Automated Network Feature Weighting-Based Intrusion Detection Systems

Dat Tran, *Senior Member, IEEE*, Wanli Ma, Dharmendra Sharma
Faculty of Information Sciences and Engineering
University of Canberra, ACT 2601, Australia
Email: dat.tran@canberra.edu.au

***Abstract*—A common problem for network intrusion detection systems is that there are many available features describing network traffic and feature values are highly irregular with burst nature. Some values such as octets transferred range several orders of magnitudes, from several bytes to million bytes. The role of network features depends on which pattern to be detected: normal or intrusive one. Intrusion detection rates would be better if we know which network features are more important for a particular pattern. We therefore propose an automated feature weighting method for network intrusion detection based on a fuzzy subspace approach. Experimental results show that the proposed weighting method can improve the detection rates.**

Keywords: **Network intrusion detection, automated feature weighting, subspace vector quantization, fuzzy c-means, fuzzy entropy.**

I. Introduction

Network intrusion detection systems are automated systems that detect intrusions in computer network systems. Data mining-based 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]. The main advantage of this system is that it can accurately detect known attacks, while its drawback is that it cannot detect novel, previously unseen attacks. An anomaly behavior detecting-based intrusion detection system builds normal traffic profile and uses this profile to detect abnormal traffic patterns and intrusion attempts. The goal of this intrusion detection system is to determine whether an unknown network data item belongs to "normal" or to an intrusive pattern. 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. Different techniques have been proposed to train network attack models [4]-[7].

A common problem for all network intrusion detection systems is that 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 features 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 10^8 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.

Current network intrusion detection methods provide low detection rates because of the multi-dimensional data problem. For example, a simple variant of single-linkage clustering was applied in [8] to learn network traffic patterns on unlabelled noisy data. The KDD CUP 1999 dataset [9] was used and this approach achieved from 40% to 55% detection rate and from 1.3% to 2.3% false positive rate.

We presented fuzzy c-means vector quantization (VQ) modeling for network intrusion detection in our previous work [10]. A typical network intrusion detection system using fuzzy VQ is presented in Figure 1. This method achieved higher detection rates than the traditional k-means VQ modeling method. For further investigation, we carefully considered network data to improve the network intrusion detection. We found that network data values are highly irregular with burst nature. Some values such as octets transferred range several orders of magnitudes, from several bytes to million bytes. The detection system would be better if we know which network features are important. So we considered network data as a set X of feature vectors of M dimensions, i.e. M features. Each feature vector is considered as a point in an M-dimensional space. For example, $M = 41$ in the KDD CUP 1999 dataset used in

our experiments. We extracted subsets of feature vectors of M' dimensions where $M' < M$ from the set X. Feature vectors in these subsets were considered as points in subspaces of the M-dimensional space. The choice of M' features was based on the meaning of features and our experience in computer network. We then used the same modeling method in [10] to model the network data subsets and measure the network intrusion detection rates for the entire set and all subsets. Experimental results showed that the choice of network features was dependent on the network attack type to be detected. Some features were good for detecting normal traffic pattern and other features were good for detecting abnormal traffic patterns.

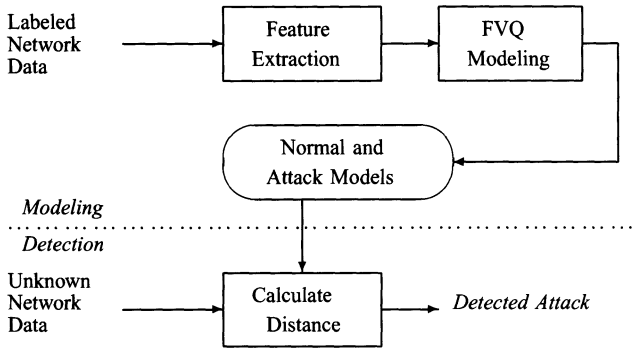

Fig. 1. Block diagram of a typical network intrusion detection system using fuzzy VQ modeling

Therefore we propose an automated feature weighting method to find out feature subspaces automatically in the entire feature space and then assign fuzzy weight values to network features depending on which subspace they belong to. The weighting algorithm based on fuzzy c-means estimation is proposed. A set of M weights for the feature vector set of M features will be calculated when we train network models. In the detection stage, these weight values are used to calculate similarity scores between unknown network data and network models. The modeling method is the same fuzzy c-means VQ. Experimental results show that the proposed weighting method can improve the detection rates.

The rest of the paper is as follows. Section 2 briefly reviews current detection methods. Section 3 presents the fuzzy VQ modeling method. Section 4 presents the fuzzy c-means-based subspace method. Section 5 describes network data and attack types. Section 6 presents experimental results. Finally, we conclude the paper in Section 7.

II. Current Network Intrusion Detection Methods

Different techniques have been proposed as detection engines, for example using probability distribution to detect intrusions [4], autonomous agents and distributed intrusion detection [5], data mining model [6] and hidden Markov model [7]. A good survey can be found in [11] and [12]. Due to the highly irregular distribution of the network data, which has "power-law distribution" and is "one-sided and heavy tailed" [13], using clustering method is strongly advocated by a number of research groups.

Most of the current clustering-based methods were evaluated using the KDD CUP 1999 dataset [9] or the DARPA 1999 dataset [14]. A simple variant of single-linkage clustering was applied in [13] to learn network traffic patterns on unlabelled noisy data. The KDD CUP 1999 dataset was used 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 (Network Analysis of Anomalous Traffic Events) in [15], [16] 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 [14]. CLAD (Clustering for Anomaly Detection) in [13] 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.

III. Vector Quantization

Vector quantization (VQ) modeling is an efficient data reduction method, which is used to convert a feature vector set into a small set of distinct vectors using a clustering technique. Advantages of this reduction are reduced storage and computation. The distinct vectors are called codevectors and the set of codevectors that best represents the training set is called the codebook. Since there is only a finite number of code vectors, the process of choosing the best representation of a given feature vector is equivalent to quantizing the vector and leads to a certain level of quantization error. This error decreases as the size of the codebook increases, however the storage required for a large codebook is non-trivial. The VQ codebook can be used as a model in pattern recognition. The key point of VQ modelling is to derive an optimal codebook which is commonly achieved by using a clustering technique.

A. Vector Quantization Modeling

VQ modeling can be summarized as follows. Given a training set of T feature vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, where each source vector $\mathbf{x}_t = (x_{t1}, x_{t2}, \ldots, x_{tM})$ is of M dimensions. Let $\lambda = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$ represent the codebook of size K, where $\mathbf{c}_k = (c_{k1}, c_{k2}, \ldots, c_{kM})$, $k = 1, 2, \ldots, K$ are code vectors. Each code vector $\mathbf{c}_k$ is assigned to an encoding region R_k in the partition $\Omega = \{R_1, R_2, \ldots, R_K\}$. Then the source vector $\mathbf{x}_t$ can be represented by the encoding region R_k and expressed by

$$V(\mathbf{x}_t) = \mathbf{c}_k, \text{ if } \mathbf{x}_t \in R_k \quad (1)$$

B. K-Means Partition

Let $U = [u_{kt}]$ be a matrix whose elements are memberships of $\mathbf{x}_t$ in the nth cluster, $k = 1, \ldots, K$, $t = 1, \ldots, T$. A K-partition space for X is the set of matrices U such that [21]

$$u_{kt} \in \{0,1\}\ \forall k,t,\ \sum_{k=1}^{K} u_{kt} = 1\ \forall t,\quad 0 < \sum_{t=1}^{T} u_{kt} < T\ \forall k \tag{2}$$

where $u_{kt} = u_k(\mathbf{x}_t)$ is 1 or 0, according to whether $\mathbf{x}_t$ is or is not in the kth cluster, $\sum_{k=1}^{K} u_{kt} = 1\ \forall t$ means each $\mathbf{x}_t$ is in exactly one of the K clusters, and $0 < \sum_{t=1}^{T} u_{kt} < T$ $\forall k$ means that no cluster is empty and no cluster is all of X because of $1 < K < T$.

The VQ method is based on minimization of the sum-of-squared-errors function as follows

$$J(U,\lambda;X) = \sum_{k=1}^{K}\sum_{t=1}^{T} u_{kt} d_{kt}^2 \tag{3}$$

where λ is a set of prototypes, in the simplest case, it is the set of cluster centers $\lambda = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$, and d_{kt} is the Euclidean norm of $(\mathbf{x}_t - \mathbf{c}_k)$. Minizing $J(U,\lambda;X)$ over the variables U and λ yields the following equations

$$\mathbf{c}_k = \sum_{t=1}^{T} u_{kt}\mathbf{x}_t \Big/ \sum_{t=1}^{T} u_{kt} \qquad 1 \le k \le K \tag{4}$$

$$u_{kt} = \begin{cases} 1 : d_{kt} < d_{jt} & j = 1, \ldots, K,\ j \ne k \\ 0 : \text{otherwise} \end{cases} \tag{5}$$

C. Fuzzy C-Means Partition

Let $U = [u_{kt}]$ be a matrix whose elements are fuzzy memberships of $\mathbf{x}_t$ in the nth cluster, $k = 1, \ldots, K$, $t = 1, \ldots, T$. A K-partition space for X is the set of matrices U such that [21]

$$u_{kt} \in [0,1]\ \forall k,t,\ \sum_{k=1}^{K} u_{kt} = 1\ \forall t,\quad 0 < \sum_{t=1}^{T} u_{kt} < T\ \forall k \tag{6}$$

where $u_{kt} \in [0,1]\ \forall k,t$ and $\sum_{k=1}^{K} u_{kt} = 1\ \forall t$ mean it is possible for each $\mathbf{x}_t$ to have an arbitrary distribution of fuzzy membership among the K fuzzy clusters, and $0 < \sum_{t=1}^{T} u_{kt} < T\ \forall k$ means that no cluster is empty and no cluster is all of X because of $1 < K < T$.

The fuzzy c-means vector quantization (FCMVQ) method is based on minimization of the sum-of-squared-errors function as follows [21], [22]

$$J(U,\lambda;X) = \sum_{k=1}^{K}\sum_{t=1}^{T} u_{kt}^{\gamma} d_{kt}^2 \tag{7}$$

where λ is a set of prototypes, $\gamma > 1$ is a weighting exponent on each fuzzy membership u_{it} and controls the degree of fuzziness, in the simplest case, it is the set of cluster centers $\lambda = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$, and d_{kt} is the Euclidean norm of $(\mathbf{x}_t - \mathbf{c}_k)$. The basic idea of the FCM method is to minimize $J(U,\lambda;X)$ over the variables U and λ on the assumption that matrix U, which is part of the optimal pairs for $J(U,\lambda;X)$, identifies the good partition of the data. The FCMVQ algorithm is summarized as follows.

1) Given a training data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, where $\mathbf{x}_t = (x_{t1}, x_{t2}, \ldots, x_{tK})$, $t = 1, 2, \ldots, T$.
2) Initialize the membership values $u_{kt}, 1 \le k \le K, 1 \le t \le T$, at random
3) Given $\epsilon > 0$ (small real number).
4) Set $i = 0$ and $J^{(i)}(U,\lambda;X) = 0$. Iteration:
   a) Compute cluster centers
   $$\mathbf{c}_k = \sum_{t=1}^{T} u_{kt}^{\gamma}\mathbf{x}_t \Big/ \sum_{t=1}^{T} u_{kt}^{\gamma} \quad ,1 \le k \le N \tag{8}$$
   b) Compute d_{kt} and $J^{(i+1)}(U,\lambda;X)$
   $$d_{kt} = \|\mathbf{c}_k - \mathbf{x}_t\|_2 \tag{9}$$
   c) Update membership values
   $$u_{kt} = \frac{1}{\sum_{n=1}^{K} \left(d_{kt}^2/d_{nt}^2\right)^{1/(\gamma-1)}} \tag{10}$$
5) If
   $$\frac{|J^{(i+1)}(U,\lambda;X) - J^{(i)}(U,\lambda;X)|}{J^{(i+1)}(U,\lambda;X)} > \epsilon \tag{11}$$
   then set $J^{(i)}(U,\lambda;X) = J^{(i+1)}(U,\lambda;X)$, $i = i+1$ and go to step (a).

IV. Fuzzy Subspace Method for FCMVQ

We present the fuzzy subspace method based on fuzzy c-means estimation for FCM VQ. A similar subspace method for VQ was proposed in [20].

Let $W = [w_1, w_2, \ldots, w_M]$ be the weight vector for M dimensions and α be a parameter weight for w_m. The equation (7) is modified as follows

$$J_\alpha(U,W,\lambda;X) = \sum_{k=1}^{K}\sum_{t=1}^{T} u_{kt}^{\gamma} \sum_{m=1}^{M} w_m^{\alpha} d_{ktm}^2 \tag{12}$$

where $\alpha > 1$, d_{ktm} is the m-th component distance of the distance d_{kt} between $\mathbf{c}_k$ and $\mathbf{x}_t$

$$d_{ktm}^2 = (c_{km} - x_{tm})^2,\quad d_{kt}^2 = \sum_{m=1}^{M} w_m^{\alpha} d_{ktm}^2 \tag{13}$$

and weight values satisfy the following conditions:

$$0 \le w_m \le 1\quad \forall m,\qquad \sum_{m=1}^{M} w_m = 1 \tag{14}$$

The basic idea of the fuzzy c-means subspace-based FCMVQ (FCMS-FCMVQ) method is to minimize $J_\alpha(U,W,\lambda;X)$ over the variables U, W, and λ on the assumption that matrix U identifies the good partition of the data, and that matrix W identifies the good dimension of the data.

A. Network Intrusion Modeling

The FCMS-FCMVQ modeling algorithm is summarized as follows:

1) Given a training data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, where $\mathbf{x}_t = (x_{t1}, x_{t2}, \ldots, x_{tM})$, $t = 1, 2, \ldots, T$.
2) Initialize memberships $u_{kt}, 1 \leq t \leq T, 1 \leq k \leq K$, at random satisfying (6)
3) Initialize weight values $w_m, 1 \leq m \leq M$ at random satisfying (14)
4) Given $\alpha > 1$, $\gamma > 1$ and $\epsilon > 0$ (small real number)
5) Set $i = 0$ and $J_\alpha^{(i)}(U, W, \lambda; X) = 0$. Iteration:
   a) Compute cluster centers using (8)
   b) Compute distance components d_{ktm} and distances d_{kt} using (13)
   c) Update weight values

$$w_m = \frac{1}{\sum_{n=1}^{M} \left(D_m^2 / D_n^2\right)^{1/(\alpha-1)}}$$
$$D_m^2 = \sum_{k=1}^{K} \sum_{t=1}^{T} u_{kt} d_{ktm}^2 \tag{15}$$

   d) Update membership values using (10)
   e) Compute $J_\alpha^{(i+1)}(U, W, \lambda; X)$ using (12)
   f) If

$$\frac{|J_\alpha^{(i+1)}(U, W, \lambda; X) - J_\alpha^{(i)}(U, W, \lambda; X)|}{J_\alpha^{(i+1)}(U, W, \lambda; X)} > \epsilon \tag{16}$$

   then set $J_\alpha^{(i)}(U, W, \lambda; X) = J_\alpha^{(i+1)}(U, W, \lambda; X)$, $i = i + 1$ and go to step (a).

B. Network Intrusion Detection

Assuming there are N network attack models. Given an unknown network feature vector $\mathbf{x}$, the task is to find which attack $\mathbf{x}$ belongs to. The following algorithm is proposed

1) Given an unknown network feature vector $\mathbf{x}$ and N network attack models.
2) Calculate the minimum distance between $\mathbf{x}$ and $\lambda_n, n = 1, \ldots, N$

$$d_n = \min_k d(\mathbf{x}, \mathbf{c}_{kn}) \tag{17}$$

   where $d(\cdot)$ is defined in (13) and $\mathbf{c}_{kn}$ is the kth code vector in λ_n.
3) Assign $\mathbf{x}$ to the n^*-th network attack that has the minimum distance:

$$n^* = \arg\min_n (d_n) \tag{18}$$

V. Network Data and Attack Types

A. KDD CUP 1999 dataset

We consider a sample dataset which is the KDD CUP 1999 dataset. This dataset was based on MIT Lincoln Lab intrusion detection dataset, also known as DARPA dataset [14]. The data was produced for "The Third International Knowledge Discovery and Data Mining Tools Competition", which was held in conjunction with the Fifth International Conference on Knowledge Discovery and Data Mining. The raw network traffic records have already been converted into vector format. Each feature vector consists of 41 featues. The meanings of these features can be found in [9]. 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.

B. Network Attacks

The attacks listed in feature vectors of KDD CUP 1999 dataset come from MIT Lincoln intrusion detection dataset web site [14]. The labels are mostly the same except a few discrepancies. The 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 the KDD CUP 1999 dataset, but listed as dict on the MIT Lincoln lab web site. Finally, we cannot find the counterparts of syslog and warez in the KDD CUP 1999 dataset. In addition to the attack labels, the KDD CUP 1999 dataset has also the label *normal*, which means that the traffic is normal and free from any attack.

VI. Experimental Results

The proposed method for network intrusion detection was evaluated using the KDD CUP 1999 data set for training and the *Corrected* data set for testing. The number of feature vectors for training each attack was set to 5000. There were not sufficient data for all attack types, so we selected the *normal* network pattern and the 5 attacks which were *ipsweep*, *neptune*, *portsweep*, *satan*, and *smurf*. The testing data set contains 60593 feature vectors for the *normal* network pattern, and 306, 58001, 354, 1633 and 164091 feature vectors for the five attacks, respectively. Three symbolic features were not considered in our experiments.

From Table I for the *normal* only, we can see that the FCM VQ modeling technique achieved higher detection rates than the standard K-means VQ method and the FCM subspace-based FCM VQ achieved the highest rates. However, with the overall detection rates for the *normal* model and 5 attack models *ipsweep*, *neptune*, *portsweep*, *satan*, and *smurf* shown in Table II, we can see that all the systems achieved high

TABLE I
DETECTION RESULTS (IN %) FOR THE *normal* MODEL USING DIFFERENT MODELING METHODS, WHERE $\alpha = 4.0$ AND $\gamma = 1.1$

NORMAL	Model Size				
Modeling	4	8	12	16	20
K-MeansVQ	33.6	31.7	31.2	32.0	32.4
FCMVQ	41.7	38.3	59.6	61.5	61.8
FCMS-FCMVQ	42.2	44.4	64.0	62.5	64.7

TABLE II
DETECTION RESULTS (IN %) FOR THE *normal* MODEL AND 5 ATTACK MODELS *ipsweep*, *neptune*, *portsweep*, *satan*, AND *smurf* USING DIFFERENT MODELING METHODS, WHERE $\alpha = 4.0$ AND $\gamma = 1.1$

ALL	Model Size				
Modeling	4	8	12	16	20
K-MeansVQ	68.0	75.1	75.5	75.5	75.9
FCMVQ	77.5	76.3	81.0	81.1	81.1
FCMS-FCMVQ	77.6	77.6	81.9	82.0	82.4

detection rates comparing with the results for the *normal* model only, where the fuzzy modeling methods performed better than the standard K-means VQ one. The FCM subspace-based FCM VQ method again provided higher detection rates than the FCM VQ one. In all the cases, the FCM subspace-based FCM VQ method achieved the highest detection rates.

We also conducted a set of experiments for the network data using the normalization technique as follows

$$x'_{tm} = \frac{x_{tm} - \mu_m}{s_m}, \qquad s_m = \frac{1}{T}\sum_{t=1}^{T} |x_{tm} - \mu_m| \qquad (19)$$

where x_{tm} is the m-th feature of the t-th feature vector, μ_m the mean value of all T feature vectors for feature m, and s_m the mean absolute deviation.

TABLE III
DETECTION RESULTS (IN %) FOR THE *normal* MODEL USING DIFFERENT MODELING METHODS, WHERE $\alpha = 4.0$ AND $\gamma = 1.1$. ALL NETWORK DATA WERE NORMALISED.

NORMAL	Model Size				
Modeling	4	8	12	16	20
K-MeansVQ	83.4	89.5	94.2	95.0	95.4
FCMVQ	85.0	89.6	94.3	95.2	95.8
FCMS-FCMVQ	82.1	93.7	95.3	96.0	96.4

TABLE IV
DETECTION RESULTS (IN %) FOR THE *normal* MODEL AND 5 ATTACK MODELS *ipsweep*, *neptune*, *portsweep*, *satan*, AND *smurf* USING DIFFERENT MODELING METHODS, WHERE $\alpha = 4.0$ AND $\gamma = 1.1$. ALL NETWORK DATA WERE NORMALISED.

ALL	Model Size				
Modeling	4	8	12	16	20
K-MeansVQ	29.9	46.4	52.9	53.1	53.3
FCMVQ	30.2	62.3	62.4	64.1	64.2
FCMS-FCMVQ	62.1	64.5	65.0	65.2	65.3

With the normalised network data, the detection rates for the *normal* model in Table III are very high comparing with those in Table I. However the overall detection rates in Table IV are lower than those in Table II. Although the proposed fuzzy subspace method achieved the highest results in all of the experiments, the lower detection rates in Table IV are a challenge.

VII. CONCLUSION

We have proposed automated feature weighting method by considering feature subspaces in a multidimensional feature space for network data. The proposed method is based on fuzzy c-means estimation to assign fuzzy weight values to network features depending on which subspace they belong to. We have used the KDD CUP 1999 dataset as the sample data to evaluate the proposed method. For both unnormalised and normalised network data, the proposed method provided better recognition results. The proposed subspace methods have just considered the difference between dimensions and have not considered clusters in each dimension. For further investigation, we are considering other automated weighting subspace methods that can assign different weights to clusters even in the same dimension. This may help find a better solution to improve the overall detection rates shown in Table IV.

REFERENCES

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

[2] Cisco. http://www.cisco.com/en/US/products/sw/secursw/ps2113/products_white_paper09186a008010e5c8.shtml.

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

[4] E. Eskin, "Anomaly Detection over Noisy Data Using Learned Probability Distributions", in the 17th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, USA, 2000, pp. 255-262.

[5] J.S. Balasubramaniyan, J.O. Garcia-Fernandez, et al., "An Architecture for Intrusion Detection using Autonomous Agents", in Proceedings of the 14th IEEE ACSAC 1998, Scottsdale, AZ, USA, pp. 13-24.

[6] W. Lee and D. Xiang, "Information theoretic measures for anomaly detection", in 2001 IEEE Synposium on Security and Privacy, pp. 130-143.

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

[8] L. Portnoy, 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, USA, pp. 333-342.

[9] ACM KDD CUP 1999 Data Set, available at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[10] D. Tran, W. Ma, D. Sharma and T. Nguyen, "Fuzzy Vector Quantization for Network Intrusion Detection", IEEE International Conference on Granular Computing, pp. 566-570, Silicon Valley, 2-4 November 2007, USA

[11] J.S. Sherif, R. Ayers, and T. G. Dearmond, "Intrusion Detedction: the art and the practice", Part 1. Information Management and Computer Security, 2003, vol. 11, no. 4, pp. 175-186.

[12] J.S. Sherif and R. Ayers, "Intrusion detection: methods and systems", Part II. Information Management and Computer Security, 2003, vol. 11, no. 5, pp. 222-229.

[13] P.K. Chan, M.V. Mahoney, and M.H. Arshad, "A Machine Learning Approach to Anomaly Detection", Technical Report CS-2003-06, 2003.

[14] DARPA Intrusion Detection Evaluation Data Sets 1999, available at http://www.ll.mit.edu/IST/ideval/data/data_ index.html

[15] C. Taylor 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, pp. 18-26.

[16] C. Taylor 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, pp. 89-96.
[17] X. Li 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, USA, Springer-Verlag, vol. 3073, pp. 1611-3349.
[18] C. Caruso and D. Malerba, "Clustering as an add-on for firewalls", Data Mining, WIT Press, 2004.
[19] H. Yang, F. Xie, and Y. Lu, "Clustering and Classification Based Anomaly Detection", Lecture Notes in Computer Science, 2006, vol. 4223, pp. 1611-3349.
[20] J.Z. Huang, M. K. Ng, H. Rong, and Z. Li, "Automated Variable Weighting in *k*-means Type Clustering", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2005, vol. 27, no. 5, pp. 657-668.
[21] D. Tran and T. Pham, "Modeling Methods for Cell Phase Classification", Book chapter in the book Advanced Computational Methods for Biocomputing and Bioimaging, Editors: T.D. Pham, H. Yan, D. I. Crane, Nova Science Publishers, New York, USA, ISBN: 1-60021-278-6, 2007, chapter 7, pp. 143-166.
[22] D. Tran, W. Ma, D. Sharma and T. Nguyen, "Fuzzy Vector Quantization for Network Intrusion Detection", IEEE International Conference on Granular Computing, Silicon Valley, 2-4 November 2007, USA.
[23] D. Tran and W. Wagner, "Fuzzy entropy clustering", in Proceedings of FUZZ-IEEE Conference, 2000, vol. 1, pp. 152-157.
[24] S.J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P.K. Chan, "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, pp. 1130-1144.
[25] R. Anderson 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.