Automated Feature Weighting for Network Anomaly Detection

Dat Tran, Wanli Ma, and Dharmendra Sharma

University of Canberra, Faculty of Information Sciences and Engineering, Australia

Summary
A number of network features is used to describe normal and intrusive traffic patterns. However the choice of features is dependent on which pattern to be detected. In order to identify which network features are more important for a particular network pattern, we propose an automated feature weighting method based on a fuzzy subspace approach to vector quantization modeling that can assign a weight to each feature when network models are trained. The proposed method not only increases the detection rate but also reduces false alarm rate as presented in our experiments.

Key words:
Network anomaly detection, automated feature weighting, subspace vector quantization, fuzzy c-means, fuzzy entropy.

1. Introduction

An anomaly behavior detecting-based intrusion detection system builds normal traffic model and uses this model to detect abnormal traffic patterns and intrusion attempts. The goal of this anomaly detection system is to determine whether an unknown network data item belongs to normal or to an intrusive pattern [1]-[4]. This is different from signature-based intrusion detection. A signature-based intrusion detection system constantly scans the network and tries to match network traffic with some predefined patterns [5]-[7]. 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.

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. 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 this 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.

In order to identify which network features are more important for a particular network pattern, we considered network data as a set X of feature vectors of M dimensions, i.e. M features. Each feature vector was 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 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.

Therefore we propose an automated feature weighting method to find out feature subspaces automatically in the entire feature space and then assign weight values to network features depending on which subspace they belong to. The weighting algorithms based on fuzzy *c*-means and fuzzy entropy techniques are proposed. A set of M weights for the feature vector set of M features will be calculated when we train the normal network model. In the detection stage, these weight values are used to calculate similarity scores between unknown network data and the normal network model. Experimental results show that the proposed weighting method not only increases the detection rate but also reduces false alarm rate.

Manuscript received February 5, 2008
Manuscript revised February 20, 2008

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

2. Current Anomaly Detection Methods

Anomaly detection systems compute statistical models for normal network traffic and generate alarms when there is a large deviation from the normal model. Some systems have been developed, for example SPADE [10], PHAD [11] and ALAD [12]. Other techniques have been proposed as detection engines, for example using clustering and classification [13], autonomous agents and distributed intrusion detection [14], and hidden Markov model [15]. A good survey can be found in [16] and [17].

The popular network databases for system evaluation are the KDD CUP 1999 dataset [9] and the DARPA 1999 dataset [18]. A simple variant of single-linkage clustering was applied in [5] 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 [19] and [20] 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 [18]. CLAD (Clustering for Anomaly Detection) in [5] used k-NN algorithm and an unsupervised training process. CCAS [21] 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 [22] 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. Vector Quantization Modeling

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 modeling is to derive an optimal codebook which is commonly achieved by using a clustering technique [24].

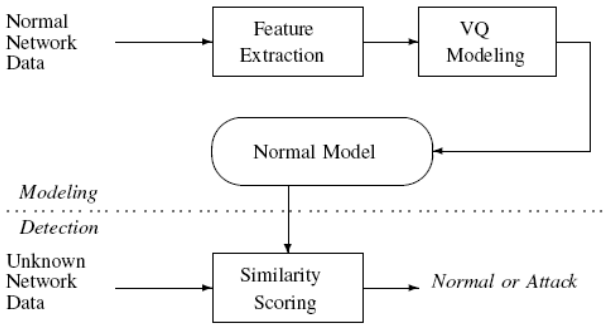

Fig. 1 Block diagram of a typical network anomaly detection system using vector quantization (VQ) modeling.

VQ modeling can be summarized as follows. Given a training set of T feature vectors $X = \{x_1, x_2, ..., x_T\}$, where each source vector $x_t = (x_{t1}, x_{t2}, ..., x_{tM})$ is of M dimensions. Let $\lambda = \{c_1, c_2, ..., c_K\}$ represent the codebook of size K, where $c_k = (c_{k1}, c_{k2}, ..., c_{kM})$, k = 1, 2, …, K are code vectors. Each code vector c_k is assigned to an encoding region R_k in the partition $\Omega = \{R_1, R_2, ..., R_K\}$. Then the source vector x_t can be represented by the encoding region R_k and expressed by

$$V(x_t) = c_k \qquad if \qquad x_t \in R_k \tag{1}$$

The codebook is built using K-means partition described as follows. Let $U = [u_{kt}]$ be a matrix whose elements are memberships of x_t in the nth cluster, k = 1, 2, …, K, t = 1, 2, …, T. A K-partition space for X is the set of matrices U such that [24]

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

where $u_{kt} = u_k(\mathrm{x}_t)$ is 1 or 0, according to whether x_t is or is not in the *k*th cluster, $\sum_{k=1}^{K} u_{kt} = 1 \; \forall t$ means each 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 $\lambda = \{\mathrm{c}_1, \mathrm{c}_2, ..., \mathrm{c}_K\}$, and d_{kt} is the Euclidean norm of $(\mathrm{x}_t - \mathrm{c}_k)$. Minimizing $J(U,\lambda;X)$ over the variables *U* and λ yields the following equations

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

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

4. Fuzzy Subspace Methods for VQ

We present the fuzzy subspace methods based on fuzzy *c*-means (FCM) and fuzzy entropy (FE) for VQ.

4.1 Fuzzy *C*-Means Subspace Modeling

Let $W = [w_1, w_2, ..., w_M]$ be the weight vector for *M* dimensions and α be a parameter weight for w_m.

The equation (3) is modified as follows

$$J_\alpha(U,W,\lambda;X) = \sum_{k=1}^{K}\sum_{t=1}^{T} u_{kt} \sum_{m=1}^{M} w_m^\alpha d_{ktm}^2 \tag{6}$$

where $\alpha > 1$, d_{ktm} is the *m*th component distance of the distance d_{kt} between x_t and c_k

$$d_{ktm}^2 = (c_{km} - x_{tm})^2, \qquad d_{kt}^2 = \sum_{m=1}^{M} w_m^\alpha d_{ktm}^2 \tag{7}$$

and weight values satisfy the following conditions:

$$0 \le w_m \le 1 \; \forall m, \quad \sum_{m=1}^{M} w_m = 1 \tag{8}$$

The basic idea of the fuzzy *c*-means subspace *K*-Means-based (FCMS-KM) VQ 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.

The FCMS-KM VQ modeling algorithm is summarized as follows:

1. Given a training data set $X = \{\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_T\}$, where $\mathrm{x}_t = (x_{t1}, x_{t2}, ..., x_{tM})$, $t = 1, 2, \ldots, T$.
2. Initialize memberships u_{kt}, $1 \le t \le T$, $1 \le k \le K$, at random satisfying (2)
3. Initialize weight values w_m, $1 \le m \le M$ at random satisfying (8)
4. Given $\alpha > 1$ and $\varepsilon > 0$ (small real number)
5. Set $i = 0$ and $J_\alpha^{(i)}(U,W,\lambda;X)$ to a large number. Iteration:
   a. Compute cluster centers using (4)
   b. Compute distance components d_{ktm} and distances d_{kt} using (7)
   c. Update weight values
   $$w_m = \frac{1}{\sum_{n=1}^{M} (D_m^2 / D_n^2)^{1/(\alpha-1)}}$$
   $$D_m^2 = \sum_{k=1}^{K}\sum_{t=1}^{T} u_{kt} d_{ktm}^2 \tag{9}$$
   d. Update membership values using (5)
   e. Compute $J_\alpha^{(i+1)}(U,W,\lambda;X)$ using (6)
   f. If
   $$\frac{J_\alpha^{(i+1)}(U,W,\lambda;X) - J_\alpha^{(i)}(U,W,\lambda;X)}{J_\alpha^{(i+1)}(U,W,\lambda;X)} > \varepsilon \tag{10}$$
   set $J_\alpha^{(i)}(U,W,\lambda;X) = J_\alpha^{(i+1)}(U,W,\lambda;X)$, $i = i + 1$ and go to step (a).

4.2 Fuzzy Entropy Subspace Modeling

Let $W = [w_1, w_2, ..., w_M]$ be the weight vector for *M* dimensions and β be a parameter weight for w_m. The equation (3) is modified as follows

$$J_\beta(U,W,\lambda;X) = \sum_{k=1}^{K}\sum_{t=1}^{T} u_{kt} \sum_{m=1}^{M} w_m d_{ktm}^2 + \beta \sum_{m=1}^{M} w_m \log w_m \quad (11)$$

where $\beta > 0$, d_{ktm} is the mth component distance of the distance d_{kt} between x_t and c_k

$$d_{ktm}^2 = (c_{km} - x_{tm})^2, \qquad d_{kt}^2 = \sum_{m=1}^{M} w_m d_{ktm}^2 \quad (12)$$

The basic idea of the fuzzy entropy subspace *K*-means-based (FES-KM) VQ method is to minimize $J_\beta(U,W,\lambda;X)$ 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.

The FES-KM VQ algorithm is summarized as follows

1. Given a training data set $X = \{x_1, x_2, ..., x_T\}$, where $x_t = (x_{t1}, x_{t2}, ..., x_{tM})$, t = 1, 2,…, T.
2. Initialize memberships u_{kt}, $1 \le t \le T$, $1 \le k \le K$, at random satisfying (2)
3. Initialize weight values w_m, $1 \le m \le M$ at random satisfying (8)
4. Given $\beta > 0$ and $\varepsilon > 0$ (small real number)
5. Set i = 0 and $J_\beta^{(i)}(U,W,\lambda;X)$ to a large number.

   Iteration:
   a. Compute cluster centers using (4)
   b. Compute distance components d_{ktm} and distances d_{kt} using (12)
   c. Update weight values

$$w_m = \frac{e^{-D_m^2/\beta}}{\sum_{n=1}^{M} e^{-D_n^2/\beta}}$$

$$D_m^2 = \sum_{k=1}^{K}\sum_{t=1}^{T} u_{kt} d_{ktm}^2 \quad (13)$$

   d. Update membership values using (5)
   e. Compute $J_\beta^{(i+1)}(U,W,\lambda;X)$ using (6)
   f. If

$$\frac{J_\beta^{(i+1)}(U,W,\lambda;X) - J_\beta^{(i)}(U,W,\lambda;X)}{J_\beta^{(i+1)}(U,W,\lambda;X)} > \varepsilon \quad (10)$$

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

4.3 Network Anomaly Detection

Assuming λ is the *normal* model. Given an unknown network feature vector x, the task is to determine x is normal or intrusive. The following algorithm is proposed

1. Given an unknown network feature vector x and the *normal* model λ
2. Set a threshold value θ
3. Calculate the minimum distance between x and λ

$$d_{\min} = \min_k d(x, c_k) \quad (15)$$

   where $d(.)$ is defined in (7) or (12) and c_k is the kth code vector in λ.

4. If $d_{\min} < \theta$ then x is normal else x is intrusive

It can be seen that when the threshold value increases, the anomaly detection rate and the false alarm rate also increase. If the false alarm rate is fixed, we can determine the corresponding values for the threshold value and the anomaly detection rate.

5. Experimental Results

5.1 Network Data and Attack Types

We consider a sample dataset, which is the KDD CUP 1999 dataset [9]. The raw network traffic records have already been converted into vector format. Each feature vector consists of 41 features. The meanings of these features can be found in [9]. In this paper, we ignore features with symbolic values. The attacks listed in feature vectors of KDD CUP 1999 dataset come from MIT Lincoln intrusion detection dataset web site [18]. The labels are mostly the same except a few discrepancies. 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.

5.2 Anomaly Detection and False Alarm 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. For training, the number of feature vectors for training the *normal* model was set to 5000. The testing data set contains 60593 feature vectors for the *normal* network pattern, and 224385 feature vectors for all attacks.

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

$$x'_{tm} = \frac{x_{tm} - \mu_m}{\sigma_m}, \quad \sigma_m = \frac{1}{T}\sum_{t=1}^{T} | x_{tm} - \mu_m | \quad (16)$$

where x_{tm} is the mth feature of the tth feature vector, μ_m the mean value of all T feature vectors for feature m, and σ_m the mean absolute deviation.

Table 1. Anomaly detection results (in %), where α = 4.0 and β = 2.0. All network data were normalized. Codebook size = 4

Modeling	False Alarm Rate (in %)				
	0.0	0.1	1.0	10.0	100.0
KM VQ	45.6	46.1	46.7	48.4	77.4
FES-KM VQ	50.8	56.2	57.0	60.1	83.6
FCMS-KM VQ	98.0	98.1	98.3	98.4	98.8

Table 2. Anomaly detection results (in %), where α = 4.0 and β = 2.0. All network data were normalized. Codebook size = 8

Modeling	False Alarm Rate (in %)				
	0.0	0.1	1.0	10.0	100.0
KM VQ	45.9	50.8	54.2	60.3	79.6
FES-KM VQ	54.7	79.8	83.4	84.1	88.6
FCMS-KM VQ	98.2	98.3	98.3	98.5	98.9

Table 3. Anomaly detection results (in %), where α = 4.0 and β = 2.0. All network data were normalized. Codebook size = 16

Modeling	False Alarm Rate (in %)				
	0.0	0.1	1.0	10.0	100.0
KM VQ	64.9	81.2	82.1	83.3	94.8
FES-KM VQ	66.8	81.7	84.0	84.3	95.3
FCMS-KM VQ	98.8	98.9	98.9	98.9	99.2

Table 4. Anomaly detection results (in %), where α = 4.0 and β = 2.0. All network data were normalized. Codebook size = 32

Modeling	False Alarm Rate (in %)				
	0.0	0.1	1.0	10.0	100.0
KM VQ	83.5	84.7	86.5	87.0	95.0
FES-KM VQ	84.3	85.1	86.6	87.0	95.4
FCMS-KM VQ	98.9	99.0	99.0	99.0	99.3

Anomaly detection rates versus false alarm rates are presented in Tables 1, 2, 3, and 4 where the codebook size is set to 4, 8, 16, and 32, respectively. We chose 5 false alarm rates (in %) which were 0.0, 0.1, 1.0, 10.0, and 100.0 to compare the corresponding anomaly detection rates for the K-means VQ modeling and the two proposed FCMS-KM VQ and FES-KM VQ modeling. The ideal value for false alarm rate is 0.0, and from the four tables, we can see that the FES-KM VQ performed better than the KM VQ modeling and the FCMS-KM VQ modeling outperformed the KM VQ modeling. Moreover, the FCMS-KM VQ modeling achieved very high detection rates even with the smallest codebook size.

All the considered methods could not achieved the highest anomaly detection rate of 100% even though we changed the threshold value to accept all attack patterns (i.e., the false alarm rate is 100%). With codebook size of 32, the FCMS-KM VQ modeling achieved very good results even with the lowest false alarm rate. The training data set contained 5000 feature vectors.

5. Conclusion

We have proposed two automated feature weighting methods based on fuzzy *c*-means and fuzzy entropy modeling 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 methods. The fuzzy *c*-means subspace VQ modeling outperformed the standard *K*-means vector quantization modeling. For further investigation, we will consider other automated weighting subspace methods that can assign different weights to clusters even in the same dimension.

References

[1] Y. Yasami, M. Farahmand, V. Zargari, "An ARP-based Anomaly Detection Algorithm Using Hidden Markov Model in Enterprise Networks", Second International Conference on Systems and Networks Communications, 2007, pp. 69 - 75

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

[3] 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.

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

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

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

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

[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] Stanifor, Hoagland and McAlerney, "Practical Automated Detection of Stealthy PortScans", Journal of Computer Security, 2002, vol. 10, no. 1, pp. 105-136

[11] M. V. Mahoney and P.K. Chan, "PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic", Technical report, Florida Tech., CS-2001-4, 2001
[12] M. Mahoney, "Network Traffic Anomaly Detection Based on Packet Bytes", Proc. ACM. Symposium on Applied Computing, 2003, pp. 346-350
[13] H. Yang, F. Xie, and Y. Lu, "Clustering and Classification Based Anomaly Detection", Lecture Notes in Computer Science, 2006, vol. 4223, pp. 1611-3349.
[14] 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.
[15] D. Ourston, S. Matzner, et al., "Coordinated Internet attacks: responding to attack complexity", Journal of Computer Security, 2004, vol. 12, pp. 165-190.
[16] 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.
[17] 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
[18] DARPA Intrusion Detection Evaluation Data Sets 1999, available at http://www.ll.mit.edu/IST/ideval/data/data_index.html
[19] 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.
[20] 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.
[21] 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.
[22] C. Caruso and D. Malerba, "Clustering as an add-on for firewalls", Data Mining, WIT Press, 2004.
[23] 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.
[24] 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.
[25] 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.
[26] D. Tran and W. Wagner, "Fuzzy entropy clustering", in Proceedings of FUZZ-IEEE Conference, 2000, vol. 1, pp. 152-157.
[27] 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.
[28] 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.

Dr. Dat Tran received the B.S. and M.S. degrees in Theoretical Physics from University of Science (US) in 1984 and 1994, respectively. During 1984-1995, he stayed in US as a lecturer and researcher. He moved to Australia in 1995 and received Graduate Diploma and Ph.D. degree in Information Sciences and Engineering from University of Canberra in 1996 and 2001, respectively. He was awarded an internship to work at IBM Watson Research Center, New York, from June 2000 to Sep 2000. Currently, he is senior lecturer at University of Canberra, Faculty of Information Sciences and Engineering, Australia. His research interests include biometrics authentication, network intrusion detection, spam email filtering, pattern recognition, and fuzzy set theory.

Dr. Wanli Ma received Ph.D. degree from the Australian National University, Australia, in 2001. He was lecturer at Department of Computer Science and Engineering, University of Science and Technology of China, Hefei, PR China from 1988 to 1993. From 1996 to 2003, he was team leader and senior systems administrator at Computer Services Center, Client Services Division, University of Canberra, Australia. Currently, he is a senior lecturer of Faculty of Information Sciences and Engineering, University of Canberra. His research interests include computer security (intrusion detection, biometrics, and computer forensics) and multi-agent system (system structure, applications, and agent-based software engineering). He also has 6 year's first hand experience in running IT infrastructure and IT security operations.

A/Prof. Dharmendra Sharma received Ph.D. degree from the Australian National University, Australia, in 1993. From 1980 to 2000, he was lecturer in Mathematics and then senior lecturer in Computer Science, University of the South Pacific, Suva, Fiji. From 2001 to 2003, he was senior lecturer in Computer Science, University of Canberra, Australia. He was Head of Software Engineering Discipline from 2001 to 2003, Head of School of Information Sciences and Engineering from 2003 to 2007, both at University of Canberra. Currently, he is Associate Professor in Information Sciences and Dean of Faculty of Information Sciences and Engineering, University of Canberra, Australia. His research interests include multi-agent systems, network security, and biometrics authentication.