Fuzzy Vector Quantization for Network Intrusion Detection

Dat Tran[(1)], Wanli Ma[(1)], Dharmendra Sharma[(1)], and Thien Nguyen[(2)]
[(1)]*University of Canberra, Australia,* [(2)]*De Anza College, CA, USA*
dat.tran@canberra.edu.au

Abstract

This paper considers anomaly network traffic detection using different network feature subsets. Fuzzy c-means vector quantization is used to train network attack models and the minimum distortion rule is applied to detect network attacks. We also demonstrate the effectiveness and ineffectiveness in finding anomalies by looking at the network data alone. Experiments performed on the KDD CUP 1999 dataset show that time based traffic features in the last two second time window should be selected to obtain highest detection rates.

1. Introduction

Network intrusion detection systems can be classified into signature based intrusion detection and anomaly behavior detecting based intrusion detection. A signature based intrusion detection system constantly scans the network and try to match network traffic with some predefined patterns [1-3]. An anomaly behavior detecting based intrusion detection system builds normal traffic profile and uses this profile to detect abnormal traffic patterns and intrusion attempts. Extensive domain knowledge is required to provide signatures, yet the process to identify new signatures is time consuming and always legs behind the new attacks. On the other hand, an anomaly behavior detecting based intrusion detection system uses a statistical method in data mining to learn the patterns of network traffic. It promises reactive detections and also less or nil human intervening. The learning process could be unsupervised just from raw network data or supervised from labeled network data. Labeling network data is a time consuming process and requires domain knowledge and human involvement. However, it is seen that the system after labeling and training network attack models can operate itself and can detect not only labeled attacks but also new attacks. Different techniques have been proposed to train network attack models [4-7].

There are many available features describing network traffic. Basic features for a network connection include the duration of the current connection, the source IP address, the destination IP address, octets transferred (both inbound and outgoing), the protocol type, the service port, the connection flags etc. Some of these features are of symbolic values such as the protocol types (HTTP, FTP, TCP, and UDP) and connection flags (ACK and RST). Other features are digital values such as duration of the connection and the octets transferred. Note that the source IP address, the destination IP address, and the service port features are regarded as symbolic values although they appear in digital format, because the values are just served as identities. Compound features, such as the number of connections happened in a fixed time window and the number of service ports contacted in the fixed time windows, can be calculated from the basic features over the time. They are often used to construct traffic profile. The selection of the feature has direct impact on the results of anomaly detection.

Values of network traffic octets features range in several orders of magnitudes, from several bytes to 108 bytes. Network also has unique burst nature. The number of connections and the volumes of octets transferred may be boosted to extraordinary large numbers from time to time and cannot be predicted beforehand. The reasons which caused the burst are diverse, ranging from normal operation to being under attacks. To make these values comparable, normalization techniques are required.

After selecting network traffic features, we use fuzzy c-means vector quantization (FVQ) to train network attack models. A FVQ model is a set of cluster centers found using fuzzy c-means (FCM) clustering to cluster the training dataset. FCM clustering is the most widely used approach in both theory and practical applications of fuzzy clustering techniques to unsupervised classification. It is an

0-7695-3032-X/07 $25.00 © 2007 IEEE
DOI 10.1109/GrC.2007.124

IEEE computer society

extension of the hard c-means (k-means) technique [20-21].

Experiments were performed on the KDD CUP 1999 dataset. Different feature subsets were selected to evaluate the intrusion detection rate. Experimental results showed that time based traffic features in the last two second time window should be selected to obtain highest detection rates.

The rest of the paper is as follows. Section 2 briefly reviews current clustering-based methods. Section 3 describes the KDD CUP 1999 dataset and the attack types, respectively. Section 4 presents fuzzy vector quantization technique. Section 5 presents experimental results. Finally, we conclude the paper in Section 6.

2. Current Clustering-Based Intrusion Detection Methods

Most of the current clustering-based methods were evaluated using the KDD CUP 1999 dataset or the DARPA 1999 dataset. A simple variant of single-linkage clustering was applied in [12] to learn network traffic patterns on unlabelled noisy data. The KDD CUP 1999 dataset was used in this approach [13] but it was not clear that what features were selected. This approach achieved from 40% to 55% detection rate and from 1.3% to 2.3% false positive rate. NATE [14, 15] was proposed to select some of the traffic records to improve the detection performance. The selected features include the frequency of TCP flags, the average and total number of bytes transferred, the percentage of session control flags, and also network packet header information. The dataset was MIT Lincoln lab data [16]. CLAD (Clustering for Anomaly Detection) in [11] used *k*-NN algorithm and an unsupervised training process. CCAS [17] was proposed for supervised clustering and classification. They chose clustering method because it relies very little on the distribution models of data. Weka data mining tools [18] was used and selected features were time stamps, protocol, destination IP, Source IP, Service port, number of packets, duration, and the country of source IP address. However it is unclear that how symbolic values (protocol) were handled.

3. Network Data and Attacks

3.1. KDD CUP 1999 dataset

This dataset was based on MIT Lincoln Lab intrusion detection dataset, also known as DARPA dataset [16]. The data was produced for "The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining" [13]. The raw network traffic records have already been converted into vector format. Each vector has 40 features. The meanings of these features can be found in [13] and [19]. In this paper, we ignore features with symbolic values. Other features are classified into the following four categories:

- Category I: Features of a connection, including duration, octets transferred, and wrong fragmentation flags.
- Category II: Features that are actually not traffic features. They cannot be obtained by looking at traffic records alone.
- Category III: Features that are time based traffic features. They are statistics of traffic features in the previous 2-second window. The calculation is based on the source IP address.
- Category IV: Features that are the same as Category III, except that the calculation is destination IP address oriented

3.2. Network Attacks

The attacks listed in feature vectors of KDD CUP 1999 dataset come from MIT Lincoln intrusion detection dataset web site [16]. The labels are mostly the same except a few discrepancies. MIT Lincoln lab web site lists 2 types of buffer overflow attack: *eject* and *ffb*. The former explores the buffer overflow problem of *eject* program of Solaris, and the later explores the buffer overflow problem of *ffb* config program. Guessing user logon names and passwords through remote logon via telnet session is labeled as *guess_passwd* in KDD CUP 1999 dataset, but listed as dict on MIT Lincoln lab web site. Finally, we cannot find the counterparts of *syslog* and *warez* in KDD CUP 1999 dataset. In addition to the attack labels, KDD CUP 1999 dataset also has the label *normal*, which means that the traffic is normal and free from any attack. The labels used in KDD CUP 1999 dataset are as follows. The meanings of the labels are mainly from [16].

4. Fuzzy Vector Quantization

Let $X = \{x_1, x_2, ..., x_T\}$ be a set of T vectors, the structure of which is analyzed by means of a cluster analysis technique. Fuzzy clustering known as unsupervised learning in X is a fuzzy partitioning of X into c fuzzy subsets or c clusters, $1 < c < T$. The most important requirement is to find a suitable measure of clusters, referred to as a fuzzy clustering criterion. Objective function methods allow the most precise formulation of the fuzzy clustering criterion. The most well known objective function for fuzzy clustering in X is the least-squares functional, that is, the infinite family of fuzzy c-means (FCM) functions, generalized from the classical within-groups sum of squared error function

$$J_m(U, \lambda; X) = \sum_{i=1}^{c}\sum_{t=1}^{T} u_{it}^m d_{it}^2 \tag{1}$$

where $U = \{u_{it}\}$ is a fuzzy c-partition of X, each u_{it} represents the degree of vector x_t belonging to the ith cluster and is called the fuzzy membership function. For $1 \leq i \leq c$ and $1 \leq t \leq T$, we have

$$0 \leq u_{it} \leq 1, \quad \sum_{i=1}^{c} u_{it} = 1, \text{ and } 0 < \sum_{t=1}^{T} u_{it} < T \tag{2}$$

$m \geq 1$ is a weighting exponent on each fuzzy membership u_{it} and is called the degree of fuzziness; $\lambda = (\mu_1, ..., \mu_c)$ set of cluster centers and, d_{it} is the distance from vector x_t to center μ_i, known as a measure of dissimilarity

$$d_{it}^2 = \| x_t - \mu_i \|^2 \tag{3}$$

The basic idea in the FCM is to minimize J_m over the variables U and λ, on the assumption that matrices U that are part of optimal pairs for J_m identify good partitions of the data. Minimizing the fuzzy objective function J_m in (1) gives

$$u_{it} = \left[\sum_{k=1}^{c} (d_{it} / d_{kt})^{\frac{2}{m-1}} \right]^{-1} \tag{4}$$

$$\mu_i = \sum_{t=1}^{T} u_{it}^m x_t \Big/ \sum_{t=1}^{T} u_{it}^m \tag{5}$$

The training and classification procedures based on this FVQ technique can be summarized as follows

Training:

1. Choose any inner product norm metric for $\Re^d$, fix c and m, $2 < c < T$, $m > 1$. Generate matrix U at random satisfying (2)
2. For $i = 1, \ldots, c$, compute the c fuzzy mean vectors $\{\mu_i\}$ with (5) and the distances d_{it} with (3). If $d_{it} = 0$ for some t, set $u_{it} = 1$, $u_{is} = 0$, $\forall s \neq t$
3. Update matrix U using (4)
4. Stop if the decrease in the value of the fuzzy objective function J_m at the current iteration relative to the value of the J_m at the previous iteration is below a chosen threshold, otherwise go to step 2.

Detection:

1. Given x as an unknown network feature vector and $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_M\}$ as a set of M trained attack models
2. Calculate the minimum distance between x and λ_m, $m = 1, \ldots, M$

$$d(x, \lambda_m) = \min_i d^2(x, \mu_i^{(m)}) = \min_i \| x - \mu_i^{(m)} \|^2 \tag{6}$$

3. Assign x to the attack model λ_{m^*} that has the minimum distance:

$$m^* = \arg\min_m d(x, \lambda_m) \tag{7}$$

5. Experimental Results

The proposed method for the network intrusion detection was evaluated using the KDD CUP 1999 data set for training and the *Corrected* data set for testing. Training sets for the 23 attacks mentioned above were extracted from the KDD CUP 1999 dataset and the maximum number of feature vectors for each of the training sets was set to 2000. All 311029 feature vectors in the testing set were used. There were no feature vectors for the *spy* and *warezclient* attacks and only 2 feature vectors for the *loadmodule* attack found in the testing set. However, there were 58001, 60593 and 164091 found for the *neptune*, *normal* and *smurf* attacks, respectively.

Because the feature values have different ranges, the following normalization of features was used

$$x'_{tj} = \frac{x_{tj} - \mu_j}{s_j} \tag{8}$$

where x_{tj} is the j-th feature of the t-th vector, μ_j the mean value of all T vectors for feature j, and s_j the absolute standard deviation, that is

$$s_j = \frac{1}{T}\sum_{t=1}^{T} | x_{tj} - \mu_j | \tag{9}$$

We trained 23 models for the 23 attacks using the training sets extracted from the KDD CUP 1999 data set. We used all feature vectors in the testing set to test the models. We run experiments with 15 different combinations of the feature categories listed in Section III. Each experiment is conducted with the raw data and normalized data.

TABLE I
RECOGNITION RATES FOR DATA WITH THE "NORMAL" LABEL

Category	With normalization	Without normalization
I	38.4%	10.1%
II, III, IV	51.3%	52.7%
II, III	52.2%	20.5%
I, II, III	53.3%	10.2%
I, II, III, IV	59.6%	12.4%
II, IV	61.7%	34.7%
II, III, IV	62.8%	50.2%
I, II	70.2%	12.4%
II	70%	70%
III, IV	79.5	52.9%
III	80.9%	17.6%
I, III, IV	83.6%	15.4%
I, III	86%	13.4%
I, IV	88.7%	15.1%
IV	91.1%	37.5%

From Table 1, we can see that Category IV features contribute most to the recognition rate, and the other features actually more or less contribute negatively. Category I features are least important. When combining with other features, they drag down the recognition rate. Category II features cannot be obtained from network traffic data only. These features in most vectors, about 70%, just repeat the exactly same values. This shows why Category II has the same results for with or without data normalization. Experimental results on Category III yields almost the same as that on Category IV. However, combining Category III and Category IV features does not increase the recognition rate.

6. Conclusion

We have studied the impact of feature selection and data normalization on detecting anomaly network traffic. We have used the KDD CUP 1999 dataset as the sample data for the study, and the detection algorithm used fuzzy c-means vector quantization. We trained 23 network attack models for the 23 labels of the dataset. We used all feature vectors in the testing set to test the models. We run the experiments with 15 different feature sets. Each experiment was conducted with the raw data and normalized data. We have found that the time based traffic features in the last two second time window should be selected to obtain highest detection rates

7. References

[1] Snort. Snort web site, http://www.snort.org.

[2] Cisco. Cisco IOS Firewall Intrusion Detection System, http://www.cisco.com/en/US/products/sw/secursw/ps2113/products_white_paper09186a008010e5c8.shtml.

[3] Paxson, V. Bro: a system for detecting network intruders in real-time. in Proceedings of the 7th USENIX Security Symposium. 1998. San Antonio, Texas, USA.

[4] Eskin, E. Anomaly Detection over Noisy Data Using Learned Probability Distributions. in The 17th International Conference on Machine Learning. 2000. Morgan Kaufmann, San Francisco, CA, USA.

[5] Balasubramaniyan, J.S., J.O. Garcia-Fernandez, et al. An Architecture for Intrusion Detection using Autonomous Agents. in 14th IEEE Computer Security Applications Conference (ACSAC '98). 1998. Scottsdale, AZ, USA: IEEE Computer Society.

[6] Lee, W. and D. Xiang. Information theoretic measures for anomaly detection. in 2001 IEEE Synposium on Security and Privacy. 2001.

[7] Ourston, D., S. Matzner, et al., Coordinated Internet attacks: responding to attack complexity. Journal of Computer Security, 2004. 12: p. 165-190.

[8] Anderson, R. and A. Khattak. The use of Information Retrieval Techniques for Intrusion Detection. in First International Workshop on Recent Advances in Intrusion Detection (RAID'98). 1998. Louvain-la-Neuve, Belgium.

[9] Sherif, J.S., R. Ayers, and T.G. Dearmond, Intrusion Detedction: the art and the practice, Part 1. Information Management & Computer Security, 2003. 11(4): p. 175-186.

[10] Sherif, J.S. and R. Ayers, Intrusion detection: methods and systems, Part II. Information Management & Computer Security, 2003. 11(5): p. 222-229.

[11] Chan, P.K., M.V. Mahoney, and M.H. Arshad, A Machine Learning Approach to Anomaly Detection.

2003, Florida Institute of Technology: Melbourne, FL, USA.

[12] Portnoy, L., E. Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. in Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001). 2001. Philadelphia, PA, USA: ACM Press.

[13] ACM. KDD CUP 1999 data. [cited 12 January 2007]; Available from: http://kdd.ics.uci.edu//databases/kddcup99/kddcup99.html.

[14] Taylor, C. and J. Alves-Foss. An Empirical Analysis of NATE: Network Analysis of Anomalous Traffic Events. in 10th New Security Paradigms Workshop. 2002. Virginia Beach, Virginia, USA: ACM Press.

[15] Taylor, C. and J. Alves-Foss. NATE: Network Analysis of Anomalous Traffic Events, a low-cost approach. in Proceedings of New Security Paradigms Workshop. 2001. Cloudcroft, New Mexico, USA.

[16] DARPA. DARPA Intrusion Detection Evaluation Data Sets. 1999 [cited 2006 15 October 2006]; Available from: http://www.ll.mit.edu/IST/ideval/data/data_index.html.

[17] Li, X. and N. Ye. Mining Normal and Intrusive Activity Patterns for Computer Intrusion Detection. in Intelligence and Security Informatics: Second Symposium on Intelligence and Security Informatics. 2004. Tucson, AZ, USA: Springer-Verlag.

[18] Caruso, C. and D. Malerba. Clustering as an add-on for firewalls. in Fifth International Conference on Data Mining, Text Mining and Their Business Applications (DATA MINING 2004). 2004. Malaga, Spain: WIT Press, Southampton, UK.

[19] Stolfo, S.J., W. Fan, et al. Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection: Results from the JAM Project. in Proceedings of 2000 DARPA Information Survivability Conference and Exposition. 2000.

[20] Duda, R.O. and P.E. Hart, Pattern classification and scene analysis. 1973, New York, USA: John Wiley & Sons.

[21] Bezdek, J.C., Pattern Recognition with Fuzzy Objective Function Algorithms. 1981, New York and London: Plenum Press