Adaboost Ensemble with Genetic Algorithm Post Optimization for Intrusion Detection

Hany M. Harb[1] and Abeer S. Desuky[2]

[1]Computers and Systems Engineering Dept., Faculty of Eng., Azhar University
Cairo, Egypt

[2]Mathematics Dept., Faculty of Science, Azhar University
Cairo, Egypt

Abstract

This paper presents a fast learning algorithm using AdaBoost ensemble with simple genetic algorithms (GAs) for intrusion detection systems. Unlike traditional approaches using AdaBoost algorithms, it proposed a Genetic Algorithm post optimization procedure for the found classifiers and their coefficients removing the redundancy classifiers which cause higher error rates and leading to shorter final classifiers and a speedup of classification. This approach has been implemented and tested on the NSL-KDD dataset and its experimental results show that the method reduces the complexity of computation, while maintaining the high detection accuracy. Moreover, the method improves the processing time, so it is especially appealing for the real-time processing of the intrusion detection system.

Keywords: *Intrusion Detection; AdaBoost; Genetic Algorithm; Feature Selection; Classification; NSL-KDD dataset.*

1. Introduction

With the development of the Internet, the information security threat is becoming one of the most crucial problems. Reliable connections, information integrity and privacy on computer systems and networks are demanded more intensively nowadays than ever before. Therefore, intrusion detection systems (IDS) are used to monitor systems during their lifetime and used to detect possible attacks against them.

IDS are usually classified in two categories: misuse and anomaly [1]. A misuse or knowledge-based IDS aims at detecting the occurrence of states or action sequences that have been previously identified to be an intrusion. Thus, in this kind of IDS, attacks must be known and described a priori and IDS are usually unable to deal with new or unknown attacks. Alternatively, an anomaly or behavior-based IDS assumes that an intrusion can be detected by observing deviations from a normal or expected behavior of a monitored entity. The valid behavior is extracted from previous reference information about the system. The IDS later compares the extracted model with the current activity and raises an alert each time that a certain degree of divergence from the original model is observed.

An effective Intrusion Detection system needs to limit false positives—incorrectly identifying an attack when there is none. At the same time it needs to be effective at catching attacks. False alarms are distracting and so they reduce the effectiveness of an Intrusion Detection system. Hence, the basic target of IDS research is reducing the false positives rate while maintaining high detection rate.

One of the main problems with IDS is the overhead, which can become prohibitively high. The detection response time is another major problem in the IDS. Computer networks have a dynamic nature in a sense that information and data within them are continuously changing. Therefore, detecting an intrusion accurately and promptly is crucial especially in real time IDS.

Most of the research works are aimed to introduce the most time efficient methodologies [2]. Our research goal is to improve the AdaBoost (adaptive boosting) algorithm to be suitable for the real time implementation.

In this paper we tried to construct an intrusion detection approach with a low computational complexity, a high detection accuracy, and efficiency in the real time implementation. In this correspondence, we apply the AdaBoost algorithm with Genetic algorithm to intrusion detection.

2. Related work

M. Panda and M. R. Patra [3] reported a work on the subject of intrusion detection for the anomaly detection. Authors in this paper have proposed a framework of NIDS based on Naïve Bayes algorithm. When compared their approach to the neural network based approach, their approach achieved higher detection rate which is about 95%.

S. Teng et al. in [4] proposed cooperative network intrusion detection Based on Fuzzy SVMs. They firstly preprocessed the data. Then the fuzzy membership function is introduced into SVM. Three types of detecting agents are generated according to TCP, UDP and ICMP protocol. Finally, using the KDD CUP 1999 data set, a comparison between single FSVM and multi FSVM showed that, with cooperative detection based on multi FSVMs, the accuracy rate is 91.2101%, and with single FSVM, the accuracy rate is 82.5682%.

In [5], Bankovic´ Z et al., presented an improvement to network security using genetic algorithm approach. In this work they have realized a misuse detection system based on genetic algorithm (GA) approach. To be able to process network data in real time, they have deployed principal component analysis (PCA) to extract the most important features of the data and then to keep the high level of detection rates of attacks while speeding up the processing of the data.

In [6], Weiming Hu et al. proposed an intrusion detection algorithm based on the AdaBoost algorithm. In the algorithm, decision stumps are used as weak classifiers. Their results showed 90.88% detection rate and 1.79 % false alarm rate and in comparison with other algorithms, the algorithm gave lower computational complexity and error rates.

3. The Proposed Method: GA-AdaBoost

Ideally, an IDS should be fast, simple, and accurate, while at the same time being complete. It should detect all attacks with little performance penalty. In our Work we aim at constructing a fast intrusion detection approach with a low computational complexity to be suitable for use in real time applications. To keep a high accuracy, in this correspondence, we apply the AdaBoost with Genetic algorithm to intrusion detection. The motivations for applying the AdaBoost with Genetic algorithm are as it follows:

- The AdaBoost algorithm is one of the most popular machine learning algorithms. It has been applied to many pattern recognition problems, such as the face recognition. However, the application of the AdaBoost algorithm to intrusion detection has not been explored so far [6].

- AdaBoost is a sequential forward search procedure using the greedy selection strategy. Because of its greedy character, neither the found weak classifiers nor their coefficients are optimal [7]. Genetic Algorithm, proposed as a post optimization procedure for the found classifiers and their coefficients, removes the redundancy classifiers and leads to shorter final classifiers and a speedup of classification.

To process network data in real time and perform efficient intrusion detection, we need to extract the most important pieces of information that can be deployed for efficient detection of network attacks. We have deployed an alternative way proposed in [8] to reduce the dimension of the used data. According to the obtained results, we have selected sixteen features out of forty one used to describe each connection of KDD dataset [14, 15]. Our results confirm maintenance of high accuracy while using lower dimension of data. The other benefit is that data processing and the decision making whether a connection is an attack are performed much faster.

3.1 AdaBoost

The AdaBoost algorithm is a kind of boosting algorithms which were proposed by Freund and Schapire [9]. Fig. 1 is a generalized version of the AdaBoost algorithm for binary classification problems.

Given:
$S=\{(x_1, y_1),(x_2, y_2),...,(x_l, y_l)\}$ With $x_i \in X$
And $y_i \in \{-1, +1\}$.

Initialize the distribution:
$D_l^{(i)} = 1/l,\ i = 1,\ 2,\ ...,l.$
For t = 1, 2, …, T:
Train the weak learner using the distribution
$D_t^{(i)}$, $i = 1,\ 2,\ ...,l.$
Get the weak hypothesis $c_t : X \to R.$
Update:
$D_{t+1}^{(i)} = D_t^{(i)} \exp(-\alpha_t y_i c_t(x_i))/Z_t,\ i = 1,\ 2,\ ...,\ l,$
where Z_t is a normalization factor
($D_t^{(i)}$ is still a distribution)
and $\alpha_t = \frac{1}{2} \ln\frac{1-\varepsilon_t}{\varepsilon_t}$ with
$\varepsilon_t = \sum_{i=1}^{l} D_t^{(i)}[y_i \neq c_t(x_i)].$
Output the final hypothesis:
$C(X) = sign(\sum_{t=1}^{T} \alpha_t c_t(X)).$

Fig. 1 A generalized version of the AdaBoost algorithm.

AdaBoost algorithm is used to boost the classification performance of a weak learner. It does this by combining a collection of weak classification functions to form a stronger classifier. AdaBoost combines iteratively the weak classifiers by taking into account a weight distribution on the training samples such that more weight is attributed to samples misclassified by the previous iterations. The final strong classifier takes the form of a perceptron, a weighted combination of weak classifiers followed by a threshold. As in Fig. 1 the algorithm takes as input a training set (x_1, y_1), (x_2, y_2), …, (x_l, y_l) where each x_i belongs to some domain or instance space X, and each label y_i is in some label set Y. We iterate T rounds of AdaBoost training where T is the number of weak classifiers c_t and ensemble weights α_t are yielded by learning to constitute the final strong classifiers [9] [10].

The weak classifier is the core of an AdaBoost algorithm. In our AdaBoost-based algorithm for intrusion detection, classification and regression tree (CART) algorithm - proposed by Breiman et al. [11] - is used as weak classifiers.

3.2 Genetic Algorithm Overview

Genetic algorithms (GA) are search algorithms based on the principles of natural selection and genetics. The bases of genetic algorithm approach are given by Holland [12] and it has been deployed to solve wide range of problems.

GA operates on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to the solution of the problem that GA is trying to solve. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness value in the problem domain and breeding them together using the operators borrowed from the genetic process performed in nature, i.e. crossover and mutation. This process leads to the evolution of populations of individuals that are better adapted to their environment than the individuals that they were created from, just as it happens in natural adaptation [13].

3.3 Overview of GA-AdaBoost Algorithm

According to the model of the boosted classifier $C(X) = sign(\sum_{t=1}^{T} \alpha_t c_t(X))$, the final strong classifier could be regarded as the weight combination of weak classifiers $\{c_1, c_2, \ldots, c_T\}$.

Each weak classifier c_t will be determined after the boosting training.

As AdaBoost is a sequential forward search procedure using the greedy selection strategy, neither the found weak classifiers nor their coefficients are optimal. Moreover, the strong classifier comprises more weak classifiers requiring more time to evaluate and more memory to occupy. This affects the performance of IDS and slows down the detection process. To address this issue, we propose an approach based on Genetic Algorithm. This algorithm selects some weak classifiers to constitute an ensemble (final strong classifier). The weak classifiers are selected according to some evolving weights that could characterize the fitness of the included weak classifiers in the ensemble. The approach procedure is summarized in Fig. 2.

Given:

$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$ With $x_i \in X$

And $y_i \in \{-1, +1\}$.

Initialize the distribution:

$D_l^{(i)} = 1/l, \ i = 1, 2, \ldots, l.$

For t = 1, 2, …, T:

Train the weak learner using the distribution

$D_t^{(i)}, \ i = 1, 2, \ldots, l.$

Get the weak hypothesis $g_t : X \to R.$

Update:

$D_{t+1}^{(i)} = D_t^{(i)} \exp(-\alpha_t y_i g_t(x_i)) / Z_t, \ i = 1, 2, \ldots, l,$

where Z_t is a normalization factor ($D_t^{(i)}$ is still a distribution)

and $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$ with

$\varepsilon_t = \sum_{i=1}^{l} D_t^{(i)} [y_i \neq g_t(x_i)].$

Output the final hypothesis:

$G(X) = sign(\sum_{t=1}^{T} \alpha_t G_t(X)).$

- Generate a population of bit strings b.
- Evolve population where the fitness of each individual is measured as

$f(b) = w_1 * (1 - l/T) + w_2 * (1/E_b^s)$

- b* is the evolved best individual.
- Use vector b* to update -reduce- the classifiers with their weights.
- Generate a new population of reduced weight vectors w.
- Evolve population where the fitness of each individual is measured as $f(w) = 1/E_w^s$
- w* is the evolved best individual.

Outbut: ensamble G*

$G^*(X) = sign(\sum^{T} w^* G_t(X))$

Fig. 2 Post-Optimization procedure of a given boosted strong classifier based on Genetic algorithms.

The proposed Algorithm is depicted in Fig. 3 which consists of the following three phases:

- Pre-processing and Features Extraction Phase:
  In this phase, randomly two separated training and testing data sets are selected from the NSL-KDD dataset, the symbolic features are converted into numerical ones, and the most suitable features are selected.
- AdaBoost Training and Testing Phase:
  Training subphase: In this phase AdaBoost is trained using the training dataset. It iterates T rounds of AdaBoost training, where T is the number of weak classifiers c_t and ensemble weights α_t are yielded by learning to constitute the final strong classifiers.
  Testing subphase: the performance of the system with the testing data set is measured.
- Post Optimization Procedure Based on Genetic Algorithm: This phase is implemented in two steps:

Step 1: Initialize population, each individual in the evolving population is a vector b composed of bit string (b_1, b_2, …, b_T) denoting the weak classifiers. These weak classifiers constitute the strong classifier. The b_i=1 denotes the i^{th} weak classifier appearance in the ensemble while b_i=0 denotes its absence. In order to evaluate the goodness of the individuals, a validation data set is employed. Let E_b^s be the validation error of the ensemble corresponding to the individual b on the validation set s. It is obvious that E_b^s expresses the goodness of b in the way that the smaller E_b^s is, the better b is. Furthermore, let $l = \sum_{i=1}^{T} b_i$ be the number of the selected weak classifiers, where T is the total number of weak classifiers. It is preferable that fewer weak classifiers give a correct prediction for a given object rather than more weak classifiers. Therefore, $f(b) = w_1 * (1 - l/T) + w_2 * (1/E_b^s)$ is used as the fitness function, where w1 and w2 are fitness weights that can be adjusted to balance the efficiency, mode size and accuracy of classifiers.

Step 2: In this step we use the resulted vector b to update - reduce- the number of classifiers and their corresponding weights and then a new population is initialized. Each individual in this population is a vector $w = (w_1, w_2, \ldots, w_T)$ which denotes the corresponding weights of the weak classifiers. As in step one let E_w^s be the validation error of the ensemble corresponding to the individual w on the validation set s and $f(w) = 1/E_w^s$ is used as the fitness function. The procedure is summarized in Fig. 2.

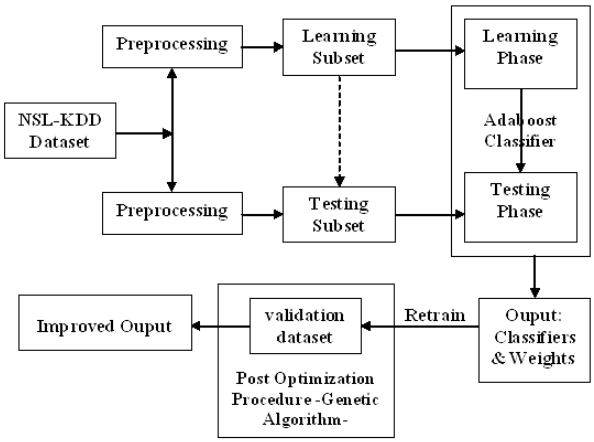

Fig. 3 The Proposed Model Structure

4. Experimental Results

We have run our experiments using Matlab 7, on a system with a 1.86GHZ Intel(R) Celeron(R) M processor and 512MB of RAM running Microsoft Windows XP Professional (SP2).

In our experiment, we used NSL-KDD data set [14]. It has solved some of the inherent problems of the KDDcup99 [15]. It is considered as a standard benchmark for intrusion detection algorithms evaluation. The training dataset of NSL-KDD is similar to KDDcup99 consisting of approximately 1,074,992 single connection vectors each of which contains 41 features. We used NSL-KDD full training dataset that contains 25192 connection records with one target value or labeled class either normal or attack.

The AdaBoost algorithm and the weak learner employed by our algorithm is implemented by the GML AdaBoost Matlab Toolbox developed by Alexander Vezhnevets [16]. The parameters used for evolution were: 0.05 crossover rate and 0.05 mutation rate, and the population was initialized randomly. In our algorithm, the GA terminates if there is no found better individual within the next 100 generations or if the validation error is equal to zero. From the experiment results shown in Table 1, we can see that the numbers of weak classifiers of the boosted strong classifier trained by standard AdaBoost are reduced by about 42% due to the Genetic Algorithms

optimization. Moreover, as the weak classifiers reduction, the average classification time (measured in seconds) of the boosted strong classifier with GA-optimization is about 600% faster. The accuracy of the classification is also slightly increased by 0.64, as also shown in Fig. 4 and Fig. 5. The results along with the comparison to other existing methods using NSL-KDD Dataset are shown in Table 2.

Table 1 Comparison of The Boosted Classifiers Without and With Post-Optimization

AdaBoost classifiers	*Our classifiers*	*AdaBoost time*	*Our time*	*AdaBoost accuracy*	*Our accuracy*
20	17	6.81	0.08	97.00	97.89
40	23	7.19	0.13	98.36	98.59
60	40	6.08	0.20	98.48	99.02
80	46	6.69	0.22	98.97	99.14
100	67	6.94	0.27	99.21	99.32
120	71	6.77	0.30	99.41	99.48
140	89	6.91	0.44	99.48	99.38
160	81	6.52	0.45	99.51	99.55
180	104	7.02	0.52	99.54	99.44
200	109	7.39	0.50	99.53	99.57
Average		6.83	0.31	98.95	99.14

Table 2 Detection Accuracy Comparison of Machine learning algorithms using NSL-KDD Dataset

Classifier	*Detection Accuracy(%)*	*Classification time(seconds)*
Discriminative Multinomial Naïve Bayes [17]	96.5	1.11
Discriminative Multinomial Naïve Bayes +PCA [17]	94.84	118.36
AdaBoost [18]	90.31	***
Decision Trees (J48) [14]	81.05	***
AdaBoost + GA (proposed)	99.57	0.5

*** indicates data not provided by the authors in their paper.

5. Conclusion

The goal of a network-based intrusion detection system (IDS) is to distinguish the attacks on the Internet from normal use of the Internet. It is an indispensable part of the information security system. Due to the variety of network behaviors and the rapid development of attack fashions, it is necessary to develop machine-learning-based intrusion detection algorithms with high detection rates and low false-alarm rates and detection time. We have proposed an AdaBoost ensemble with Genetic algorithm for intrusion detection. The method has effectively improved results of the boosted classifier. The experimental results have demonstrated that a given boosted classifier with our post optimization based on GA (Genetic Algorithm) has fewer weak classifiers and an increase of the classification accuracy and speed which is important for real time network applications.

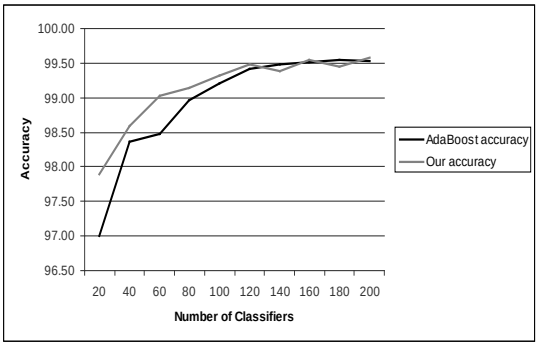

Fig. 4 AdaBoost Accuracy and Post-Optimization Accuracy.

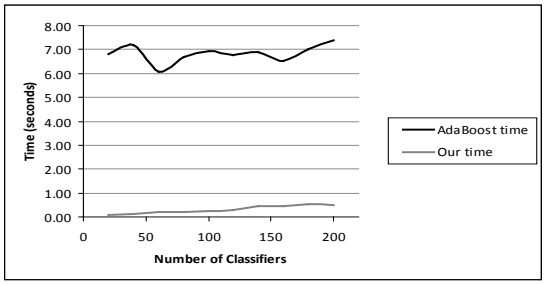

Fig. 5 AdaBoost Time and Post-Optimization Time.

References

[1] Ian Stewart, "A Modified Genetic Algorithm and Switch-Based Neural Network Model Applied to Misuse-Based Intrusion Detection", M.s. thesis, School of Computing In conformity, Queen's University, Kingston, Ontario, Canada, February 2009.

[2] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," Applied Soft Computing, vol. 10, 2010, pp. 1–35.

[3] Mrutyunjaya Panda and Manas Ranjan Patra, "Network Intrusion Detection Using Naïve Bayes", IJCSNS

International Journal of Computer Science and Network Security, VOL.7, No.12, 2007, pp. 258-263.
[4] Shaohua Teng, Hongle Du, Naiqi Wu, Wei Zhang, Jiangyi Su, "A Cooperative Network Intrusion Detection Based on Fuzzy SVMs", Journal Of Networks, VOL. 5, NO. 4, 2010, pp. 475-483.
[5] Bankovic, Z., Moya, José M., Araujo, A., Bojanic, S., and Nieto-Taladriz, "Improving network security using genetic algorithm approach", Computers & Electrical Engineering, Vol.33, Issue 5-6, 2007, pp. 438-451.
[6] Weiming Hu, Wei Hu, and Steve Maybank, "AdaBoost-Based Algorithm for Network Intrusion Detection", IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 38, No. 2, 2008, pp. 577-583.
[7] Zhang Z, Li S Z, Zhang H, "Real-time multi-view face detection", Fifth IEEE International Conference on Automatic Face and Gesture Recognition, Washington D.C, USA, 2002, pp. 142-147.
[8] Hany M. Harb, Afaf A. Zaghrot, Mohamed A. Gomaa and Abeer S. Desuky, "Selecting Optimal Subset of Features for Intrusion Detection Systems", Advances in Computational Sciences and echnology, Research India Publications, Volume 4 Number 2, 2011, pp. 179-192.
[9] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” in Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, 1996, pp. 148-156.
[10] Zhou Z-H, WU J, Tang W, "Ensembling neural networks: many could be better than all", Artificial Intelligence, 2002, pp. 239-263.
[11] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and Regression Trees", Chapman and Hall, New York, USA, 1984.
[12] Holland J. "Adaptation in natural and artificial system. Ann Arbor", The University of Michigan Press, 1975.
[13] B. Abdullah, I. Abd-alghafar, Gouda I. Salama and A. Abd-alhafez, " Performance Evaluation of a Genetic Algorithm Based Approach to Network Intrusion Detection System", 13th International Conference on Aerospace Sciences & Aviation Technology, Asat- 13, May 26 – 28, 2009, pp. 1-17.
[14] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali, A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", proceeding of IEEE symposium on computational Intelligence in security and defence application, 2009.
[15] KDD Cup 1999 Data Set, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.htm
[16] A. Vezhnevets, GML AdaBoost Matlab Toolbox [http://research. graphicon.ru/], Graphics and Media Laboratory, Computer Science Department, Moscow State University, Moscow, Russian Federation.2006.
[17] Panda, M., Abraham, A., Patra, M.R., "Discriminative multinomial Naïve Bayes for network intrusion detection", Information Assurance and Security (IAS), 2010 Sixth International Conference, 2010, pp. 5-10.
[18] V.P. Kshirsagar and Dharmaraj R. Patil, "Application of Variant of AdaBoost based Machine Learning Algorithm in Network Intrusion Detection", International Journal of Computer Science and Security (IJCSS), Vol. 4, Issue.2, 2010, pp. 1-6.