is minimized. For a continuous feature f, the range of attribute values is split by a threshold v, and the decision stump takes the form

$$h^f(x) = \begin{cases} 1 & x^f > v \\ -1 & x^f \le v \end{cases} \quad (2)$$

The threshold v is determined by minimizing the false alarm rate for the training samples.

Online GMM: For the samples of each attack type or the normal samples, we use a GMM to model the data on each feature component. Let c ∈ {+1,−1,−2, . . . ,−M} be a sample label, where +1 represents the normal samples and "−1,−2,. . .−M" represents different types of attacks where M is the number of attack types.

B. Adaptable Initial Sample Weights:

We use the detection rate and the false alarm rate to evaluate the performance of the algorithm for detecting network attacks. It is necessary to pay more observation to the false alarm rate because, in real applications, more network behaviors are normal. A high false alarm rate wastes resources, as each alarm has to be checked. For Adaboost-based learning algorithms, the detection rate and the false alarm rate depend on the initial weights of the training samples. So we suggestadjusting the initial sample weights in order to balance the detection rate and the false alarm rate. We initiate a parameter r∈ (0, 1) for setting the initial weight λ of each training sample

$$\lambda = \begin{cases} \frac{Nnormal + Nintrusion}{Nnormal}.r & for\ normal\ connections \\ \frac{Nnormal + Nintrusion}{Nintrusion}.(1-r) & for\ network\ intrusions \end{cases}$$

Where N_{normal} and $N_{intrusion}$ are approximated using the no. of normal samples and attack samples that have been input online to train the classifier. The sums of the weights for the

normal samples and the attack samples are $(N_{normal} + N_{intrusion}) \cdot r$ and $(N_{normal} + N_{intrusion}).(1 - r)$, respectively. Through adjusting the value of the parameter r, we change the importance of normal samples or attack samples in the training process, and then make a tradeoff between the detection rate and the false alarm rate of the final detector. The selection of r depends on the proportion of the normal samples in the training data, and the requirements for the detection rate and the false alarm rate in specific applications.

C. Local Parameterized Models

Subsequent to the construction of the weak classifiers and the online Adaboost learning, a local parameterized detection model ϕ is formed in each node. The local model consists of the parameters ϕ_w of the weak classifiers and the parameters ϕ_d for constructing the Adaboost strong classifier: $\phi = \{\phi_w, \phi_d\}$. The parameters for each decision stump-based weak classifier include the subsets C^f_i and C^f_n for each categorical feature and the thresholds v for each continuous feature. The parameters for each GMM-based weak classifier include a set of GMM parameters $\varphi_w = \{\theta^c_j | j = 1, 2, ..., D;\ c = 1, -1, -2, ...\}$. The parameters of the strong classifier for the online Adaboost algorithm include a set of ensemble weights $\varphi_d = \{\alpha_t | t = 1, 2, ..., D\}$ for the weak classifiers.

VI. GLOBAL DETECTION MODELS

The local parametric detection model are shared among all the nodes and merged in each node to produce a global intrusion detector using a small number of samples left in the node .This global intrusion detector is more correct than the local detectors that may be only adequate for specific attack types, due to the limited training data available at each node. Some researchers fuse multi classifiers by combining the output results of all the classifiers into a vector, and then using a classifier, such as SVM or ANN, to classify the vectors. The combination of the local intrusion detection models has two problems. First, there may be large performance gaps between the local detection models for different types of attacks, mainly for new attack types that have not appeared previously. So, the sum rule may not be the best choice for combining the local detectors. Second, some of the local models may be similar for a test sample. If the results of the local models for the test sample are merged into a vector, the dimension of the vector has to be reduced to choose the best combination of the results from local models. To solve the above two problems, we integrate the PSO and SVM algorithms, in each node, to construct the global detection model. The PSO is a population search algorithm that simulates the social behavior of birds' flocking. The SVM is a learning algorithm based on the structural risk minimization principle from statistical learning theory. It has good performance even if the set of the training samples is small.

We use the knowledge discovery and data mining (KDD) CUP 1999 dataset to test our algorithms. This dataset is still the most trustful and tenable public benchmark dataset for evaluating network intrusion detection algorithms. In the dataset, 41 features including nine categorical features and 32 continuous features are extracted for each network connection. Attacks in the dataset fall into the following four main categories. 1) DOS: denial-of-service. 2) R2L: unauthorized access from a remote machine, e.g., guessing password. 3) U2R: unauthorized access to local super-user (root) privileges. 4) Probe: surveillance and other probing, e.g., port scanning. Each of the four categories contains some low-grade attack types. The test dataset includes some attack types that do not exist in the training dataset. The numbers of normal connections and each type of attacks in the training and test datasets are listed in Table I. In the following, we first introduce the performances of our online learning-based intrusion detection algorithms: one with decision stumps and the traditional online Adaboost process, and the other with online GMMs and our proposed online Adaboost process. Then, the performance of our PSO and SVM-based distributed intrusion detection algorithm is evaluated.

TABLE I The KDD CUP 1999 Dataset

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

A. Experimental Setup

All our experiments are performed on following hardware and software.

Hardware:

Pentium, 1.1 GHz processor with 1 GB RAM and 20GB hard Disk and Network Interface

Software:

Windows 7 64 bit, Java and J2EE (J2SDK 1.5), Net Beans IDE 8.0.1

Database: MySQL

Algorithm	Features	Training data		Test data	
		Detection rate (%)	False Alarm rate	Detection rate (%)	False Alarm rate
Decision stumps +Traditional online Adaboost	Only continuous	98.68	8.35	90.05	13.76
	Continuous +Categorical	98.93	2.37	91.27	8.38
Online GMM + Our online Adaboost	Only continuous	98.79	7.83	91.33	11.34
	Continuous +Categorical	99.02	2.22	92.66	2.67

NUMBER OF SAMPLES OF VARIOUS TYPES IN THE TRAINING SET

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

NUMBER OF SAMPLES OF VARIOUS TYPES IN THE Test SET

Normal	Attack					Total
	DOS	U2R	R2L	PROBE	Others	
	223298	39	5993	2377	18729	
60593	250436					311029

RESULTS AND DISCUSSION

In machine learning and data mining algorithms, many different measures are used to evaluate the Classification models

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

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

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

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

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)

$$\frac{\text{Total detected attacks} * 100}{\text{Total attacks}}$$

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

$$\frac{\text{Total misclassfiedinstances} * 100}{\text{Total normalinstances}}$$

IV. CONCLUSION

In this paper, we proposed online Adaboost-based intrusion detection algorithms, in which decision stumps and online GMMs were used as weak classifiers for the traditional online Adaboost and our proposed online Adaboost, respectively. The results of the algorithm using decision stumps and the traditional online Adaboost were compared with the results of the algorithm using online GMMs and our online Adaboost. We further proposed a distributed intrusion detection framework, in which the parameters in the online Adaboost algorithm formed the local detection model for each node, and local models were combined into a global detection model in each node using a PSO and SVM-based algorithm. The advantages of our work are as follows:

1) Our online Adaboost-based algorithms successfully overcame the difficulties in handling the mixed-attributes of network connection data;
2) The online mode in our algorithms assures the adaptability of our algorithms to the changing environments; the information in new samples was incorporated online into the classifier, while maintaining high detection accuracy;
3) Our local parameterized detection models were suitable for information sharing: only a very small number of data were shared among nodes;
4) No original network data were shared in the framework so that the data privacy was protected.

REFERENCES

[1] Weiming Hu, Jun Gao, Yanguo Wang, Ou Wu and Stephen Maybank, "Online Adaboost-Based Parameterized Methods for Dynamic Distributed

Network Intrusion Detection" IEEE Transactions On Cybernetics, Vol. 44, No. 1, January 2014

[2] W. M. Hu, W. Hu, and S. Maybank, "Adaboost-based algorithm for network intrusion detection," IEEE Trans. Syst., Man, Cybernetics, Part B: Cybernetics, vol. 38, no. 2, pp. 577–583, Apr. 2008.

[3] S. Parthasarathy, A. Ghoting, and M. E. Otey, "A survey of distributed mining of data streams," in Data Streams: Models and Algorithms. C. C. Aggarwal (Ed.) New York: Springer, Nov. 2006.

[4] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. pp. 1942–1948.doi:10.1109/ICNN.1995.488968W.-K.Chen,LinearNetworksandSystems.Belmont, CA:Wadsworth, 1993, pp. 123–135.

[5] Y. Shi and R. C. Eberhart,"A modified particle swarm optimizer," in Proc. IEEE Int. Conf. Evolut. Comput., 1998, pp. 69–73.

[6] D. Denning, "An intrusion detection model," IEEE Trans. Softw. Eng., vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[7] J. B. D. Caberera, B. Ravichandran, and R. K. Mehra, "Statistical traffic modeling for network intrusion detection," in Proc. Modeling, Anal. Simul. Comput. Telecommun. Syst., 2000, pp. 466–473.

[8] MahbodTavallaee, EbrahimBagheri, Wei Lu, and Ali A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set". Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009). ISBN 978-1-4244-3764-1/09.

[9] Douglas Reynolds "Gaussian Mixture Models∗", MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02140, USA dar@ll.mit.edu

[10] Kailas S. Elekar M.E. IT (IInd), Department of Computer Engineering Dattakala Faculty of Engineering, Swami Chincholi, Daund, Pune, India ekailas@gmail.com

[11] Prof. M.M. Waghmare, Assistant Proffesor Department of Computer Engineering Dattakala Faculty of Engineering, Swami Chincholi, Daund, Pune, India monawaghmare25@gmail.com, "Effective Intrusion Detection System using Combination of Data Mining Techniques"

[12] Z. Zhang and H. Shen, ―Online training of SVMs for real-time intrusion detection,‖ in Proc. Adv. Inform. Netw. Appl., vol. 2, 2004, pp. 568–573.

[13] H. Lee, Y. Chung, and D. Park, ―An adaptive intrusion detection algorithm based on clustering and kernelmethod,‖ in Proc. Int. Conf.Adv. Inform. Networking Appl., 2004, pp. 603–610.

[14] W. Lee and S. J. Stolfo, ―A framework for constructing features and models for intrusion detection systems,‖ ACM Trans. Inform. Syst.Security, vol. 3, no. 4, pp. 227– 261, Nov. 2000.

[15] A. Fern and R. Givan, ―Online ensemble learning: An empirical study,‖ in Proc. Int. Conf. Mach. Learning, 2000, pp. 279–286.

[16] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, ―On combining classifiers,‖ IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 3, pp.226–238, Mar. 1998.