Data Reduction and Ensemble Classifiers in Intrusion Detection

Anazida Zainal, Mohd Aizaini Maarof and Siti Mariyam Shamsuddin
Faculty of Computer Science and Information Systems,
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
anazida@utm.my, aizaini@utm.my and mariyam@utm.my

Abstract

Efficiency is one of the major issues in intrusion detection. Inefficiency is often attributed to high overhead and this is caused by several reasons. Among them are continuous detection and the use of full feature set to look for intrusive patterns in the network packet. The purpose of this paper are; to address the issue of continuous detection by introducing traffic monitoring mechanism and a lengthy detection process by selectively choose significant features to represent a network connection. In traffic monitoring, a new recognition paradigm is proposed in which it minimizes unnecessary recognition. Therefore, the purpose of traffic monitoring is two-folds; to reduce amount of data to be recognized and to avoid unnecessary recognition. Empirical results show 30 to 40 percent reduction of normal connections is achieved in DARPA KDDCup 1999 datasets. Finally we assembled Adaptive Neural Fuzzy Inference System and Linear Genetic Programming to form an ensemble classifiers. Classification results showed a small improvement using the ensemble approach for DoS and R2L classes.

1. Introduction

The preventive measure such as firewall, authentication and cryptography are often insufficient to safeguard the network. Thus, intrusion detection system has become a very important defense mechanism to address the vulnerabilities exposed in a computer network. Intrusion detection is classified into two types: misuse detection and anomaly detection. Misuse detection uses well defined patterns known as signatures of the attacks. Meanwhile, anomaly detection builds a normal profile and anomalous traffic is detected when the deviation from the normal model reaches a preset threshold. This paper concerns issue of data reduction and accuracy in intrusion detection. The former is achieved by imposing traffic monitoring and feature selection. Meanwhile, the latter is achieved using ensemble classifiers.

This paper is organized as follows: Section 2 gives an overview of related works in the area of features selection, Section 3 presents ensemble approach and related works. In Section 4 we present the experiments, results followed by some analysis and discussion. Finally, Section 5 concludes the study presented in this paper and some future works that will be pursued.

2. Data Reduction and Related Works

Data reduction can be achieved by filtering, data clustering and feature selection [1]. Generally, the capability of an anomaly intrusion detection is often hinders by inability to accurately classify variation of normal behavior as an intrusion. Additionally, network traffic data is huge and it causes a prohibitively high overhead and often becomes a major problem in IDS [2]. According to [3], the existence of these irrelevant and redundant features generally affects the performance of machine learning or pattern classification algorithms. [4] proved that proper selection of feature set has resulted in better classification performance. [5] have demonstrated that the elimination of these unimportant and irrelevant features did not significantly lowering the performance of IDS. [1] tackled the issue of effectiveness of an IDS in terms of real-time and detection accuracy from the feature reduction perspective. In their work, features were reduced using two techniques, Bayesian Network (BN) and Classification and Regression Trees (CART). They have experimented using four sets of feature subset which are 12, 17, 19 and all the variables (41) from one network connection. Data used was KDD cup 99. The work suggested no generic feature subset instead different features with different length were proven to be good for different types of attack. Their work also highlighted the need to implement ensemble classifiers for better accuracy. Details of their findings can be found in [1]. Meanwhile [6] used decision tree

978-0-7695-3136-6/08 $25.00 © 2008 IEEE
DOI 10.1109/AMS.2008.146

approach to selectively choose the features and they came up with 12 features. It can be concluded that feature selection is an important preprocessing task which can eliminate noise and contribute to better detection.

2.1. Shewhart Control Chart

Traffic monitoring is aimed at reducing the data that needs to be recognized by the recognizer and to eliminate unnecessary recognition. Related work pertaining to data reduction was done by Kwok et al [7]. The aim was to reduce the data volume that needs to be recognized in order to improve efficiency of the recognition process. They extracted the components that best characterize user behavior particularly user login session. Besides this work, most of the feature selection works discussed in Section 2 were also meant to reduce the data besides improve the accuracy of a detection. Our approach to traffic monitoring focused on filtering aspect where special type of normal activity will be filtered out. Besides reducing the amount of data, this approach will also avoid unnecessary recognition. Descriptive statistics and the method of confidence interval were used. Training data was plotted to determine the mean ($\bar{x}$) and standard deviation (σ). We took 3σ limit as a confidence interval. This concept is originated from Shewhart Control Chart where Upper Control Limit (UCL) and Lower Control Limit (LCL) are defined as:

$$\text{UCL} = \bar{x} + 3\sigma \quad \textbf{(1)}$$

$$\text{LCL} = \bar{x} + 3\sigma \quad \textbf{(2)}$$

Shewhart control chart is a statistical approach to the study of manufacturing process variation for the purpose of improving the economic effectiveness of the process. These methods are based on continuous monitoring of process variation. It is usually applied to detect large abrupt changes in the monitored variable [8].

3. Ensemble Approach and Related Work

According to [1], an important advantage for combining redundant and complementary classifiers is to increase robustness, accuracy and better overall generalization. [9] have demonstrated the use of ensemble classifiers gave the best accuracy for each category of attack patterns. In designing a classifier, their first step was to carefully construct different connectional models to achieve best generalization performance for classifiers. [1] proposed CART-BN approach, where CART performed best for *Normal*, *Probe* and *U2R* and the ensemble approach worked best for *R2L* and *DoS*. Meanwhile, [6] proved that ensemble Decision Tree was suitable for *Normal*, LGP for *Probe*, *DoS* and *R2L* and Fuzzy classifier was for *R2L*. Later, [10] demonstrated the ability of their proposed ensemble structure in modeling light-weight distributed IDS. Meanwhile, [11] proposed three variants of Neural Networks, SVM and MARS as components in their IDS. This combining approach has demonstrated better performance when compared to single classifier approach. Here, we have chosen two soft computing techniques to develop our classifiers and they are: Linear Genetic Programming and Adaptive Neural Fuzzy Inference.

3.1. Adaptive Neural Fuzzy Inference System

Due to complex relationships that exist between the features and the nature of the traffic data which has the grey boundary between normal and intrusive, fuzzy inference system is among the recent approaches which were deployed in intrusion detection. Similar to [12], we deployed ANFIS due to difficulty in determining the parameters associated with variations in the data values to the chosen membership function. ANFIS is the hybrid of approximate reasoning method with the learning capabilities of neural network. In ANFIS, the learning mechanism is implemented in feed-forward supervised approach.

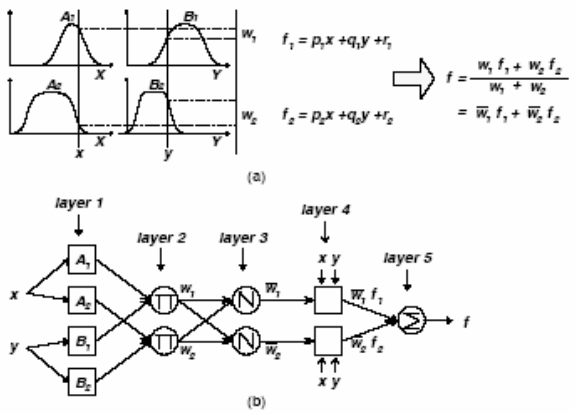

Figure 1. (a) Sugeno Fuzzy Reasoning; (b) equivalent ANFIS structure [13]

The square and circle nodes are for adaptive nodes with parameters and fixed nodes without parameters, respectively. The first layer consists of square nodes that perform fuzzification with chosen membership function. The parameters in this layer are called premise parameters. In the second layer T-norm operation is performed to produce the firing strength of each rule. The ratio of *i*th rule of the firing strength to the sum of all rules' firing strength is calculated in the third layer, generating the normalized firing strengths.

The fourth layer consists of square nodes that perform multiplication of normalized firing strengths with the corresponding rule. The parameters in this layer are called consequent parameters. The overall output is calculated by the sum of all incoming signals in the fifth layer [13].

Extensive work on IDS using fuzzy was done by [12]. They developed five parallel ANFIS binary classifiers. Limitation of the learning to 50 epochs may be due to slow performance since they used full features (41). FIS was used to classify whether the input was normal or intrusive based on 5 outputs from ANFIS; Normal, Probe, DoS, U2R and R2L. Finally, Genetic Algorithm (GA) was used to optimize the structure of their fuzzy decision engine. Different learning flavor of fuzzy was deployed by [14] where GA based learning was adopted and their experiment was to discriminate between normal and attack.

3.2. Linear Genetic Programming

Recent developments in GP, which include increased speed through use of linear genomes constructed of machine code instructions and development of homologous crossover operators have motivated the study in network security issues [15].

Genetic programming is a technique to automatically discover computer programs using principles of Darwinian evolution [16]. It can create a working computer program from a high-level problem statement of the problem and breeds a population of programs to solve a problem. GP iteratively transforms a population of computer programs into a new generation of program by applying genetic operations. These genetic operations include crossover, mutation, reproduction, gene duplication and gene deletion [16]. The fitness of the resulting solutions are evaluated and suitable selection strategy is then applied to determine which solutions will be maintained into the next generation [10].

Linear genetic programming is a variant of the GP technique which uses a specific linear representation of computer programs. Its main characteristics in comparison to tree-based GP lies in that the evolvable units are not expressions of a functional programming language (like LISP), but the programs of an imperative language (like c/c++) and the basic evolvable unit is a native machine code instruction that runs on the floating point processor unit (FPU) [10].

[10] demonstrated the capability of three GP variants in the application of IDS where Multi Expression Programming (MEP) outperformed the rest in 3 cases except Probe and DoS. It also came up with very few discriminative features (3, 4, 6, 2 and 7) in which its classification score is above 95% in all cases. Meanwhile [15] claimed that GP could be executed in realtime due to its detection speed and high level of accuracy. LGP could outperform SVM and ANN in terms of detection accuracy if the population size, program size, crossover rate and mutation rate are appropriately chosen [9].

4. Experiment Setup and Results

We obtained the data for our experiments from 1998 DARPA intrusion detection evaluation program and was prepared by MIT Lincoln Labs. For each TCP/IP connection, 41 attributes were associated to it plus one label. Table **1** shows the attributes and their types. Attacks fall under four main categories (KDDCup 1999)[11]:

i) **DoS**: Denial of Service (DoS) is where an attacker makes a computing or memory resource too busy or too full to handle legitimate requests, thus denying legitimate users access to a machine.

ii) **R2L**: A remote to user (R2L) attack is where an attacker sends packets to a machine over a network, then exploits the machine's vulnerability to illegally gain local access as a user.

Table 1. Attributes for intrusion detection dataset

No	Variable Name	No	Variable Name
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_byte	26	srv_serror_rate
6	dst_byte	27	rerror_rate
7	land	28	srv_rerror_rate
8	wrong_fragment	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_host_rate
11	num_failed_login	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_srv_serror_rate
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_host_login		

iii) **U2R**: User to root (U2R) exploits are where an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system.

iv) **Probing**: Surveillance and Other Probing is a class of attack where an attacker scans a network to gather information to find known vulnerabilities and use the information to look for exploits.

4.1. Experiments

Experiments presented in this paper are of supervised training and its flow is depicted in Figure **2.**

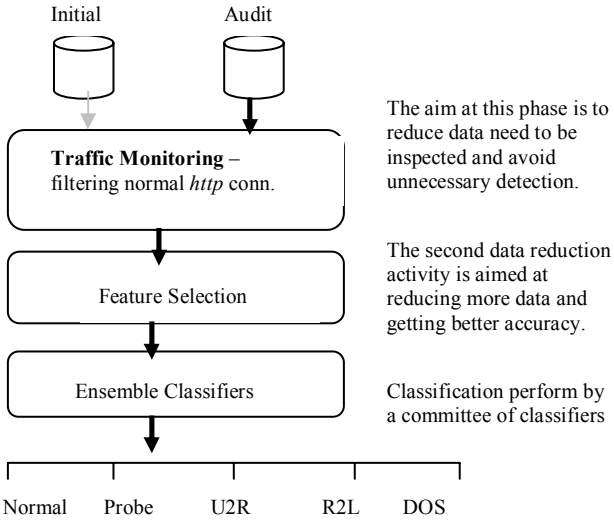

Figure 2. Experimental Flow

The process to obtain important features was done offline using Initial Audit Data. Similarly, our Traffic Monitoring process also used the same Initial Audit Data to obtain the appropriate values for mean, standard deviation, upper control limit and lower control limit in establishing the confidence interval. Training data was presented to the committee of classifiers. This training dataset has five classes and they are *Normal*, *U2R*, *R2L*, *Probe* and *DoS*.

4.2. Results and Analysis

The performance of our filter was tested on both datasets, known and unknown attacks. Each contains 494,020 and 311,030 data respectively. Nominal data were converted to numerical values between [0 1] and other numerical values data were scaled to [0 1] as well. Since *http* dominated most of the *Normal* connection, we focused on filtering out the *Normal* data with *http* service (f3) and flag was *SF* (f4) with constant rate for features f25, f27, f28, f34, f35, f37 and f40. The mean and standard deviation values obtained were:

i) Mean = 1.455 iii) Standard Deviation = 0.0205
ii) UCL = 1.5314 iv) LCL = 1.4550

The value for LCL has to be modified and estimated at 1.455 because *Normal* data used in this study is not symmetric instead it is skewed. Any input data that fall between 1.5314 and 1.4550 will be filtered out thus the classifier will not be invoked. In other words, unnecessary recognition will be avoided. Figure 3 shows example graphs show two attacks were plotted onto a Shewhart Chart. Table 4 below shows the reduction percentage of data that needs to be examined by the classifier.

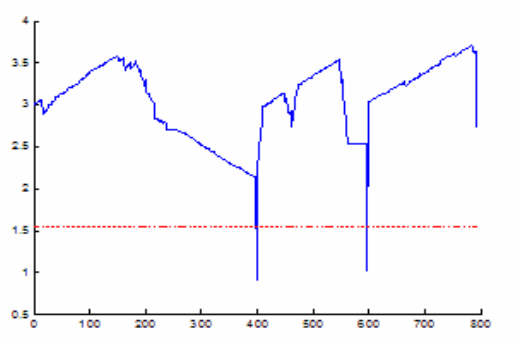

Figure 3 (a) : Distribution of *apache2* (an example of DoS attack)

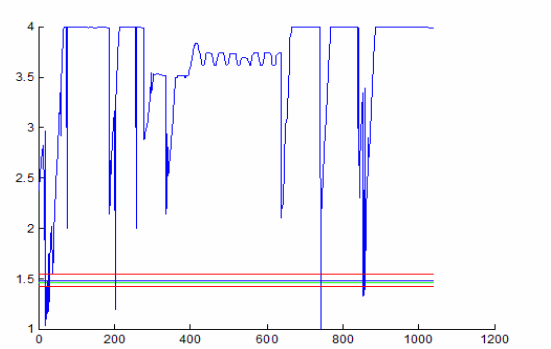

Figure 3 (b) : Distribution of *portsweep* (an example of Probe attack)

The finding shows that with traffic monitoring in place, the size of *Normal* data can be reduced to 30 to 40 percent. In real environment, this reduction percentage may become more significant since most of the time, *Normal* data usually overpopulate the whole dataset and this is general assumption used by unsupervised approach.

Table 4: Performance of Monitoring using Shewhart method

Dataset	Attack			Normal (reduction)		
	Original data size	Wrongly filtered	Correctly filtered	Original data size	After filtering	Reduction
Known Attack	395,587 (100%)	988 (0.25%)	99.75%	97,277 (100%)	65,983 (67.8%)	32.2%
Unknown Attack	250,436 (100%)	230 (0.09%)	99.91%	60,594 100%	36,807 (60.74%)	39.3%

Besides, it also indicates that the classifier will not be burden with unnecessary recognition. The finding also shows that the proposed method has a potential to be used as a filter due to its nature that can detect abrupt change. This finding confirms [17] who implemented CUSUM for monitoring process variability. Meanwhile, 0.25% of wrongly filtered data for known attack is considered acceptable from the statistical process control literature which stated that +/-3σ monitoring would have false alarm of 0.27% [18]. For the subsequent experiments, we used the training data and test data comprises of 5,092 and 6,890 records respectively. Rough-Discrete Particle Swarm Optimization was used to selectively choose significant features. Detail procedure of feature selection can be found in [19]. Below were the reduced features obtained:

1. Normal (8 features) :
   f12, f31, f32, f33, f35, f36, f37 and f41
2. Probe (6 features) : f2, f3, f23, f34, f36 and f40
3. DoS (8 features) :
   f5, f10, f24, f29, f33, f34, f38 and f40
4. U2R (6 features) : f3, f4, f6, f14, f17 and f22
5. R2L (6 features) : f3, f4, f10, f23, f33 and f36

Meanwhile, the *neuro-fuzzy* classifier was trained at 300 epochs of learning and two membership functions (MF) in the form of Bell-shape were selected for the input and output fuzzy sets. Five ANFIS were produced, one for each class. As for LGP, we used population size of 2048 and below, mutation rate in the range of 96.7% to 78.1% and crossover rate of 71.7% to 30.1%. Table 5 shows performance of each classifier.

LGP outperforms in most of the classes in terms of accuracy. Meanwhile, performance of ANFIS and LGP are almost equivalent particularly in the class of *U2R* and *R2L*. Our ensemble classifier model which consists of Adaptive Neural Fuzzy Inference System (ANFIS) and Linear Genetic Programming (LGP) was constructed in the following manner. First, we developed ANFIS and LGP models using the reduced features individually to obtain a good generalization performance. The final outputs of our ensemble model were decided as follows: each classifier would output the strength of their decision and we took the average decision values. Last column in Table 5 illustrates the best performance. From the results, we can see that ensemble model produced better accuracy and TP rate for both *DoS* and *R2L* classes. LGP performs best in *Normal*, *Probe* and *U2R*. This improvement is due to the nature of ensemble approach where it exploits the differences in misclassification (by individual models) and improves overall performance [1].

5. Conclusion

In this paper we have demonstrated an improved classification to intrusion detection problem by performing two layer data reduction and ensemble classifiers. The former was addressed by introducing traffic monitoring and a hybrid feature selection approach. Their performance was evaluated on the DARPA benchmark data. We have also demonstrated the capability of each individual classifier.

Table 5: Performance of two individual classifiers – ANFIS and LGP

Classes	**LGP**			**ANFIS**			**Best Performance**		
	Accuracy	FP	TP	Accuracy	FP	TP	Accuracy	FP	TP
Normal	98.83	0.0029	99.71	96.314	0.0029	96.314	98.83	0.0029	99.71
Probe	99.68	0.0000	99.86	95.414	0.0000	55.570	99.68	0.0000	99.86
DoS	97.45	0.0000	97.43	92.656	0.0007	88.770	**97.56**	0.0000	**97.62**
U2R	99.91	0.0000	80.00	99.768	0.0000	44.000	99.91	0.0000	80.00
R2L	99.63	0.0000	98.58	99.492	0.0000	95.027	**99.79**	0.0000	**99.70**

LGP is better than ANFIS in terms of detection accuracy performance based on the dataset used in our experiment. We have also proven that even though a single classifier (LGP) can capture most of the normal and attack patterns, its performance in *DoS* and *R2L* can be improved with the deployment of ensemble model. We will focus on embedding the adaptation ability into our IDS model with the aim to further reduce the false alarm rate in our future work. We will also explore and consider other control charts approach that can increase the reduction percentage of Normal data and decrease the number of wrongly filtered data.

Acknowledgement

Authors would like to thank Ministry of Higher Education, Ministry of Science Technology and Innovation and Universiti Teknologi Malaysia for sponsoring this study. Special thank goes to Professor Ajith Abraham who has helped with this work and finally we would like to thank anonymous reviewers for reviewing this paper.

References

[1] S. Chebrolu, A. Abraham and J.P. Thomas. "Feature Deduction and Ensemble Design of Intrusion Detection Systems." *International Journal of Computers and Security*, Vol 24, Issue 4, June 2005, pp. 295-307.

[2] A.H. Sung and S. Mukkamala. ''The Feature Selection and Intrusion Detection Problems.'' ASIAN 2004. LNCS, vol. 3321, Springer Hieldelberg, 2004, pp. 468–482.

[3] B. Chakraborty. ''Feature Subset Selection by Neuro-rough Hybridization.'' LNCS, Springer Hiedelberg, 2005, pp. 519–526.

[4] A. Hassan, M.S. Nabi Baksh, A.M. Shaharoun, and H. Jamaluddin. ''Improved SPC Chart Pattern Recognition Using Statistical Feature.'' *International Journal of Production Research* 41(7), 2003, pp. 1587–1603.

[5] A.H. Sung and S. Mukkamala. "Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks." In Proceedings of the Symposium on Applications and Internet (SAINT'03), pp. 209-216.

[6] A. Abraham and R. Jain. "Soft Computing Models for Network Intrusion Detection Systems." Soft Computing in Knowledge Discovery: Methods and Applications, Springer Hiedelberg, Chapter 16, 2004, 20 pages.

[7] Y.L. Kwok, L. Hui and L.C. Siu. "A Data Reduction Method for Intrusion Detection." *Journal of Systems and Software*, Vol. 33, Issue 1, 1996, pp. 101-108.

[8] P. Kang and D. Birtwhistle. "On-line Condition Monitoring of Tap Changers." In Proceedings of IEEE CIRED 2001, 18-21 June 2001.

[9] S. Mukkamala, A.H. Sung and A. Abraham, A. "Modeling Intrusion Detection Systems Using Linear Genetic Programming Approach." LNCS 3029, Springer Hiedelberg, 2004, pp. 633-642.

[10] A. Abraham, C. Grosan and C.M. Vide. "Evolutionary Design of Intrusion Detection Programs." *International Journal of Network Security*, Vol. 4, No. 3, 2007, pp. 328-339.

[11] S. Mukkamala, S., A.H. Hung and A. Abraham. "Intrusion Detection Using an Ensemble of Intelligent Paradigms." *Journal of Network and Computer Applications*, Vol. 28, 2005, pp. 167-182.

[12] A.N. Toosi, and M. Kahani. "A new approach to intrusion detection based on a evolutionary soft computing model using neuro-fuzzy classifiers." *Journal of Computer Communications*, Vol. 30, 2007, pp. 2201-2212.

[13] J.R. Jang. "ANFIS: Adaptive-Network-Based Fuzzy Inference System." *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 3, May 1993. pp. 665-685.

[14] M.S. Abadeh, J. Habibi and C. Lucas. "Intrusion Detection Using a Fuzzy Genetics-based Learning Algorithm." *Journal of Network and Computer Applications* Vol 30, 2007, pp. 414-428.

[15] J.V. Hansen, P.B. Lowry, R.D. Meservy and D.M. McDonald. "Genetic Programming for Prevention of Cyberterrorism through Dynamic and Evolving Intrusion Detection." *Journal of Decision Support Systems* Vo. 43 2007, pp. 1362-1374.

[16] J.R. Koza and R. Poli. "A Genetic Programming Tutorial."http://www.genetic-programming.com/jkpdf /burke2003tutorial.pdf

[17] A. Hassan, "Online Recognition of Developing Control Chart Patterns", PhD thesis, Univ Teknologi Malaysia, 2002.

[18] D.C. Montgomery, *Introduction to Statistical Quality Control*. 3rd. ed. New York: John Wiley & Sons, 1996.

[19] A. Zainal, M.A. Maarof and S.M. Shamsuddin, "Feature Selection Using Rough-DPSO in Anomaly Detection." LNCS 4705, Part 1 Springer Hiedelberg 2007, pp. 512-524.