STATISTICAL TEXTURE ANALYSIS METHODS FOR NETWORK TRAFFIC CLASSIFICATION

Thorsten Kisner, Alex Essoh and Firoz Kaderali
Department of Communication Systems
Faculty of Mathematics and Computer Science
FernUniversität in Hagen, Germany
{thorsten.kisner,alex.essoh,firoz.kaderali}@fernuni-hagen.de

ABSTRACT
Traffic modeling and classification is important in many areas like intrusion detection, anomalous traffic detection, network planning or bandwidth management. A novelty in the work presented in this paper is the use of texture classification methods from the domain of digital image processing for network traffic classification. We use strategies based on Co-occurence Matrices to derive statistical properties for network traffic classification. Using the well known kNN-Classificator we are able to distinguish different classes with a high probability.

KEY WORDS
Traffic Engineering, Network Traffic Classification, Intrusion Detection, Anomaly Detection

1 Introduction

The issue of IT-Awareness has been of great interest in the last few years. Not surprisingly when the dramatic increase of incidents and vulnerabilities over the past ten years is taken into consideration. Therefore, many organisations try to protect themselves by enforcing security guidelines or even a security policy for their network. In most cases there is a security officer (network administrator) responsible for the implementation of this security policy. As an example the firewall policy of an organisation could be to block diverse ports: the telnet port (23), the standard port where the MySQL server runs (3306) and common P2P ports or only allow incoming traffic on Port 80. This solution only provides limited protection with regard to enforcing security policies for several reasons. Firstly, malicious users are able to bypass the firewall by configuring their server to listen to port 80 although the service offered is not HTTP and forbidden. Secondly using HTTP Tunneling, encapsulated in HTTP packets, data packets of applications which are normally not permitted can reach their destination by bypassing the firewall and violating the security policy of the organisation. Since many security officers responsible for network administration mainly use simple port analysis to detect applications which are normally not permitted, they are not able to detect the security violations described in the first two scenarios as well as some others (see [1] [2] for more). When it comes to the detection of traffic which is not allowed, a simple port analysis cannot really help as has been confirmed in several studies carried out in the past [3] [4]. A signature based Intrusion Detection System such as Snort (www.snort.org), can be used to access the content of the packet and highlight the differences between port indication and the content as such. But such systems only work well on links where the speed is not too high. Furthermore, through the increase in use of cryptographic algorithms to protect applications, the payload of the packets and even the header cannot be accessed. In those cases the security officer is ”blind” and has no chance to detect applications carried by the packets. To address this problem, diverse methods have been proposed with the aim of identifying the service related to traffic flow (traffic classification).

The idea behind traffic classification is, that each application has a profile, an extensive analysis of this profile can be used to identify applications running on top of the transport protocols. It has been shown that port based, but also payload based analysis [3] [5] [6] have their limitations with regard to application identification. Therefore, transport layer statistics ([4] [7] [8] [9]) have been used to build traffic profiles and diverse methods (see section 2) for the classification of traffic. Hereby, a differentiation has to be made with regard to which information is used to perform the classification: header, payload or just metainformation based on header and or payload.

In this paper we introduce a method to classify encrypted network traffic using statistical texture analysis methods. We analyse time-aggregated network traffic with a co-occurrence matrix [10] [11] and related statistical metrics to determine both HTTP and SMTP traffic. Since they considerably differ from each other, traffic can either be classified as HTTP or SMTP by using one or several of these parameters. There has not been an application of statistical texture analysis methods, specifically co-occurrence matrices and related parameters for network traffic classification to date. Indeed Mizuki et al. [12] proposes a method for the detection of masqueraders using the so called *Eigen Co-occurrence Matrices*, but the detection method works on Unix commands and not on network packets.

The rest of this paper is organised as follows: in section 2 we summarise related work. Section 3 gives an overview of texture analysis methods with co-occurrence

matrices. In section 4 we present the results of our proposal and in section 5 we summarise and provide direction for future work.

2 Related Work

In the past diverse approaches have been proposed to deal with traffic classification. Auld et al. [9] extended the work done by Moore et al. [4] by using a Bayesian Neural Network classificator. More than two hundred features are extracted from the TCP traffic flows, normalised between 0 and 1 and used as an input to a neural network (Multilayer Perceptron). The neural network then provides the probability of a traffic flow to belong to one of 10 traffic classes. The classification is carried out without access to the contents of the packets. Moreover, neither IP address nor port information is taken into consideration. Divakaran et al. [8] use the concept of a packet train which was introduced by Jain [13]. A packet train is a socket connection between two hosts and consists of packet train length (number of packets in a socket communication) and packet train size (total number of bytes being transmitted in the socket connection). These two parameters are used to model the traffic flow using Vector Quantisation (VQ) and Gaussian Mixture Models (GMM). The k-Nearest Neighbors (kNN) algorithm is used to classify the traffic flow as belonging to one of the following standard protocols (HTTP, SMTP, DNS, SSH and POP3).

Bernaille et al. [14] follow a completely different approach. They argue that in a traffic flow, the size of the first p packets is characteristic of each application. Their method works in two phases: In the offline phase each TCP flow (consisting of the first p packets) is mapped to a p dimensional vector. Hereby, the size of each of the p packets in the TCP traffic stream is represented by one dimension. The similarity between flows is then given by the Euclidean distance and the k-Means algorithm is used to build traffic classes. To classify a new flow (online phase) the Euclidean distance between the new flow and the center of each of the cluster is calculated and the flow is classified as belonging to the application where the distance is minimum. They only need the first five packets to establish the relation between the TCP flow and the application. Apart from the DBSCAN[1] algorithm, Erman et al. [15] also use the clustering algorithm k-Means and the Euclidean distance as metric for classification. Further classification methods are: a decision tree classifier [2] for the classification of application protocols based on the analysis of the TCP state flags, Hidden Markov Models (HMM) in [16] for encrypted applications based on features which remain intact after encryption and finally a rule based classification method relying on a predifined set of algorithms to determine a large class of backdoors operating on non standard ports [1].

In contrast to the related work we point out that we do not have access to single packet information. We only estimate the amount of data transmitted in a given time interval and introduce texture analysis methods to network traffic classification.

[1] Density-Based Spatial Clustering of Applications with Noise

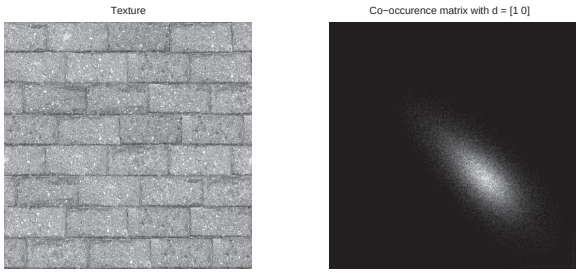

Figure 1. Texture with brightness coded GLCM

3 Grey Level Co-occurrence Matrix

One challenge in digital image processing is the classification of textures. A sampled and quantisised digital image can be interpreted as a matrix $\mathbf{G} = g(\vec{x}) = g(n_x, n_y) \in \{0, 1, ..., N_{g-1}\}$ and the description of combinations of pixel brightness values (grey levels) in this image is called *Grey Level Co-occurrence Matrix* (GLCM) $\mathbf{C}(\delta, T) = [s(i, j, \delta, T)]$ or *Grey Tone Spatial Dependency Matrix*.

Each element $s(i, j, \delta, T)$ is a second order probability going from one grey level i to another grey level j given the displacement vector $\delta = (\Delta x, \Delta y)$. T defines a tile of the original picture. Each element in $\mathbf{C}(\delta, T)$ can be determined with

$$s(i, j, \delta, T) = \frac{\Theta\{\vec{x}|\vec{x}, \vec{x} + \delta \in T, g(\vec{x}) = i, g(\vec{x} + \delta) = j\}}{\Theta\{\vec{x}|\vec{x}, \vec{x} + \delta \in T\}} \tag{1}$$

where Θ denotes the number of elements in each set [11]. The dimension of $\mathbf{C}(\delta, T)$ is $N_g \mathrm{x} N_g$.

Figure 1 shows the co-occurrence matrix for the corresponding texture. Elements along the diagonal of the matrix represent neighboring pixel pairs with less or no difference in the grey level. The farther away from the diagonal the higher the grey level difference becomes.

Haralick et al. proposed 14 criteria extracted from the GLCM to describe a texture [10] and used them as an input vector for a classificator. Conners et al. pointed out six significant parameters from the original 14 in [11] and we use the *Correlation* (8) as a seventh parameter. The parameters with $\sigma_i = \sum_i \sum_j (i - \mu_i)^2 \cdot s(i, j)$ and $\sigma_j = \sum_i \sum_j (j - \mu_j)^2 \cdot s(i, j)$ are defined as follows:

$$ASM = \sum_i \sum_j (s(i, j))^2 \tag{2}$$

$$ENT = -\sum_i \sum_j s(i, j) \cdot \log(s(i, j)) \tag{3}$$

$$IDM = \sum_i \sum_j \frac{s(i,j)}{1+(i-j)^2} \quad (4)$$

$$INE = \sum_i \sum_j (i-j)^2 \cdot s(i,j) \quad (5)$$

$$CS = \sum_i \sum_j ((i-\mu_i)+(j-\mu_j))^3 \cdot s(i,j) \quad (6)$$

$$CP = \sum_i \sum_j ((i-\mu_i)+(j-\mu_j))^4 \cdot s(i,j) \quad (7)$$

$$CORR = \sum_i \sum_j \frac{(i-\mu_i)(j-\mu_j) \cdot s(i,j)}{\sigma_i \cdot \sigma_j} \quad (8)$$

The variable μ_i is defined as $\mu_i = \sum_i \sum_j i \cdot s(i,j)$ and $\mu_j = \sum_i \sum_j j \cdot s(i,j)$.

The *Angular Second Moment* (2) describes the energy of the matrix and the *Entropy* (3) reflects the information content. *Inertia* (5) can be interpreted as a contrast to the greyscale image and *Inverse Difference Moment* (4) as an inverse weighted measure of contrast. *Cluster Shade* (6) describes spots with homogene intensity and a high contrast to the remaining structure, the grey level of clusters are characterised by *Cluster Prominence* (7).

4 Analysing Network Traffic with GLCM Parameters

In our work we examine the in- and outgoing network traffic of two different servers. Both are running *SuSE Linux 9.2* and are dedicated to one service only. The first is an SMTP-Server running *Exim 4.60*, the second is an HTTP-Proxyserver with *Squid 2.3*. The network traffic is measured at the gateway to the external network with the built-in packet and byte counter of `iptables`. The traffic was measured in a period of 14 weeks at weekdays between 7:30am and 4:30pm resulting in 70 independent traces of 9 hours for each type of traffic. A typical time series for network traffic is shown in figure 2. There are a few peaks of network traffic and several sections of burst traffic as well as sections with less or no traffic.

Like the windowing mechanism (see T in eq. (1)) in the texture analysis we divide each 9 hour time series in 6 segments of 90 minutes each resulting in a total of 840 segments for further analysis. This is the training set for our research, another measurement of 10 days and 120 segments is independent of the training set and will verify our results.

We transfer the mechanisms of analysing textures to classification of network traffic. Our basic data are one dimensional time series of throughout measurements in contrast to two dimensional images in texture analysis.

In the scope of texture analysis this two dimensional aspect is taken into consideration in the displacement vector to generate the co-occurrence matrix. A rotation invariant co-occurrence matrix is computed by averaging several GLCMs which are based on different displacement vectors

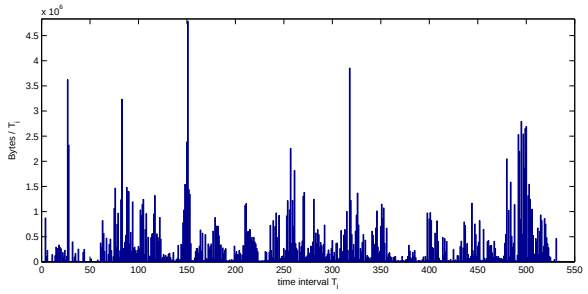

Figure 2. Typical network traffic time series

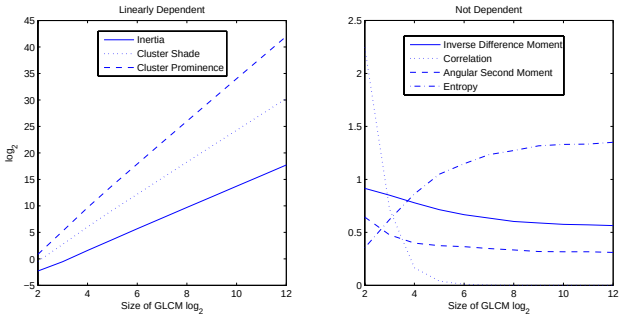

Figure 3. Parameters as a function of matrix size

(horizontal, vertical, diagonal-up and diagonal-down). In our case the direction of the vector is implicitly given by the time, thus we have only one displacement vector along the time scale representing the difference between two successive time intervals.

In the following we discuss the necessary steps to transfer the texture analysis technique to network traffic and present the results of the calculated GLCM paramters of our datasets. We will then discuss some statistical approaches to extract pattern from the calculated characteristics.

4.1 Determining the GLCM matrix size

In the case of texture analysis the size of the co-occurrence matrix is explicitly given by the range of the greyscale values, e.g. a 8-bit greyscale image results into a matrix size of 2^8x2^8. In our scenario the source for the co-occurrence is a time series with no explicitly given limit for the values, the limit is given by the bandwidth of the measured link. Without normalisation this would result in a huge matrix size, in addition to this, a peak in the measured data would also increase the size. A peak of 10 MByte/minute e.g. will result in a matrix size to the magnitude of 10^7x10^7, which does not make sense thus requiring quantisation.

We analysed a linear quantisation to a matrix size of 2^i with $i \in \{2,3,...,12\}$. Figure 3 shows the computed parameters (eq. 2 to 8) as a function of the matrix size. On the left side we see a linear dependency of the values *Inertia* (5), *Cluster Shade* (6) and *Cluster Prominence* (7) to

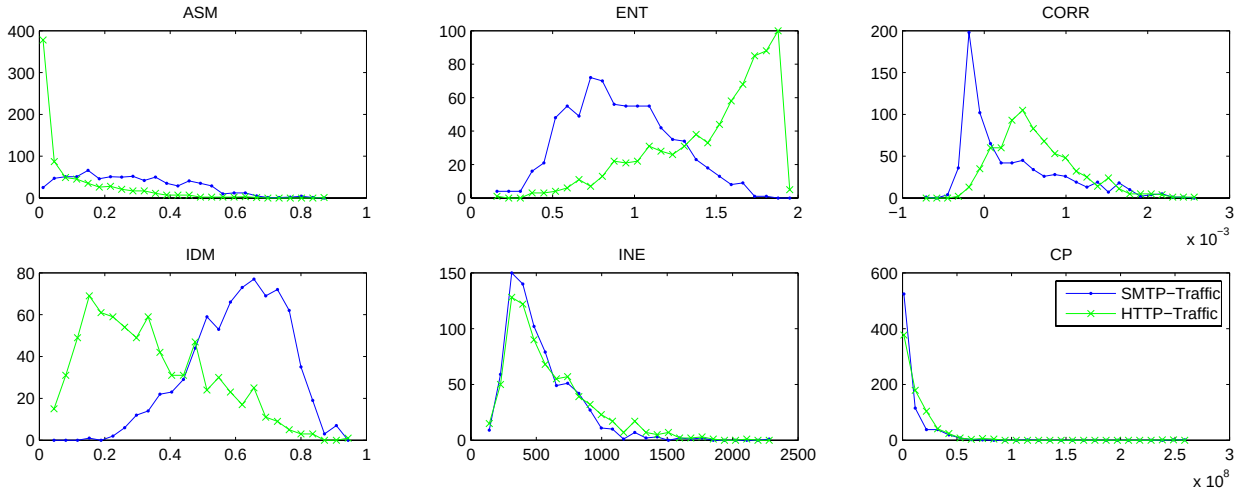

Figure 4. Histograms of selected parameters

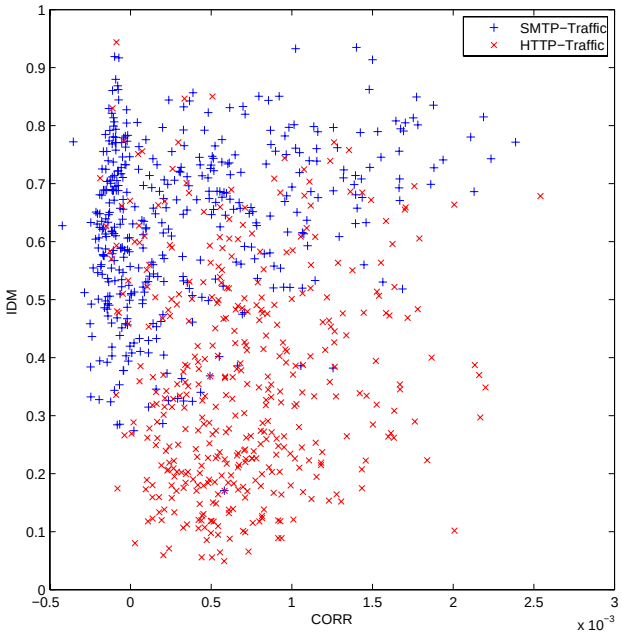

Figure 5. Plot of *IDM* and *CORR*

the matrix size. From this, it follows that the distribution of these three parameters are independent of the size of the co-occurrence matrix. The other values *Inverse Difference Moment* (4), *Correlation* (8), *Angular Second Moment* (2) and *Entropy* (3) do not show such a simple correlation in the diagram of figure 3 on the right. But for values higher than 6 you can see a nearly constant graph for all parameters. With regard to fast computation we have chosen a matrix size of 2^7 for all further computations.

4.2 Evaluation of feature vectors

Figure 4 shows the histogram of our training dataset for parameters described in section 3. The histograms for *Inertia* (5) and *Cluster Prominence* (7) (as well as for *Cluster Shade* (6), which is not printed in the figure) show nearly the same structure for both traffic types and thus cannot be used for classification.

The other diagrams (*ASM*, *ENT*, *CORR* and *IDM*) expose a significant difference between SMTP and HTTP traffic which is used in the following sections to categorise the type of network traffic. Table 1 shows the mean μ and the standard deviation σ of both SMTP and HTTP datasets. With σ nearly equal or even higher than μ it is obvious that the parameters *Cluster Prominence* and *Cluster Shade* are not a good choice for classification.

As an example the parameters *Inverse Difference Moment* and *Correlation* are plotted against each other in figure 5. Although there is an intersection of both classes (SMTP and HTTP) a clustering of each class can be observed.

For further analysis we use the Principal Components Analysis to extract the significant features.

4.3 Principal Components Analysis

Very often when conducting research, more than one variable has to be analysed and very often situations arise where the data to be analysed is simply too large (large data matrix). Since in many areas, the processing time is an issue, it would be desirable only to eliminate variables which are irrelevant, because the processing complexity grows with the number of variables to be analysed. The goal of the Principal Component Analysis is to accomplish this task, to reduce the dimension of the data by just taking relevant variables (so called principal components) into account and making sure that the loss is low.

	SMTP		HTTP	
Parameter	μ	σ	μ	σ
Angular Second Moment	$0,301$	0.174	0.113	0.135
Entropy	0.935	0.315	1.501	0.374
Correlation	$0.366 \cdot 10^{-3}$	$0.581 \cdot 10^{-3}$	$0.701 \cdot 10^{-3}$	$0.484 \cdot 10^{-3}$
Inverse Difference Moment	0.633	0.139	0.360	0.186
Intertia	567.14	270.7	619.29	313.07
Cluster Shade	$87.323 \cdot 10^3$	$80.626 \cdot 10^3$	$107.71 \cdot 10^3$	$89.92 \cdot 10^3$
Cluster Prominence	$13.108 \cdot 10^6$	$17.479 \cdot 10^6$	$17.037 \cdot 10^6$	$18.245 \cdot 10^6$

Table 1. First order statistics

Let **M** be a matrix of observations with p being the number of variables to be analysed and n the number of observations for each variable. **M** can be written as follows

$$\mathbf{M} = \begin{bmatrix} \mathtt{x_{11}} & \mathtt{x_{12}} & \mathtt{x_{13}} & \dots & \mathtt{x_{1p}} \\ \mathtt{x_{21}} & \mathtt{x_{22}} & \mathtt{x_{23}} & \dots & \mathtt{x_{2p}} \\ \mathtt{x_{31}} & \mathtt{x_{32}} & \mathtt{x_{33}} & \dots & \mathtt{x_{3p}} \\ \dots & & & & \\ \mathtt{x_{n1}} & \mathtt{x_{n2}} & \mathtt{x_{n3}} & \dots & \mathtt{x_{np}} \end{bmatrix} \tag{9}$$

Let **X** be the multivariate described by the matrix above. To characterise this multivariate we need to determine the mean and the so called variance-covariance matrix of the variable. The mean $\bar{X}$ of the multivariate X is

$$\bar{X} = \left[\bar{X}_1, \bar{X}_2, \bar{X}_3, \bar{X}_4, \dots \bar{X}_p\right] \tag{10}$$

The variance-covariance matrix **S** can be calculated as follows [17]

$$\mathbf{S} = \frac{1}{(n-1)} \cdot \sum_{i=1}^{i=n} (X_i - \bar{X})(X_i - \bar{X})^t, \tag{11}$$

Hereby X_i is a row vector of the matrix **M**. To determine the principal component of the variance-covariance matrix, at first the eigenvalues and related eigenvectors are determined. To determine the eigenvalues of **S**, the linear equation (12) must be solved.

$$det(\mathbf{S} - \lambda \cdot \mathbf{I}) = 0 \tag{12}$$

When the values have been determined, we can proceedfurther and determine the vectors by solving the following equation.

$$(\mathbf{S} - \lambda \cdot \mathbf{I}) \cdot \vec{e} = \vec{0} \tag{13}$$

After calculating the eigenvalues and the eigenvectors, the next step is to construct a matrix **T** composed of eigenvectors, with columns of the matrix as the eigenvectors, whereby the eigenvectors with the k largest eigenvalues are chosen. In [18] criteria for the choice of the new dimensions k have been specified. The relation between the original Matrix **X** and the principal component matrix **Z** is

$$\mathbf{Z} = \mathbf{T} \cdot \mathbf{X} \tag{14}$$

Traffic	Positive	Negative	Classification rate
HTTP	55	5	91.67%
SMTP	52	8	86.67%
Total	107	13	89.17%

Table 2. Accuracy of classification

The error related to the dimension reduction equals:

$$e = \frac{1}{2} \sum_{i=k+1}^{i=n} \lambda_i \tag{15}$$

4.4 Classification of network traffic

We use the k-Nearest-Neighbors (kNN) algorithm with $k = 5$ to classify the 60 segments of each SMTP and HTTP traffic and to estimate the classification rate of our proposal. To get reasonable results for the Euclidian distance we transform each feature vector to the same standard deviation of $\sigma = 1$ and only use the four most relevant parameters (*Angular Second Moment* (2), *Entropy* (3), *Inverse Difference Moment* (4) and *Inertia* (5)).

Table 2 shows the results: The classification accuracy of HTTP is about 91.67% and slightly higher than that of SMTP with 86.67% resulting in a total classification accuracy of 89.17%.

5 Conclusion and Future Work

We have presented a novel approach for identifying network traffic. Using statistical texture analysis methods we were able to map the given time series into the known co-occurrence matrix. Even with the inaccurate available data on time series with one minute resolution, we show that it is possible to classify different types of network traffic. We demonstrated the use of the kNN algorithm to classify data with an accuracy of 90%.

For future work we will analyse the time series at different time scales (with a different time interval) and in-

clude other parameters like the number of packets per time interval to obtain a multi-dimensional time series.

We also intend to further examine network traffic with the proposed method on packet level also including network flow information. Due to the increasing rate of Peer-to-Peer traffic this type of traffic will be investigated in detail as well as superposed traffic.

References

[1] Y. Zhang and V. Paxson, Detecting Backdoors, *Proc. 9th USENIX Security Symposium*, Denver, Colorado, August 2000, 157-170.

[2] J. P. Early, C. E. Brodley and C. Roseberg, Behavioral Authentication of Server Flows, *Proceedings of the 19th Annual Computer Security Applications Conference*, Las Vegas, USA, December 8-12 2003, 49-55.

[3] T. Karagiannis, K. Papagiannaki, M. Faloutsos, BLINC: multilevel traffic classification in the dark, *ACM SIGCOMM*, Philadelphia, USA, August 2005.

[4] A.W. Moore, D. Zuev, Internet Traffic Classification Using Bayesian Analysis Techniques, *Proceedings of the 2005 ACM SIGMETRICS*, Banff, Canada, June 6-10 2005, 50-60.

[5] A. W. Moore, K. Papagiannaki, Toward the Acurate Identification of Network Applications, *In PAM 2005*, Boston, USA, March 31-April 1 2005.

[6] S. Sen, O. Spatscheck, D. Wang: *Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures*, In WWW 2004, New York, USA, May 17-22 2004.

[7] C. V. Wright, F. Monrose and G. M. Masson, Using Visual Motifs to Classify Encrypted Traffic. *To appear in Proceedings of the ACM Workshop on Visualization for Computer Security (VizSEC)*, 2006.

[8] D. M. Divakaran, H. A. Murthy and T. A. Gonsalves, Traffic Modeling and Classification Using Packet Train Length and Packet Train Size, *6th IEEE International Workshop on IP Operations and Management, IPOM-2006*, Ireland, October 2006.

[9] T. Auld, A. Moore and S. Gull, Bayesian Neural Networks For Internet Traffic Classification. *To appear in IEEE Transactions on Neural Networks* 17, November 2006.

[10] R. M. Haralick, K. Shanmugam and I. Dinstein, Textural features for image classification, *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6), November 1973, 610-621.

[11] R.W. Conners, M. M. Trivedi, C.A. Harlow, Segmentation of a High-Resolution Urban Scene using Texture Operators, *Computer Vision, Graphics and Image Processing*, 25, 1984, 273-310.

[12] O. Mizuki, O. Yoshihiro, A. Hirotake and K. Kazuhiko, Anomaly Detection Using Layered Networks Based on Eigen Co-occurence Matrix. *Recent Advances in Intrusion Detection*, Nice, France, 15-17 September 2004, 223-237.

[13] R. Jain, S. Routhier, Packet Trains-Measurements and a New Model for Computer Network Traffic, *IEEE Journal on Selected Areas in Communications*, SAC 4(6), 1986, 986-995.

[14] L. Bernaille, R. Teixeira, I. Akodjenou, A. Soule and K. Salamatian, Traffic Classification On the Fly, *Computer Communication Review*, 36 (2), April 2006, 23-26.

[15] J. Erman, M. Arlitt, A. Mahanti, Traffic Classification Using Clustering Algorithms, *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, Pisa, Italy, 2006.

[16] C. V. Wright, F. Monrose and G. M. Masson, On Inferring Application Protocol Behaviors in Encrypted Network Traffic, *To appear in the Journal of Machine Learning Research (JMLR): Special issue on Machine Learning for Computer Security*, 2006.

[17] N. Ye, Q. Chen, An Anomaly Detection Technique based on a Chi-Square Statistic for Detecting Intrusions into Information Systems, Quality and Reliability Engineering International, 17 (2) 2001), 105-112.

[18] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification*, New York, John Wiley & Sons Inc., 2000.