International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:6, 2010

Mining Network Data for Intrusion Detection through Naïve Bayesian with Clustering

Dewan Md. Farid, Nouria Harbi, Suman Ahmmed, Md. Zahidur Rahman, and Chowdhury Mofizur Rahman

***Abstract*—**Network security attacks are the violation of information security policy that received much attention to the computational intelligence society in the last decades. Data mining has become a very useful technique for detecting network intrusions by extracting useful knowledge from large number of network data or logs. Naïve Bayesian classifier is one of the most popular data mining algorithm for classification, which provides an optimal way to predict the class of an unknown example. It has been tested that one set of probability derived from data is not good enough to have good classification rate. In this paper, we proposed a new learning algorithm for mining network logs to detect network intrusions through naïve Bayesian classifier, which first clusters the network logs into several groups based on similarity of logs, and then calculates the prior and conditional probabilities for each group of logs. For classifying a new log, the algorithm checks in which cluster the log belongs and then use that cluster's probability set to classify the new log. We tested the performance of our proposed algorithm by employing KDD99 benchmark network intrusion detection dataset, and the experimental results proved that it improves detection rates as well as reduces false positives for different types of network intrusions.

***Keywords*—**Clustering, detection rate, false positive, naïve Bayesian classifier, network intrusion detection.

I. Introduction

INTRUSION detection system (IDS) is security tools that collect information from a variety of network sources, and analyze the information for signs of network intrusions. IDS can be host-based or network-based systems [1]. Host-based IDS locates in servers to examine the internal interfaces, and network-based IDS monitors network packets to discover network intrusions. The success of an IDS can be characterized in both detection rates (DR) and false positives (FP) for different types of intrusions [2]. Ideally, IDS should have an attack detection rate of 100% along with false positive of 0%, which is really hard to achieve. Detection rate is the percentage of correctly identified true attacks by IDS, and false positive is alarm that rose by IDS for normal activities. Nowadays, data mining methods have become indispensable tools for analyzing large volume of network logs or audit data to identify the patterns of the normal behaviors and pattern of the intrusions in computer network that are useful in classifying network intrusions [3]-[7]. The main motivation of using data mining methods in intrusion detection is automation. Data mining technologies, such as decision tree (DT), naïve Bayesian classifier (NB), neural network (NN), support vector machine (SVM), k-nearest neighbors (KNN), fuzzy logic model, and genetic algorithm have been widely used to analyze network logs to gain intrusion related knowledge to improve the performance of IDS in last decades [8]-[11]. To apply data mining techniques in intrusion detection, first the collected network logs or audit data needs to be preprocessed and converted to the format that suitable for mining. Next, the reformatted data will be used to develop a clustering or classification model. Data mining provide decision support for intrusion management, and also help IDS for detecting new vulnerabilities and intrusions by discovering unknown patterns of attacks or intrusions.

Intrusion detection is a process of gathering intrusion related knowledge that occurred in the computer networks or systems and analyzing them for detecting future intrusions. Intrusion detection can be broadly divided into two categories: misuse detection and anomaly detection. Misuse detection systems detect attacks based on known attack patterns that stored in a database, while anomaly detection systems detect deviations in activity from normal profiles. Misuse detection systems use various methods including rule-based expert systems, model-based reasoning systems, state transition analysis, genetic algorithms, fuzzy logic, and keystroke monitoring. The main advantage of misuse detection system is to produce very low false positives (FP), but it requires regular updates of rules, and not capable of detecting unknown or new intrusions. On the other hand, anomaly detection systems detect deviations from normal behavior, and based on a threshold value determines if it is normal behavior or intrusion [12]. There are various approaches for anomaly detection including statistical analysis, neural networks, machine learning, and artificial immune system. The main disadvantage of anomaly detection is that it provides many false positives. A variety of IDS have been introduced in last decades, but still there some issues that should be consider in

Dewan Md. Farid is with the ERIC Laboratory, University Lumière Lyon 2 – 5 av. Pierre Mendes, France – 69676 BRON Cedex, France (phone: +33 0648882531; fax: +33 478772375; e-mail: dewanfarid@gmail.com).

Nouria Harbi is with the ERIC Laboratory, University Lumière Lyon 2– France (e-mail: nouria.harbi@univ-lyon2.fr).

Suman Ahmmed is with the University Lumière Lyon 2–France (e-mail: suman@cse.uiu.ac.bd).

Mohammad Zahidur Rahman is with the Department of Computer Science and Engineering, Jahangirnagar University, Bangladesh (e-mail: rmzahid@juniv.edu).

Chowdhury Mofizur Rahman is with the Department of Computer Science and Engineering, United International University, Bangladesh (e-mail: crmuiu@gmail.com).

the current IDS like low detection accuracy, unbalanced detection rates for different types of attacks, and high false positives. In this paper, we proposed a new learning algorithm for mining network logs to detect network intrusions through naïve Bayesian classifier, which clusters the network logs into several groups based on similarity, and then calculates the prior and conditional probabilities for each group of logs. For classifying a new log, the algorithm checks in which cluster the log belongs and then use that cluster's probability set to classify the log. We tested the performance of our proposed algorithm by employing KDD99 benchmark intrusion detection dataset that proved it improves the detection rates as well as reduces the false positives for different types of network intrusions in comparison with other existing methods.

The remainders of the paper are organized as follows. Section II presents the overview of IDS based on data mining. The proposed algorithm is introduced in Section III. In Section IV, the experimental analysis is presented. Finally, our conclusions and future works are mentioned in Section V.

II. Data Mining based IDS

A. Related works

In 1980, the concept of intrusion detection began with Anderson's seminal paper [13]; he introduced a threat classification model that develops a security monitoring surveillance system based on detecting anomalies in user behavior. In 1986, Dr. Denning proposed several models for commercial IDS development based on statistics, Markov chains, time-series, etc [14]. In the early 1980's, Stanford Research Institute (SRI) developed an Intrusion Detection Expert System (IDES) that monitors user behavior and detects suspicious events [15]. In 1988, a statistical anomaly-based IDS was proposed by Haystack [16], which used both user and group-based anomaly detection strategies. In 1996, Forrest et al. proposed an analogy between the human immune system and intrusion detection that involved analyzing a program's system call sequences to build a normal profile [17]. In 2000, Valdes et al. [18] developed an anomaly based IDS that employed naïve Bayesian network to perform intrusion detecting on traffic bursts. In 2003, Kruegel et al. [19] proposed a multisensory fusion approach using Bayesian classifier for classification and suppression of false alarms that the outputs of different IDS sensors were aggregated to produce single alarm. In the same year, Shyu et al. [20] proposed an anomaly based intrusion detection scheme using principal components analysis (PCA), where PCA was applied to reduce the dimensionality of the audit data and arrive at a classifier that is a function of the principal components. In 2003, Yeung et al. [21] proposed an anomaly based intrusion detection using hidden Markov models that computes the sample likelihood of an observed sequence using the forward or backward algorithm for identifying anomalous. Lee et al. [22] proposed classification based anomaly detection using inductive rules to characterize sequences occurring in normal data. In 2000, Dickerson at al. [23] developed the Fuzzy Intrusion Recognition Engine (FIRE) using fuzzy logic that process the network data and generate fuzzy sets for every observed feature and then the fuzzy sets are used to detect network attacks. In 2003, Ramadas et al. [24] presented the anomalous network traffic detection with self organizing maps using DNS and HTTP services that the neurons are trained with normal network traffic then real time network data is fed to the trained neurons, if the distance of the incoming network traffic is more than a preset threshold then it raises an alarm.

B. Architecture of data mining based IDS

An IDS monitors network traffic in a computer network like a network sniffer and collects network logs. Then the collected network logs are analyzed for rule violations by data mining algorithms. When any rule violation is detected, the IDS alert the network security administrator or automated intrusion prevention system (IPS). The generic architectural model of data mining based IDS is shown in Fig 1.

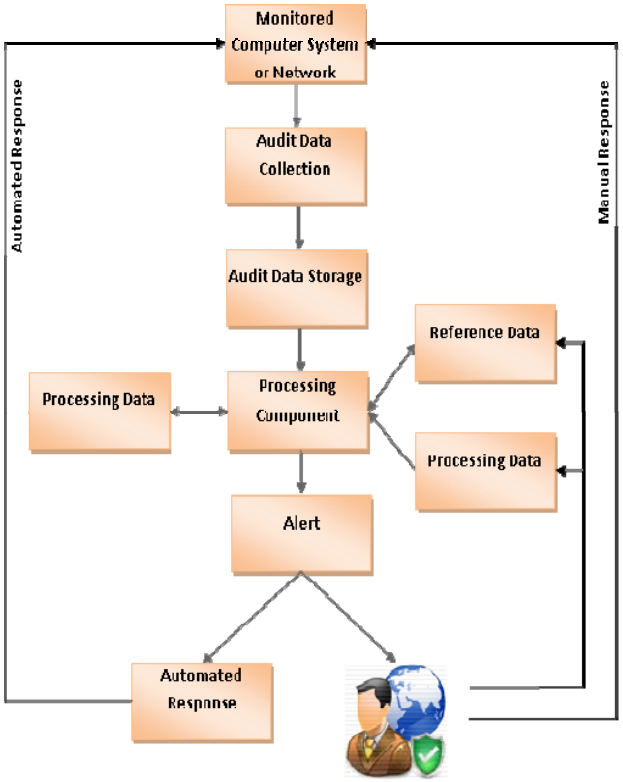

Fig. 1 Organization of a generalized data mining based IDS

- Audit data collection: IDS collect audit data and analyzed them by the data mining algorithms to detect suspicious activities or intrusions. The source of the data can be host/network activity logs, command-based logs, and application-based logs.
- Audit data storage: IDS store the audit data for future reference. The volume of audit data is extremely large. Currently adaptive intrusion detection aims to solve the problems of analyzing the huge volumes of audit data

and realizing performance optimization of detection rules.

- Processing component: The processing block is the heart of IDS. It is the data mining algorithms that apply for detecting suspicious activities. Algorithms for the analysis and detection of intrusions have been traditionally classified into two categories: misuse (or signature) detection, and anomaly detection.
- Reference data: The reference data stores information about known attacks or profiles of normal behaviors.
- Processing data: The processing element must frequently store intermediate results such as information about partially fulfilled intrusion signatures.
- Alert: It is the output of IDS that notifies the network security officer or automated intrusion prevention system (IPS).
- System security officer or intrusion prevention system (IPS) carries out the prescriptions controlled by the IDS.

III. Proposed Learning Algorithm

Given a database $D = \{t_1, t_2, ..., t_n\}$ where $t_i = \{t_{i1}, ..., t_{ih}\}$ and the database D contains the following attributes $\{A_1, A_2, ..., A_n\}$ and each attribute A_i contains the following attribute values $\{A_{i1}, A_{i2}, ..., A_{ih}\}$. The attribute values can be discrete or continuous. Also the database D contains a set of classes $C = \{C_1, C_2, ..., C_m\}$. Each example in the database D has a particular class C_j. The algorithm first clusters the database D into several clusters $\{D_1, D_2, ..., D_n\}$ depending on the similarity of examples in the database D. A similarity measure, $sim(t_i, t_l)$, defined between any two examples, t_1, t_2 in D, and an integer value k, the clustering is to define a mapping $f: D \rightarrow \{1,\ldots,K\}$ where each t_i is assigned to one cluster K_j. Suppose for two examples there is a match between two attribute values then the similarity becomes 0.5. If there is a match only in one attribute value, then similarity between the examples is taken as 0.25 and so on. Then the algorithm calculates the prior probabilities $P(C_j)$ and conditional probabilities $P(A_{ij}|C_j)$ for each cluster. The prior probability $P(C_j)$ for each class is estimated by counting how often each class occurs in the cluster. For each attribute A_i the number of occurrences of each attribute value A_{ij} can be counted to determine $P(A_i)$. Similarly, the conditional probability $P(A_{ij}|C_j)$ for each attribute values A_{ij} can be estimated by counting how often each attribute value occurs in the class in the cluster. For classifying a new example whose attribute values are known but class value is unknown, the algorithm checks in which cluster the new example belongs and then use that cluster's probability set to classify the new example. For classifying a new example, the prior probabilities and conditional probabilities are used to make the prediction. This is done by combining the effects of the different attribute values from that example. Suppose the example e_i has independent attribute values $\{A_{i1}, A_{i2}, ..., A_{ip}\}$, we know the $P(A_{ik} \mid C_j)$, for each class C_j and attribute A_{ik}. We then estimate $P(e_i \mid C_j)$ by

$$P(e_i \mid C_j) = P(C_j) \prod_{k=1 \rightarrow p} P(A_{ij} \mid C_j) \quad (1)$$

To classify the example, the probability that e_i is in a class is the product of the conditional probabilities for each attribute value with prior probability for that class. The posterior probability $P(C_j \mid e_i)$ is then found for each class and the example classifies with the highest posterior probability for that example. The main procedure of proposed algorithm is described as follows.

Algorithm

Input:

Database, D

Output:

Intrusion Detection Model

Learning Algorithm:

Step 1. **for each** example $t_i \in D$, check the similarity of examples: $sim(t_i, t_l)$;

Step 2. Put examples into cluster: $D_i \leftarrow t_i$;

Step 3. **for each** cluster D_i, calculate the prior probabilities: $P(C_j) = \dfrac{\sum t_{i \rightarrow C_j}}{\sum_{i=1}^{n} t_i}$;

Step 4. **for each** cluster D_i, calculate the conditional probabilities: $P(A_{ij}|C_j) = \dfrac{\sum_{i=1}^{n} A_{i \rightarrow C_j}}{\sum t_{i \rightarrow C_j}}$;

Step 5. **for each** cluster D_i, store the prior probabilities, $S_1 = P(C_j)$; and conditional probabilities, $S_2 = P(A_{ij}|C_j)$;

Step 6. For classifying new example, check in which cluster the example belongs and then use that cluster's probability set to classify the example.

IV. Experimental Analysis

A. Intrusion Detection Dataset

The KDD99 cup dataset was used in the 3rd International Knowledge Discovery and Data Mining Tools Competition for building a network intrusion detector, a predictive model capable of distinguishing between intrusions and normal connections [25]. In 1998, DARPA intrusion detection evaluation program, a simulated environment was set up to acquire raw TCP/IP dump data for a local-area network (LAN) by the MIT Lincoln Lab to compare the performance of various intrusion detection methods. It was operated like a real environment, but being blasted with multiple intrusion attacks and received much attention in the research community of adaptive intrusion detection. The KDD99 dataset contest uses a version of DARPA98 dataset. In KDD99 dataset, each example represents attribute values of a class in the network data flow, and each class is labeled either normal or attack. The classes in KDD99 dataset can be categorized into five main classes (one normal class and four main intrusion classes: probe, DOS, U2R, and R2L).

1) Normal connections are generated by simulated daily user behavior such as downloading files, visiting web pages.

2) Denial of Service (DoS) attack causes the computing power or memory of a victim machine too busy or too full to handle legitimate requests. DoS attacks are classified based on the services that an attacker renders unavailable to legitimate users like apache2, land, mail bomb, back, etc.

3) Remote to User (R2L) is an attack that a remote user gains access of a local user/account by sending packets to a machine over a network communication, which include send-mail, and Xlock.

4) User to Root (U2R) is an attack that an intruder begins with the access of a normal user account and then becomes a root-user by exploiting various vulnerabilities of the system. Most common exploits of U2R attacks are regular buffer-overflows, load-module, Fd-format, and Ffb-config.

5) Probing (Probe) is an attack that scans a network to gather information or find known vulnerabilities. An intruder with a map of machines and services that are available on a network can use the information to look for exploits.

In KDD99 dataset these four attack classes (DoS, U2R, R2L, and probe) are divided into 22 different attack classes that tabulated in Table I.

TABLE I. ATTACK CLASSES IN KDD99 DATASET

4 Main Attack Classes	22 Attack Classes
Denial of Service (DoS)	back, land, neptune, pod, smurt, teardrop
Remote to User (R2L)	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster
User to Root (U2R)	buffer_overflow, perl, loadmodule, rootkit
Probing	ipsweep, nmap, portsweep, satan

There are total 41 input attributes in KDD99 dataset for each network connection that have either discrete or continuous values and divided into three groups. The first group of attributes is the basic features of network connection, which include the duration, prototype, service, number of bytes from source IP addresses or from destination IP addresses, and some flags in TCP connections. The second group of attributes in KDD99 is composed of the content features of network connections and the third group is composed of the statistical features that are computed either by a time window or a window of certain kind of connections. Table II shows the number of examples of 10% training data and 10% testing data in KDD99 dataset. There are some new attack examples in testing data, which is no present in the training data.

TABLE II. NUMBER OF EXAMPLES IN TRAINING AND TESTING KDD99 DATA

Attack Types	Training Examples	Testing Examples
Normal	97277	60592
Denial of Service	391458	237594
Remote to User	1126	8606
User to Root	52	70
Probing	4107	4166
Total Examples	494020	311028

B. Experimental Analysis

In order to evaluate the performance of proposed learning algorithm, we performed 5-class classification using KDD99 network intrusion detection benchmark dataset. All experiments were performed using an Intel Core 2 Duo Processor 2.0 GHz processor (2 MB Cache, 800 MHz FSB) with 1 GB of RAM. The detection rates (DR) and false positives (FP) are used to estimate the performance of IDS, which are given as bellow:

$$\text{Detection Rate} = \frac{Total\_\det ected\_attacks}{Total\_attacks} * 100 \quad (2)$$

$$\text{False Positive} = \frac{Total\_misclassified\_process}{Total\_normal\_process} * 100 \quad (3)$$

The experimental results of proposed algorithm with naive Bayesian classifier (NB) are tabulated in Table III and Table IV.

TABLE III. RESULTS USING 41 ATTRIBUTES

Method	Normal	Probe	DoS	U2R	R2L
Proposed Algorithm (DR %)	99.66	99.24	99.62	99.19	99.08
Proposed Algorithm (FP %)	0.08	0.86	0.09	0.18	7.85
NB (DR %)	99.27	99.11	99.69	64.00	99.11
NB (FP %)	0.08	0.45	0.05	0.14	8.02

TABLE IV. RESULTS USING 19 ATTRIBUTES

Method	Normal	Probe	DoS	U2R	R2L
ISANBT (DR %)	99.77	99.61	99.56	99.23	99.15
ISANBT (FP %)	0.08	0.58	0.06	0.16	7.32
NB (DR %)	99.65	99.35	99.71	64.84	99.15
NB (FP %)	0.06	0.49	0.04	0.12	6.87

We also tested the performance of proposed algorithm using the reduced dataset of 12 and 17 attributes in KDD99, which increase the detection rate that are summarized in Table V.

TABLE V. EXPERIMENT ON REDUCED DATASET

Class Value	12 Attributes	17 Attributes
Normal	99.86	99.78
Probe	99.49	99.45
DoS	99.73	99.69
U2R	99,41	99.43
R2L	99.33	99.30

V. CONCLUSION

This paper introduced a learning algorithm for detecting network intrusions using naive Bayesian classifier with clustering. The proposed algorithm is suitable for analyzing large number of network logs or audit data. It improves the performance of detection rates for different types of intrusions. The main propose of this paper is to improve the performance of naïve Bayesian classifier for intrusion detection. We tested out proposed algorithm on KDD99 dataset that shows it maximized the balance detection rates for 4 attack classes in KDD99 dataset and minimized false positives at acceptable level. The future work focus on apply this algorithm in real time network and ensemble with other data mining algorithms for improving the detection rates in intrusion detection.

ACKNOWLEDGMENT

Support for this research received from University Lumière Lyon 2 – France, Department of Computer Science and

Engineering, Jahangirnagar University, Bangladesh, and Department of Computer Science and Engineering, United International University, Bangladesh.

REFERENCES

[1] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," Computer networks: The International Journal of Computer and Telecommunications Networking, 34, 2000, pp. 579-595.

[2] M. Stillerman, C. Marceau, and M. Stillman, "Intrusion detection on distributed systems," Communications of the ACM, 42(7), pp. 62-69.

[3] D. Barbara, J. Couto, S. Jajodia, L. Popyack, and N. Wu, "ADAM: Detecting intrusion by data mining," IEEE Workshop on Information Assurance and Security, West Point, New York, June 5-6, 2001.

[4] W. Lee, "A data mining and CIDF based approach for detecting novel and distributed intrusions," Recent Advances in Intrusion Detection, 3rd International Workshop, RAID 2000, Toulouse, France, October 2-4, 2000, Proc. Lecture Notes in Computer Science 1907 Springer, 2000, pp. 49-65.

[5] W. Lee, S. J. Stolfo, and K. W. Mok, "Adaptive Intrusion Detection: A Data Mining Approach," Artificial Intelligence Review, 14(6), December 2000, pp. 533-567.

[6] R. Wasniowski, "Multi-sensor agent-based intrusion detection system," In Proc. of the 2nd Annual Conference on Information Security, Kennesaw, Georgia, 2005, pp. 100-103.

[7] N.B. Amor, S. Benferhat, and Z. Elouedi, "Naïve Bayes vs. decision trees in intrusion detection systems," In Proc. of 2004 ACM Symposium on Applied Computing, 2004, pp. 420-424.

[8] YU Yan, and Huang Hao, "An ensemble approach to intrusion detection based on improved multi-objective genetic algorithm," Journal of Software, vol. 18, no. 6, June 2007, pp. 1369-1378.

[9] T. Shon, J. Seo, and J. Moon, "SVM approach with a genetic algorithm for network intrusion detection," In Proc. of 20th International Symposium on Computer and Information Sciences (ISCIS 2005), Berlin: Springer-Verlag, 2005, pp. 224-233.

[10] S. Mukkamala, G. Janoski, and A. H. Sung, "Intrusion detection using neural networks and support vector machines," In Proc. of the IEEE International Joint Conference on Neural Networks, 2002, pp.1702-1707.

[11] J. Luo, and S.M. Bridges, "Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection," International Journal of Intelligent Systems, John Wiley & Sons, vol. 15, no. 8, 2000, pp. 687-703.

[12] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," In Proc. of the SIAM Conference on Data Mining, 2003.

[13] James P. Anderson, "Computer security threat monitoring and surveillance," Technical Report 98-17, James P. Anderson Co., Fort Washington, Pennsylvania, USA, April 1980.

[14] Dorothy E. Denning, "An intrusion detection model," IEEE Transaction on Software Engineering, SE-13(2), 1987, pp. 222-232.

[15] Dorothy E. Denning, and P.G. Neumann "Requirement and model for IDES- A real-time intrusion detection system," Computer Science Laboratory, SRI International, Menlo Park, CA 94025-3493, Technical Report # 83F83-01-00, 1985.

[16] S.E. Smaha, and Haystack, "An intrusion detection system," in Proc. of the IEEE Fourth Aerospace Computer Security Applications Conference, Orlando, FL, 1988, pp. 37-44.

[17] S. Forrest, S.A. Hofmeyr, A. Somayaji, T.A. Longstaff, "A sense of self for Unix processes," in Proc. of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, USA, 1996, pp. 120-128.

[18] A. Valdes, K. Skinner, "Adaptive model-based monitoring for cyber attack detection," in Recent Advances in Intrusion Detection Toulouse, France, 2000, pp. 80-92.

[19] C. Kruegel, D. Mutz, W. Robertson, F. Valeur, "Bayesian event classification for intrusion detection," in Proc. of the 19th Annual Computer Security Applications Conference, Las Vegas, NV, 2003.

[20] M.L. Shyu, S.C. Chen, K. Sarinnapakorn, L. Chang, "A novel anomaly detection scheme based on principal component classifier," in Proc. of the IEEE Foundations and New Directions of Data Mining Workshop, Melbourne, FL, USA, 2003, pp. 172-179.

[21] D. Y. Yeung, and Y. X. Ding, "Host-based intrusion detection using dynamic and static behavioral models," *Pattern Recognition*, 36, 2003, pp. 229-243.

[22] W. Lee, S.J. Stolfo, "Data mining approaches for intrusion detection," In Proc. of the 7th USENIX Security Symposium (SECURITY-98), Berkeley, CA, USA, 1998, pp. 79-94.

[23] J.E. Dickerson, J.A. Dickerson, "Fuzzy network profiling for intrusion detection," In Proc. of the 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS), Atlanta, GA, 2000, pp. 301-306.

[24] M. Ramadas, S.O.B. Tjaden, "Detecting anomalous network traffic with self-organizing maps," In Proc. of the 6th International Symposium on Recent Advances in Intrusion Detection, Pittsburgh, PA, USA, 2003, pp. 36-54.

[25] The KDD Archive. KDD99 cup dataset, 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html