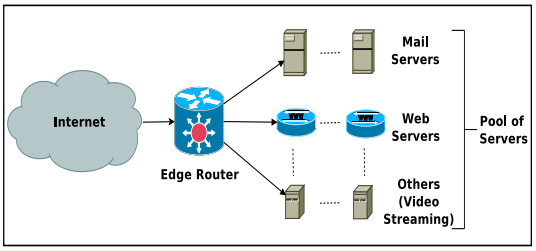
Port-based Traffic Verification as a Paradigm for Anomaly Detection

Vadiraj Panchamukhi
Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai 600036, India
Email: vadiraj@cse.iitm.ac.in

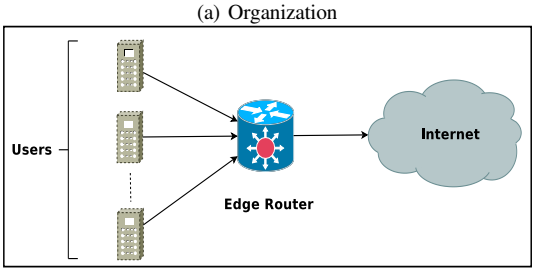
Hema A. Murthy
Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai, 600036, India
Email: hema@cse.iitm.ac.in

***Abstract*—An *anomaly* is an activity that deviates from the well-known behaviour of the system. Anomaly detection in networks is of interest from two perspectives: an organization's perspective and an Internet Service Provider's (ISP) perspective. Protection of its computer network infrastructure is an important task for all organizations. Organizations desire that their networks are robust and resilient to any kind of attack. Anomaly detection forms an important part of this network resiliency. Also the ISPs want to maximize the utilization of their resources. Hence an ISP would be interested to know any resource failure immediately so as to correct the problem. ISPs would also be interested in safeguarding their network from malicious activities. We describe here a Gaussian Mixture Model (GMM)-based *traffic verification system* as a paradigm for network anomaly detection. The traffic characteristics aggregated over a period of time is given to the model to verify the validity of the traffic. If the traffic does not obey the model then we raise an alarm flagging it as an anomaly. Our results show that the system performs with less than 1% misses and false alarms.**

***Index Terms*—Anomaly detection, Network security, Intrusion detection, Gaussian Mixture Model, Network Traffic Modeling**



(a) Organization



(b) ISP

Fig. 1: Scenarios where we can deploy the Anomaly Detection system.

I. Introduction

An *anomaly* is an activity that deviates from the well-known behaviour of the system. An anomaly need not necessarily be a malicious activity all the time. Anomaly detection in networks is of interest from two perspectives: an organization's perspective and an Internet Service Provider's (ISP) perspective. Protection of its computer network infrastructure is an important task for all organizations. Organizations desire that their networks are robust and resilient to any kind of attack. Anomaly detection forms an important part of this network resiliency. Also the ISPs want to maximize the utilization of their resources. Hence an ISP would be interested to know any resource failure immediately so as to correct the problem. ISPs would also be interested in safeguarding their network from malicious activities. We describe here a *Port-based traffic verification system* as a paradigm for network anomaly detection. We model the normal behaviour of the traffic using Gaussian Mixture Model (GMM). The traffic characteristics aggregated over a period of time is given to the model to verify the validity of the traffic. If the traffic does not obey the model then we raise an alarm flagging it as an anomaly.

Our anomaly detection system can be deployed in two scenarios as shown in Figure 1. The traffic verification system is deployed at the edge-router in both the scenarios. The traffic characteristics of various services – like HTTP, SMTP, DNS etc. are collected at the edge-router at regular time intervals (15 minutes, 30 minutes, or 60 minutes etc.) and given to the system for verifying its legitimacy. Today, most of the traffic types are tunnelled through HTTP. In this work, an attempt is made to detect the anomalies in the heterogeneous traffic that is tunnelled through HTTP. We have observed that the services like Social-networking, Webmail, Video traffic tunneled through HTTP obey certain characteristics that can be modeled. We have used GMM to model these services tunneled through HTTP and our results show that the system performs with less than 1% misses and false alarms.

It is shown that various types of traffic obey unique characteristics [1], [2]. Both these methods have used different parameters for traffic modeling. [2] have modeled the traffic with the technique Vector Quantization (VQ) and have used packet length, packet interarrival time, connection duration, connection packet count and connection byte count as the

parameters. Where as [1] have shown that GMM performs better over VQ and have used packet train length and packet train size as the parameters to model the traffic. We build on [1] and show that GMM can be used in *verification framework* to detect anomalies.

Rest of the paper is organized as follows. The *verification framework* that we propose to detect the network anomalies is discussed in detail in Section II. The related works in network anomaly detection are listed in Section III. Section IV gives experimental observations. Finally, conclusion and future work are presented in Section V.

II. Verification framework for network anomaly detection

Various services offered over Internet run on standard ports – e.g. HTTP on port 80, SMTP on port 25 etc. and these ports have to be kept open for access by the users over Internet. This exposes the server to a potential attack through these ports. We have attempted in this work to model the normal behaviour of the traffic that flows through a specific port. Now the anomaly detection is reduced to a simple *verification* task where we verify the traffic through a port is indeed a type of service offered through that port. This framework is extensively used in biometric authentication systems and in speaker verification systems [3] to verify whether an individual is the person who he/she claims to be. In this work, an attempt is made to detect the network anomalies using this framework. The framework and the components of our anomaly detection system are explained in the subsequent subsections.

A. Likelihood ratio-based traffic verification system

Let α be the network traffic pattern observed during a time interval, and let T be the hypothesized model of a particular traffic type (like HTTP etc.). The task of the anomaly detection is reduced to verifying whether α is indeed of traffic type T or *not*. Here we assume that α contains traffic pattern of a single traffic type. The problem of anomaly detection can also be viewed as a classification problem [4] in pattern recognition with number of classes being 2. The major challenge here is to minimize misses and false alarms.

The task of anomaly detection can be expressed in basic hypothesis-test terms as below:
Let

H_0: α is of traffic type T

and

H_1: α is ***not*** of traffic type T i.e. an **Anomaly**.

The likelihood-ratio test can be used to decide on either of these hypotheses as

$$\frac{p(\alpha|H_0)}{p(\alpha|H_1)} \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{reject } H_0 \end{cases} \quad (1)$$

where $p(\alpha|H_i),\ i=0$ or 1, is the probability density function indicating the probability that the traffic pattern α is from the hypothesis H_i. This is also referred to as the *likelihood* of the hypothesis H_i given the traffic pattern α. θ is the threshold to accept or reject H_0. [1] We have used Gaussian Mixture Model (GMM) to compute the likelihood values $p(\alpha|H_0)$ and $p(\alpha|H_1)$, which is explained in detail in the subsequent subsection II-B.

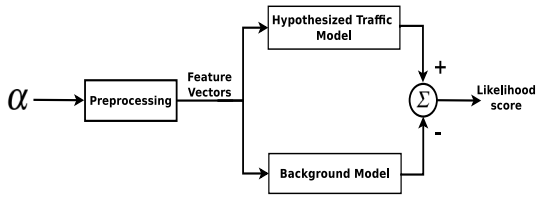


Fig. 2: Likelihood ratio-based traffic verification system

Figure 2 shows the basic components of the *verification system*. Preprocessing step takes the raw traffic data and extracts relevent features to form a feature vector that captures the information specific to each and every traffic type in the network. The format of the feature vector is explained in IV-B. The likelihood values of the hypotheses H_0 and H_1 are computed from these feature vectors.

Mathematically, H_0 is represented by λ_{hyp} that captures the characteristics of the hypothesized traffic type T. For example, if the feature vectors capturing the characteristics of H_0 are Gaussian distributed, then λ_{hyp} contains the parameters mean vector and the covariance matrix of the Gaussian distribution. Similarly if the alternative hypothesis H_1 is mathematically represented by $\lambda_{\widehat{hyp}}$, the likelihood-ratio statistic can be given as $p(Y|\lambda_{hyp})/p(Y|\lambda_{\widehat{hyp}})$, where Y is the feature vector that characterizes the given traffic pattern α. In practice, logarithm of this ratio is used to make the decision whether the traffic pattern Y (and hence α) is anomalous or not.

$$score = log\ p(Y|\lambda_{hyp}) - log\ p(Y|\lambda_{\widehat{hyp}}) \quad (2)$$

How do we go about estimating $p(Y|\lambda_{hyp})$ and $p(Y|\lambda_{\widehat{hyp}})$ is explained in sections II-B and II-C respectively.

B. Gaussian Mixture Model (GMM)

One of the important steps in the implementation of verification based systems is the choice of the likelihood function. The main approaches that can be used as a likelihood function are Gaussian mixture model (henceforth referred to as GMM) or hidden Markov model (HMM). We have used GMM to compute the likelihood value since the network traffic data in a time interval is independent of its previous time intervals and the cost of computation needed to build the model using HMM.

We use the term *transaction* to represent a trail of the access of the user over the network. For example, if the user accesses a webpage over the Internet, we say the corresponding log of this access as one *transaction*.

Let

$Q = \{$ set of all transactions in time duration $\tau\}$.

[1] Rejecting the hypothesis H_0 is same as accepting the hypothesis H_1.

$\bar{q}$ [2]: $Q \rightarrow \mathbb{R}^D$ be the mapping of each of the transactions in Q to a $D-$dimensional vector $\bar{q} = [q_1\ q_2\ \cdots\ q_D]^t$ [3].
$\bar{x}_i = \sum_{\bar{q} \in Q} \bar{q}$ be a feature vector representing the features aggregated over a time duration τ.
$\mathbf{X}$ [4]$= [\bar{x}_i], (1 \leq i \leq N)$ be the matrix representing the training examples taken over N time intervals.

For simplicity of explanation, we use λ in the remainder of this subsection to denote the hypothesis model. Given a vector $\bar{x}$, the likelihood of $\bar{x}$ belonging to λ is given by the mixture density function [5]

$$p(\bar{x}|\lambda) = \sum_{m=1}^{M} w_m\ p(\bar{x}|\Phi_m) \tag{3}$$

where M is the number of mixtures in the model. The prior probabilities w_m are called the *mixing parameters* or *mixing coefficients*. An estimate of w_m is calculated as $w_m = n_m/N$, where n_m is the number of training examples in mixture m. Further, $\sum_{m=1}^{M} w_m = 1$. $p(\bar{x}|\Phi_m)$ is the density function with the parameters Φ_m.

In GMM, the density function $p(\bar{x}|\Phi_m)$ is modeled as multivariate Gaussian with $D \times 1$ mean vector $\bar{\mu}_m$, and $D \times D$ covariance matrix $\boldsymbol{\Sigma}_m$ as parameters and is defined as

$$p(\bar{x}|\Phi_m) = p(\bar{x}|\bar{\mu}_m, \boldsymbol{\Sigma}_m) = \frac{1}{(2\pi)^{D/2}\,|\boldsymbol{\Sigma}_m|^{1/2}} e^{-\frac{1}{2}(\bar{x}-\bar{\mu}_m)^t \boldsymbol{\Sigma}_m{}^{-1} (\bar{x}-\bar{\mu}_m)} \tag{4}$$

The characteristics of the hypothesis model are captured mathematically by $\lambda = \{[w_m\ \bar{\mu}_m\ \boldsymbol{\Sigma}_m]^t\}, (1 \leq m \leq M)$. Now, estimating λ boils down to estimating the mixing coefficients w_m, mean vectors $\bar{\mu}_m$ and covariance matrices $\boldsymbol{\Sigma}_m$ for M mixtures. Though the theoretical model incorporates full covariance matrices, we have used diagonal covariance matrices as diagonal covariance matrices are computationally efficient. Expectation-maximization (EM) algorithm [6] is used to estimate λ, i.e. the parameters of the GMM. We have used the *k-means algorithm* also called as *vector quantization* in the *initialization* step of this GMM parameter estimation algorithm [4].

C. Universal Background Model

As explained in section II-A the characteristics of the hypothesized traffic type H_0 can be well captured by the model λ_{hyp} built from training traffic data of T, the same $\lambda_{\widehat{hyp}}$ for the alternative hypothesis H_1 is not well defined because $\lambda_{\widehat{hyp}}$ is supposed to capture all the characteristics of the traffic that is *not* T (i.e. anomaly). There are two main approaches for modeling the alternative hypothesis $\lambda_{\widehat{hyp}}$. The first approach is to use a set of traffic types other than T to build the hypothesis model $\lambda_{\widehat{hyp}}$. Given a set of N models of different traffic types $\{\lambda_1, \lambda_2, \lambda_3, \cdots, \lambda_N\}$, the alternative hypothesis model is given as

$$p(Y|\lambda_{\widehat{hyp}}) = f(p(Y|\lambda_1), \cdots, p(Y|\lambda_N)), \tag{5}$$

where f() is some function that represents the set of all the N traffic types. f() can be average or maximum of the likelihood values of all the N traffic types.

The second major approach to estimate alternative hypothesis model $\lambda_{\widehat{hyp}}$ is to concatenate the training data of all the N traffic types and build a single model. The hypothesis model $\lambda_{\widehat{hyp}}$ thus estimated can be used as an alternative hypothesis model for all the N traffic types. The computation cost of building the hypotheses models using this approach is also less compared to the first approach. We have used this second approach to build the alternative hypothesis model $\lambda_{\widehat{hyp}}$. This model is also referred to as universal background model (UBM). We have built the alternative hypothesis model using GMM as explained in section II-B. Hypothesized model λ_{hyp} for a particular traffic type T is then derived using the technique Adaptation explained in the section II-D.

D. Adaptation for a traffic type

In GMM-UBM system, we estimate the parameters of the hypothesized traffic model by adapting the parameters of the UBM using training examples of a particular traffic type T. We have used Maximum a posteriori (MAP) estimation as explained in [7] for this adaptation. Given the UBM $\lambda_{\widehat{hyp}}$ and the training examples of traffic type T, Algorithm-1 gives the procedure to derive the model λ_{hyp} from $\lambda_{\widehat{hyp}}$. Since we are using diagonal covariance matrices, the *step-7* in Algorithm-1 computes only the variances along every dimension. These variances in turn form the entire adapted covariance matrix.

III. Related Work

In this section, various techniques proposed in network anomaly detection literature are given. Clustering technique is used for anomaly detection in [8]. Nearest neighbour based techniques are proposed in [9], [10]. One of the earliest works [11] uses both rule-based and statistical profiling approach to detect the network anomalies. [12], [13], [14] have used spectral analysis of the data to detect anomalies in the network traffic. Factor analysis is used for anomaly detection in [15].

IV. Experimental Setup

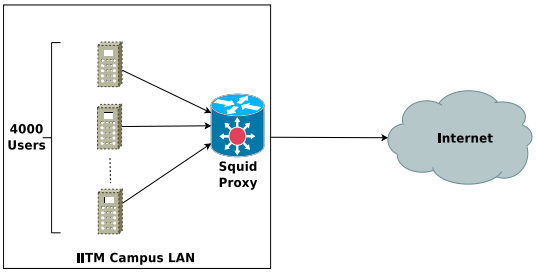


Fig. 3: Topology of the network

We have used the proxy logs of Internet access pattern of the users of Indian Institute of Technology Madras (IITM)

[2] Bar over the letter is used to denote a vector.
[3] Superscript t is used to denote the transpose of the matrix.
[4] Bold letter indicates it is a matrix.

Algorithm 1 adaptGMM($\mathbf{X}, N, M, \lambda_{\widehat{hyp}}, \rho$)

1: {$\mathbf{X}$:D-dimensional training dataset of traffic type T, N:Number of training examples in the dataset, M:Number of Mixtures, $\lambda_{\widehat{hyp}}$:UBM consisting of $[\, w_m \ \bar{\mu}_m \ \boldsymbol{\Sigma}_m \,]^t, (1 \le m \le M)$, ρ:The relevence factor (Usually initialized to 16.0 }
2: **for** $i = 1 \rightarrow N$ **do**
3: Assign $\bar{x}_i$ to cluster m as

$$label(\bar{x}_i) = \underset{m}{maxval} \frac{w_m p(\bar{x}_i|\bar{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^{M} w_j p(\bar{x}_i|\bar{\mu}_j, \boldsymbol{\Sigma}_j)}, (1 \le m \le M)$$

4: **end for**
5: Estimate η_m, the number of training examples $\bar{x}_i$ belonging to cluster $m, (1 \le m \le M)$ from $label(\bar{x}_i), (1 \le i \le N)$
6: Compute $\widehat{\bar{\mu}}_m$ =mean vector and $\widehat{\boldsymbol{\Sigma}}_m$ =covariance matrix of all $\bar{x}_i$ belonging to mixture $m, (1 \le m \le M)$
7: Compute the adapted parameters as

$$\bar{w}_m^{adapted} = (\beta_m \eta_m / N + (1 - \beta_m) w_m) \gamma$$

$$\bar{\mu}_m^{adapted} = \beta_m \widehat{\bar{\mu}}_m + (1 - \beta_m) \bar{\mu}_m$$

$$(\sigma_{md}^{adapted})^2 = \beta_m \widehat{\sigma}_{md}^2 + (1 - \beta_m)(\sigma_{md}^2 + \mu_{md}^2) - (\mu_{md}^{adapted})^2$$

where σ_{md}^2 is the variance of dth dimension of mixture m, $(1 \le d \le D)$, $\beta_m = \frac{\eta_m}{\eta_m + \rho}, (1 \le m \le M)$ and γ is the scale factor that makes $\sum_{m=1}^{M} w_m = 1$
8: **return** $[\, w_m^{adapted} \ \bar{\mu}_m^{adapted} \ \boldsymbol{\Sigma}_m^{adapted} \,]^t, (1 \le m \le M)$

[16] to evaluate our system. The proxy server used here is a Squid proxy [17]. There are about 4000 users accessing the Internet through the proxy server. Each log entry gives the HTTP request placed by one of the 4000 users sitting behind the proxy server. We have captured the logs for 9 months from May 2010 to January 2011. The topology of the network where we captured the log is shown in Figure 3. This topology can be compared to the scenario shown in Figure 1b. Though we have used Squid proxy logs to evaluate our model, this work can be generalized to any kind of trail capturing mechanism – like tcpdump [18] or even a customized trail capturing mechanism that an ISP can decide.

We have manually labeled the data as *Social networking*, *Webmail*, *Video Streaming*. Data of all the three services – namely *Social networking*, *Webmail*, *Video Streaming* together is used to build the UBM. To evaluate the model of a particular service, we have given the data of other two services as the *anomalous* data. Given the feature vector, we calculate its *score* using Equation 2. The results shown have been validated using *10-fold cross validation* to verify that the results are not biased towards a specific training dataset.

A. Proxy log format

The complete Squid proxy log format can be found here [17]. Main fields of the proxy log are mentioned below.

- Time Stamp "Unix time" (seconds since Jan 1, 1970)
- Elapsed Time in milliseconds, This is the time between the `accept()` and `close()` of the client socket
- Client IP address
- Action/Code, describes how the request was treated locally (hit, miss, etc), code : http reply code
- Size. For TCP requests, the amount of data written to the client. For UDP requests, the size of the request. (in bytes)
- Method, the HTTP request method
- URI, the requested URI
- Content, Content-type of the Object (from the HTTP reply header)

B. The feature vector

Each feature vector has 43 feature values aggregated over a time period τ. We have used $\tau = 15, 30, 60$ minutes. Few of the features include Aggregate of Elapsed Time, Number of TCP HITs, Number of TCP MISSes, Number of HTTP Status Codes 200, 400, 401 etc., Aggregate of the total size of data written over a time period τ.

C. Experimental results

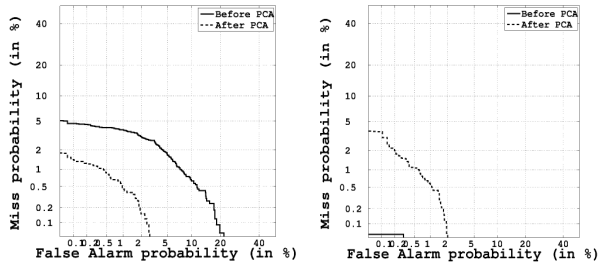
			Social Networking		Webmail		Video Streaming	
			Only Means adapted	All parameters adapted	Only Means adapted	All parameters adapted	Only Means adapted	All parameters adapted
15 minutes	Mixtures	1	3.66%	6.24%	0.20%	2.11%	0.094%	1.57%
		2	13.24%	19.83%	1.46%	3.32%	1.65%	2.18%
		4	–	–	–	–	–	–
30 minutes	Mixtures	1	2.97%	6.12%	0.10%	1.57%	0.625%	1.55%
		2	3.44%	9.91%	0.068%	2.43%	0.29%	2.58%
		4	3.89%	3.87%	4.49%	3.73%	0.45%	2.02%
60 minutes	Mixtures	1	2.77%	5.73%	0.056%	1.62%	0.034%	1.84%
		2	2.77%	5.07%	0.056%	4.65%	0.034%	1.29%
		4	2.34%	2.92%	2.70%	3.79%	0.44%	2.49%

TABLE I: EER for the data before doing PCA

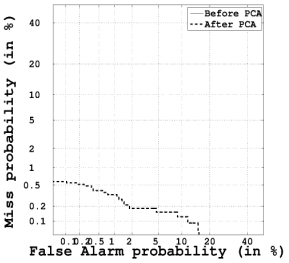
We have used Principal Component Analysis (PCA) [5] to find the dimensions that are orthogonal to one another. We project the data onto these dimensions to make all the dimensions of the data independent of one another. The results of the experiments done over data before doing PCA and after PCA are given in Tables I and II respectively. We have tested our model for time intervals τ =15, 30, 60 minutes. The dash in Table I for 15 minutes interval indicates that the model could not be built with 4 mixtures. We also have evaluated the model by adapting only the *mean* parameter while keeping weights and covariance matrices of all the mixtures same as that of UBM. This invlolves a slight modification to *step-7* of Algorithm-1 where we estimate only $\bar{\mu}_m^{adapted}$, $(1 \le m \le M)$.

			Social Networking		Webmail		Video Streaming	
			Only Means adapted	All parameters adapted	Only Means adapted	All parameters adapted	Only Means adapted	All parameters adapted
15 minutes	Mixtures	1	1.43%	10.12%	1.29%	8.92%	4.29%	3.37%
		2	4.72%	6.31%	2.45%	5.68%	5.32%	5.37%
		4	1.56%	5.82%	1.50%	6.36%	2.83%	11.92%
30 minutes	Mixtures	1	0.91%	9.36%	0.97%	8.82%	0.34%	6.07%
		2	4.98%	6.35%	2.64%	5.99%	1.63%	6.07%
		4	1.58%	4.58%	1.93%	4.87%	1.08%	14.40%
60 minutes	Mixtures	1	0.66%	6.93%	0.74%	10.09%	0.39%	2.60%
		2	5.52%	5.17%	3.51%	5.02%	1.21%	5.36%
		4	1.54%	4.95%	1.27%	4.33%	1.21%	13.60%

TABLE II: EER for the data after PCA



(a) Social Networking (b) Webmail



(c) Video Streaming

Fig. 4: DET plot of the services tunnelled through HTTP. The systems are built using *mean* only adaptation, 1 mixture and the data captured for an interval $\tau = 60$ minutes.

From Tables I and II it is clear that the services like Social Networking, Webmail and Video streaming that are tunnelled through HTTP obey unique characteristics. It can also be observed from these tables that the system performs better when we adapt only the means. In almost all the cases the system performance is better when the models are built with only one mixture. This is because, this particular data is inherently unimodal in nature. Nevertheless we use GMM to model the traffic as it is observed that the various traffic obey multimodal distribution [1]. The DET plots of the three systems Social Networking, Webmail and Video Streaming traffic built using *mean* only adaptation, 1 mixture and for $\tau = 60$ minutes are shown in Figure 4.

V. Conclusion and Future work

In this paper, we have proposed a framework for detecting the anomalies in the network traffic. Results show that various services like Social Networking, Webmail and Video Streaming that are tunnelled through HTTP obey unique characteristics. The system performs better when we adapt only the means as compared to adapting all the parameters of GMM. Also the model works with less than 1% Misses and False alarms. Though we have used Squid proxy logs to evaluate our model, this work can be generalized to any kind of trail capturing mechanism – like tcpdump [18] or even a customized trail capturing mechanism that an ISP can decide.

In this work, we have not addressed the problem of determining the appropriate time interval τ at which the traffic characteristics should be collected for verifying the legitimacy of the traffic. As a future work we would like to address this problem. We have evaluated the proposed framework for a trail capturing mechanism like proxy-log. We can extend this framework to other such mechanisms namely tcpdump. Also to evaluate the model of a particular service, we have given the data of other two services as the *anomalous* data. We like to test the model for the actual attack data by collecting the trail of the simulated attacks and then evaluate the performance of the model by providing this attack data for verification. This also involves the mechanism to determine a proper threshold that enables us to classify the traffic pattern as normal or an anomaly by comparing the *log-likelihood* ratio or *score* of the traffic pattern with this threshold.

Acknowledgment

This work has been supported by the IU-ATC (India-UK Advanced Technology Centre) of Excellence in Next Generation Networks, Systems and Services.

References

[1] D. M. Divakaran, H. A. Murthy, and T. A. Gonsalves, "Traffic Modeling and Classification Using Packet Train Length and Packet Train Size," in 6^{th} *IEEE International Workshop on IP Operations and Management, IPOM*, pp. 1–12, Oct. 2006.

[2] M. A. Saifulla, H. A. Murthy, and T. A. Gonsalves, "Identifying patterns in internet traffic," pp. 859–865, 2002.

[3] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2. ed., 2001.

[6] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm.," *J. Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[7] J. luc Gauvain and C. hui Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291–298, 1994.

[8] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," in *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001*, pp. 5–8, 2001.

[9] L. Ertöz, E. Eilertson, A. Lazarevic, P. N. Tan, V. Kumar, J. Srivastava, and P. Dokas, *MINDS - Minnesota Intrusion Detection System*. MIT Press, 2004.

[10] M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *PROCEEDINGS OF THE 2000 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA*, pp. 93–104, ACM, 2000.

[11] A. V. Debra Anderson, Thane Frivold, "Next-generation intrusion detection expert system (nides) - a summary," Tech. Rep. SRI-CSL-95-07, SRI International, Menlo Park, CA 94025-3493, May 1995.

[12] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *In ACM SIGCOMM*, pp. 219–230, 2004.

[13] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of pca for traffic anomaly detection," in *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '07, (New York, NY, USA), pp. 109–120, ACM, 2007.

[14] D. Brauckhoff, K. Salamatian, and M. May, "Applying pca for traffic anomaly detection: Problems and solutions," in *INFOCOM 2009, IEEE*, pp. 2866 –2870, april 2009.

[15] N. Wu and J. Zhang, "Factor-analysis based anomaly detection and clustering," *Decision Support Systems*, vol. 42, no. 1, pp. 375 – 389, 2006.

[16] "Indian Institute of Technology Madras, Chennai." http://www.iitm.ac.in, Oct. 2011.

[17] "Squid proxy server." http://www.squid-cache.org, Oct. 2011.

[18] "Tcpdump, a packet capturing tool." http://www.tcpdump.org, Oct. 2011.