# Investigating Hidden Markov Models Capabilities in Anomaly Detection

Shrijit S. Joshi and Vir V. Phoha

Computer Science, Louisiana Tech University
Email: ssj005@latech.edu, phoha@latech.edu

## ABSTRACT

Hidden Markov Model (HMM) based applications are common in various areas, but the incorporation of HMM's for anomaly detection is still in its infancy. This paper aims at classifying the TCP network traffic as an attack or normal using HMM. The paper's main objective is to build an anomaly detection system, a predictive model capable of discriminating between normal and abnormal behavior of network traffic. In the training phase, special attention is given to the initialization and model selection issues, which makes the training phase particularly effective. For training HMM, 12.195% features out of the total features (5 features out of 41 features) present in the KDD Cup 1999 data set are used. Result of tests on the KDD Cup 1999 data set shows that the proposed system is able to classify network traffic in proportion to the number of features used for training HMM. We are extending our work on a larger data set for building an anomaly detection system.

## Keywords

Hidden Markov Models, Anomaly Detection system

## 1. INTRODUCTION

Intrusion detection systems (IDS) [11] have become popular tools for identifying anomalous and malicious activities in computer systems and networks [8]. Anomaly detection is a key element of intrusion detection and other detection systems in which perturbations from normal behavior suggest the presence of attacks, defects etc. [14]. Anomaly detection is performed by building a model that contains metrics derived from system operation and flagging any observation as intrusive that has a significant deviation from the model [1]. The paper aims at investigating the capabilities of Hidden Markov Models for building Anomaly Detection system. For a proof-of-concept, the proposed approach is tested using the KDD Cup 1999 data set in order to assess the robustness of the method; we have selected 5 features of the data set instead of selecting all of the features (41 features). We are extending the model to more features and larger datasets.

The structure of the paper is as follows: Section 2 gives the brief introduction to the concepts of Hidden Markov Model. Section 3 deals with the strategy we have employed for making an Anomaly Detection system that can classify network traffic as an attack or normal. It covers the mathematical modeling of the problem and describes the parameter estimation, and training procedure of HMM model and recognition phase of the system. Section 4 deals with the results which we have received after applying an HMM algorithm to develop an anomaly detection system. Some concluding remarks are given in section 5.

## 2. HIDDEN MARKOV MODEL

Hidden Markov Model is an instance of a more general class of models designed by stochastic finite state networks [12]. It generates an internal sequence of symbols and a sequence of external symbols, using probabilistic rules [13] . An HMM is characterized by $\lambda$ = {A, B, $\pi$ }. HMMs are not exhaustively treated in this paper; we refer the reader to read [12] for more details. Various elements of HMM are briefly described here as follows: (1) 'N' represents the number of states in the model, (2) Individual states are denoted as S = {$S_1$, $S_2$... $S_N$}, (3) State at time 't' is denoted as '$q_t$', (4) 'M' represents the number of distinct observation symbols per state; these observation symbols correspond to the physical output of the system being modeled, (5) Individual symbols are denoted as V = {$V_1$, $V_2$... $V_M$}, (6) 'A' represents the state transition probability distribution where A = {$a_{ij}$}, (7) 'B' represents the observation symbol probability distribution where B = {$b_{jk}$}, (8) ' $\pi$ ' represents the initial state probability distribution where $\pi$ = { $\pi_i$}, and (9) A random sequence O = $O_1$, $O_2$… $O_T$ represents the indirect observations of the underlying hidden sequence of states where 'T' represents the number of observations taken.

# 3. THE STRATEGY

In this section, the proposed strategy is explained in detail. This section starts with the description of the features present in the KDD Cup 1999 data set (the data set we have used to test our Anomaly Detection system) and then gives the detailed descriptions of the training and classification procedures, focusing on the initialization and model selection issues.

## 3.1 KDD Cup 1999 data set

For our experiment, we have used the KDD Cup 1999 intrusion detection data set prepared by Lee *et al*. [9]. The data set contains 41 features representing selected measurements of normal and intrusive TCP sessions. Each labeled TCP session is either normal or a member of one of the 22 attack classes in the dataset. The first 5 features are selected, namely (1) Src_Bytes, (2) Dst_Bytes, (3) Duration, (4) Is_Host_Login, and (5) Is_Guest_Login, based on the results of Gopi *et al*. [8] which are based on the Screen Test and Critical Eigenvalue test. The objective of selection of a subset out of all the features is to assess the robustness of our system.

## 3.2 Mathematical Modeling

While modeling the stated problem, we have used Urn and Ball model of [12] In Urn and Ball model, 'N' parameter of HMM represents the number of urns and 'M' parameter of HMM represents the number of colored balls per urn. Thus, in Urn and Ball model, each state corresponds to a specific urn, and the number of colored balls present in a particular urn corresponds to the number of observation symbols (value of 'M' parameter) per state. In our stated problem, we have selected 5 features of the KDD Cup 1999 data set (as stated in subsection 3.1) which correspond to urns (in Urn and Ball model), and thus parameter 'N' has value 5 in our model.

Each feature has some value for making each TCP session of the KDD Cup 1999 data set. These values correspond to the number of colored balls present per urn (i.e. per feature in our case). We have segregated the values of each feature in 6 sets (or otherwise it would become an infinitely large number of observation symbols per state), where each set corresponds to the particular color of the ball; which in turn formed the number of distinct observation symbols per feature or per state, and thus parameter 'M' has value 6 in our model.

Table 1 shows the actual values of the selected features of one of the TCP sessions of the KDD Cup 1999 data set which is a normal means; it doesn't contain any sort of anomaly and it also shows the discrete observation symbol of the corresponding values of a TCP session.

**Table 1. Actual Values and Discrete Observation Symbol values of the Features of one of the TCP sessions of KDD Cup 1999 data set**

| Feature No. | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Values from a TCP session | 22 | 181 | 5450 | 0 | 0 |
| Observation Symbol Value | 1 | 1 | 3 | 1 | 1 |

We have received values for observation symbol after segregating the values of each feature in 6 sets.

For instance, here we have given the range of values of each Observation Symbol number of Duration feature: (1) Observation Symbol 1 is assigned to the value which is less than 1700, (2) Observation Symbol 2 is assigned to the value which is in between 1701 and 3500, (3) Observation Symbol 3 is assigned to the value which is in between 3501 and 6000, (4) Observation Symbol 4 is assigned to the value which is in between 6001 and 9500, (5) Observation Symbol 5 is assigned to the value which is in between 9501 and 165000, and (6) Observation Symbol 6 is assigned to the value which is more than 16500.

The states in our model are interconnected in such a way that any state can be reached from any state. In the Urn and Ball model, it is possible to find which ball has been taken from which urn, but in our case the combined effect of values of all the features results in a TCP session that is normal or contains some anomaly.

As values of all the features constitute TCP session and thus all these feature values make TCP session a normal or an attacked one. In our model, we have taken these values as observation sequence and classified them as their respective observation symbol number according to the guideline as explained above. This observation sequence represents the values that make the trend of normal or anomaly dependent upon the type of TCP session. The classification procedure is explained below by the following example.

For example, let us suppose in Urn and Ball model there are 5 urns each containing a red, green, blue, and white ball. Table 2 lists the sample observation sequence after picking one ball from each urn. A character R, G, B, and W shown in Table 1 represents the Red, Green, Blue, and White ball respectively.

**Table 2. Sample Observation sequence created for Urn and Ball model which is analogous to our model**

| Urn | 1 | 2 | 3 | 4 | 5 | Type |
|---|---|---|---|---|---|---|
| $O_1$ | R | G | B | R | G | $P_1$ |
| $O_2$ | R | G | B | G | W | $P_2$ |
| $O_3$ | R | G | B | R | W | $P_3$ |

The 'Type' field of Table 2 shows the type of observation sequence it is. Our HMM model trains the observation sequence of $O_1$, $O_2$, and $O_3$ and stores it as a trend followed by Type $P_1$, $P_2$, and $P_3$ respectively. Training procedure in our case (for anomaly detection system) is explained in section 3.3. While recognizing the given unknown observation sequence, if it is of Type $P_1$ then that sequence will follow the trend of $P_1$ much closer than Types $P_{2,}$ and $P_3$; as trained parameters of Type $P_1$ will recognize the unknown observation with higher probability rather than that of Types $P_{2,}$ and $P_3$. Thus, we could classify the given unknown sequence using trend analysis i.e. the one whose trained parameters recognizes the unknown observation sequence with higher probability. Classification procedure (recognition) in our case (for anomaly detection system) is explained in section 3.4.

The same analogy can be applied to our case, where each urn represents features and each ball represents the set of values of each observation symbol number. Therefore, if a TCP session is given of Type 'normal' then trained parameters of 'normal' TCP session recognize the unknown sequence with higher probability rather than that of the 'anomaly' typed TCP session.

The complete parameter set of the model can be described as $\lambda$ = {A, B, $\pi$ }. Number of Hidden states of the model ('N') is 5 represented as $S_1$, $S_2$, $S_3$, $S_4$, and $S_5$ where each state represents the feature Duration, Src_Bytes, Dst_Bytes, Is_Host_Login, and Is_Guest_Login respectively as explained above. Number of Observable symbols ('M') is 6 for states $S_1$, $S_2$, and $S_3$ and for $S_4$, and $S_5$ its value is 2.

The initial state distribution $\pi = \{\pi_i\}$.

where $\pi_i = P[S_j]$,
$1 \leq j \leq 5$

The State transition probability distribution $A = \{a_{ij}\}$

where $a_{ij} = P[q_{t+1} = S_j \mid q_t = S_i]$,
$1 \leq j \leq 5$ and
$1 \leq i \leq 5$

The Observation symbol probability distribution in state j, $B = \{b_j(k)\}$

where $b_j(k) = P[V_k \text{ at } t \mid q_t = S_i]$,
$1 \leq j \leq 5$ and
$1 \leq k \leq 6$ if j = 1, 2, or 3 else $1 \leq k \leq 2$

The following figure, Figure 1, shows the Finite State Automata of the transition of states from one state to another. $S_1$, $S_2$, $S_3$, $S_4$, and $S_5$ as shown in figure 1 represent the states of our HMM model which corresponds to Duration, Src_Bytes, Dst_Bytes, Is_Host_login, and Is_Guest_login feature of the KDD Cup 1999 data set respectively. The connecting link in the figure indicates the state transition probability for each state.

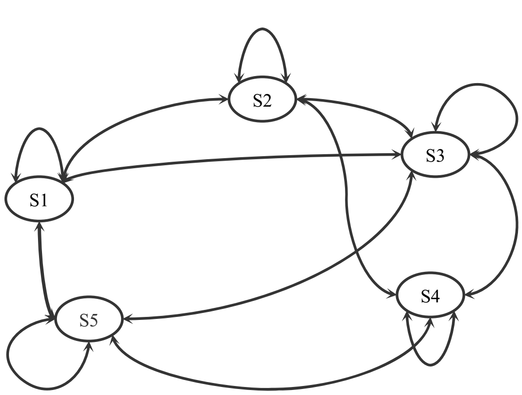

**Figure 1. Finite State Automata of the State Transition from one state to another state for 5 features of KDD Cup 1999 data set**

## 3.3 Parameter Estimation and Training

This sub-section of the paper deals with the parameter estimation and training of each TCP session of the KDD Cup 1999 data set. Section 3.3.1 exhaustively explains the initial estimation of HMM parameters and Section 3.3.2 covers the training phase of the algorithm.

### *3.3.1 Parameter Estimation*

Baum – Welch method [7] is used to estimate the HMM parameters. Baum – Welch method starts with an initial estimate, converges to the nearest local maximum of the likelihood function. Thus, the initialization process heavily affects the resulting estimate, as the likelihood function is highly multimodal.

Initial values of A, B, and $\pi$ are taken to be uniformly distributed. Table 3 and Table 4; indicate the corresponding initial values of $\pi$ and A parameter of our HMM model which correspond to the initial state probability distribution and state transition probability distribution respectively.

**Table 3. Initial State distribution (Parameter '$\pi$' of HMM) for one of the TCP sessions of KDD Cup 1999 data set**

| States | Initial State Distribution Value ($\pi_i$) |
|---|---|
| 1 | 0.000581 |
| 2 | 0.261902 |
| 3 | 0.089830 |
| 4 | 0.375828 |
| 5 | 0.271858 |

**Table 4. State transition probability distribution (Parameter 'A' of HMM) for one of the TCP sessions of KDD Cup 1999 data set**

| States | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0.1456 | 0.1062 | 0.2718 | 0.2496 | 0.2265 |
| 2 | 0.0679 | 0.3353 | 0.2774 | 0.2005 | 0.1186 |
| 3 | 0.0191 | 0.1165 | 0.4648 | 0.1878 | 0.2115 |
| 4 | 0.6308 | 0.2844 | 0.0760 | 0.0029 | 0.0056 |
| 5 | 0.1404 | 0.1976 | 0.2123 | 0.2237 | 0.2257 |

### *3.3.2 Training Session*

A practical, but fundamental issue to be solved when using an HMM is the determination of its structure, namely, the topology and the number of states i.e. finding a method to adjust the model parameters (A, B, $\pi$) to maximize the probability of the observation sequence given the model. According to [12] there is no optimal way of finding a method that analytically solves the problem. Though there are methods like Baum – Welch method i.e. EM (expectation – modification) method [7] or gradient techniques [10] which can choose $\lambda$ = (A, B, $\pi$) such that P (O | $\lambda$) is locally maximized. Some special purpose approaches (e.g. [15], [6]], [4], [5]) have been proposed for the model selection issue of HMM.

The training algorithm has the following steps (1) Initialization of Parameters of HMM, (2) Forward Procedure, (3) Backward Procedure, and (4) Re-estimation of Parameters of HMM. The first step of the training algorithm (i.e. Initialization of HMM Parameters) has been explained in subsection 3.3.1. Subsections 3.3.2.1, 3.3.2.2, and 3.3.2.3 describe the rest of the steps of the training session.

#### *3.3.2.1 Forward Procedure for Training*

After parameter estimation step, Forward Procedure [2] is applied for training HMM. The forward variable:

$$\alpha_t(i) = P(O_1, O_2, O_3, O_4, O_5,\ q_t = S_i \mid \lambda) \ \ldots\ldots\ldots\ldots \ (1)$$

The forward variable '$\alpha$' indicates the probability of the partial observation sequence, $O_1$, $O_2$, $O_3$, $O_4$, and $O_5$, and the state $S_i$ at time t, given the model $\lambda$.

Observation sequences $O_1$, $O_2$, $O_3$, $O_4$, and $O_5$ represent the discrete observation symbol number of the states $S_1$, $S_2$, $S_3$, $S_4$, and $S_5$ respectively. Thus, in our case values of $O_1$, $O_2$, $O_3$ ranges from 1 to 6 and for $O_4$ and $O_5$ it is either 1 or 2.

Steps involved in the Forward Procedure are described using equations (2), (3), and (4):

- Initialization of the forward variable value

$$\alpha_t(i) = \pi_i \times b_i(O_1) \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (2)$$

where $1 \leq i \leq 5$

- Induction step of the Forward Procedure

$$\alpha_{(t+1)}(j) = \left[\sum_{i=1}^{5} \alpha_t(i) \times a_{ij}\right] \times b_j(O_{t+1}) \ ..\ (3)$$

where $1 \leq t \leq T-1$ and $1 \leq j \leq 5$

- Termination step of the Forward Procedure

$$P(O \mid \lambda) = \sum_{i=1}^{5} \alpha_t(i) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (4)$$

Thus, P (O | $\lambda$) is the sum of all the $\alpha_t(i)$ values.

#### *3.3.2.2 Backward Procedure for Training*

After Forward Procedure, Backward Procedure [3] is applied for training. Backward variable $\beta_t(i)$ is the probability of the partial observation sequence from (t+1) to the end, given state $S_i$ at time t and the model $\lambda$. Steps involved in Backward Procedure are described using equations (5) and (6).

- Initialization of the Backward Variable

$$\beta_t(i) = 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

where $1 \leq i \leq 5$

- Induction step of the Backward Procedure

$$\beta_t(i) = \sum_{j=1}^{5} a_{ij} \times b_j(O_{t+1}) \times \beta_{t+1}(j) \quad \ldots\ldots\ldots\ldots (6)$$

where $t = T-1, T-2, T-3 \ldots 1$
$1 \leq i \leq 5$

#### *3.3.2.3 Calculation of $\gamma$ and $\varepsilon$ values*

For training, one possible optimality criterion could be to choose the state $q_t$, which is individually mostly likely. This optimality criterion maximizes the expected number of correct individual states. For implementing this solution we have taken variable $\gamma$. For updating and re-estimating the HMM parameters once they are initialized, we have defined the variable $\varepsilon$ (i, j) i.e. the probability of being in state $S_i$ at time t, and state $S_j$ at time (t + 1), given the model and the observation sequence. The value of this variable can be described using the forward and backward variables shown in equation (8).

$$\gamma_t(i) = [\alpha_t(i) \times \beta_t(i)] \,/\, \left(\sum_{i=1}^{5} \alpha_t(i) \times \beta_t(i)\right) ..(7)$$

$$\varepsilon_t(i,j) = \left(\alpha_t(i) \times a_{ij} \times b_j(O_{t+1}) \times \beta_{t+1}(j)\right) \,/\, \left(\sum_{i=1}^{5} \sum_{j=1}^{5} \alpha_t(i) \times a_{ij} \times b_j(O_{t+1}) \times \beta_{t+1}(j)\right) \ \ldots\ldots\ldots \ (8)$$

#### *3.3.2.4 Re - Estimation of HMM parameters*

This is the most important step of Training algorithm.

Steps involved in this phase are described as follows:

- Re – estimating initial state distribution values

  $\pi_i = \gamma_1(i)$ …………………………. (9)

  where $1 \leq i \leq 5$

- Re – estimating state transition probability distribution

  $a_{ij} = \sum_{t=1}^{T-1} \varepsilon_t(i,j) \;/\; \sum_{t=1}^{T-1} \gamma_t(i)$ …………...(10)

- Re – estimating observation symbol probability distribution

$b_j(k) = [\text{ (s.t.) } O_t = V_k ] \sum_{t=1}^{5} \gamma_t(j) \;/\; \sum_{t=1}^{5} \gamma_t(j)$ ……(11)

The process of re-estimation of HMM parameters continue till the desired limiting point isn't reached. At the end of the training phase, we have one model for TCP Session of the dataset. Table 5 shows the trained values of the HMM parameters 'A' i.e. for state transition probability distribution for one of the TCP sessions of the KDD Cup 1999 data set which will be useful in the recognition phase of the algorithm.

**Table 5. Trained values of State transition probability distribution (Parameter 'A' of HMM) for one of the TCP sessions of KDD Cup 1999 data set**

| S= | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 5E-78 | 0.1249 | 0.34706 | 1.55716 | 0.52794 |
| 2 | 3E-12 | 0.0066 | 0.01015 | 0.98161 | 0.00158 |
| 3 | 1E-13 | 0.0951 | 0.57270 | 0.12032 | 0.21178 |
| 4 | 1 | 1.5080 | 1.43776 | 4.61048 | 2.62517 |
| 5 | 1E-20 | 5.4928 | 7.40352 | 0.99982 | 4.80135 |

## 3.4 Recognition Phase

For checking whether the particular network traffic is normal or it contains some sort of anomaly, we give our anomaly detection system discrete observation symbol values of the following features: (1) duration, (2) Src_Bytes, (3) Dst_Bytes, (4) Is_Host_Login, and (5) Is_Guest_login that correspond to the unknown observations sequence of $O_1$, $O_2$, $O_3$, $O_4$, and $O_5$. Given an unknown observation sequence O = $O_1$, $O_2$, $O_3$, $O_4$, $O_5$; the standard maximum likelihood principle computes the value of P (O | $\lambda_i$) (where i represents the number of models present in the trained dataset). The session is assigned to the model $\lambda_i$, whose model shows the highest likelihood.

Thus, i = arg max (P O| $\lambda_i$) ……………………. (12)

Given observation sequence represents the TCP session, and this recognition algorithm classifies the TCP session as normal or anomaly by assigning the nature (normal or anomaly) of the winning model $\lambda_i$. This recognition procedure has been partly explained in section 32.

## 4. RESULTS

We present the performance of Anomaly Detection system in terms of detection accuracy. The detection accuracy of Anomaly Detection system is the percentage of attack samples detected as attacks. The false positive rate of Anomaly Detection system accounts for the detection of normal sample as an attack. We analyzed the working of our Anomaly Detection system on the KDD Cup 1999 data set.

We took 5 features (1) Src_Bytes, (2) Dst_Bytes, (3) duration, (4) Is_Host_login, and (5) Is_Guest_login of the KDD Cup 1999 data set for analyzing the capability of our Anomaly Detection system. The best results was 79 % accurate i.e. given unknown sequence of TCP sessions this Anomaly Detection system could accurately verify that it is a normal or having anomaly with 79% accuracy. The remaining 21% is accounted for false positive rate (i.e. classifying anomaly as a normal TCP session) and false negative rate (i.e. classifying normal as an attack type TCP session). One of the reasons for false positive rate and false negative rate is the amount of features (12.195% of the total features) we have selected for training session of the algorithm (instead of the full 41 features). Thus, we think we can improve the efficiency of this algorithm by proper tuning of HMM parameters and also by taking into account the significant percentage of the TCP session features or by using a larger data set.

## 5. CONCLUSIONS AND DISCUSSIONS

This paper investigated the capabilities of Hidden Markov Model in Anomaly Detection System. As described above, one HMM has been trained for each TCP session of the KDD Cup 1999 data set. While training the model, special attention is given to the initialization of A, B, and $\pi$ parameters and model selection issue. Training is performed using standard Baum- Welch procedure. Network traffic to be tested is fed to all of the models then using standard maximum likelihood principle that traffic is rated as either normal or attack using the recognition phase of our algorithm. Tests on the KDD Cup 1999 data set indicate that HMM can be applied for Anomaly Detection wherein we have just taken only 12.195 % of the total features of the data set. Thus, this indicates that Hidden Markov Methodology, with particular care to the parameter estimation and the training phase, represents a powerful approach for creating Anomaly Detection System which can find whether the traffic is normal or containing some sort of anomaly at runtime that might solve the major concern of the Computer Security. We are extending our work on a more rigorous data set for building a highly reliable anomaly detection system.

## 6. ACKNOWLEDGEMENT

This project is supported in part by the Army Research Office under Grant No. DAAD 19-01-0646.

## 7. REFERENCES

[1] Balasubramaniyan, J.S., Garcia-Fernandez, J.O., Isacoff, D., Spafford, E. and Zamboni, D., An Architecture for Intrusion Detection using Autonomous Agents. In *Proceeding of the Fourteenth Annual Computer Security Applications Conference*, (Radisson Resort Scottsdale, Phonenix, Arizona, 1998).

[2] Baum, L.E. and Egon, J.A., An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to model for a ecology. In *Bull. Amer. Meteorol. Soc.*, (1967), 360 - 363.

[3] Baum, L.E. and Sell, G.R., Growth functions for transformations on manifolds. In *Pac. J. Math.*, (1968), 211 - 227.

[4] Bicego, M., Dovier, A. and Murino, V., Designing the Minimal Structure of Hidden Markov Models by Bisimulation. In *Energy Minimization methods in Computer Vision and Pattern Recognition*, (2001), 75 - 90.

[5] Bicego, M., Murino, V. and Figueiredo, M. A Sequential Pruning Strategy for the Selection of the Number of States in Hidden Markov Models *Pattern Recognition letters*, 2003, 1395 - 1407.

[6] Brand, M., An Entropic Estimator for Structure Discovery. In *Advances in Neural Information Processing Systems*, (1999).

[7] Dempster, A.P., Laird, N.M. and Robin, D.B., Maximum likelihood from incomplete data via the EM algorithm. In *J. Roy. Stat. Soc.*, (1977), 1 - 38.

[8] Kuchimanchi, G., Phoha, V.V., Balagani, K.S. and Gaddam, S.R., Dimension Reduction using Feature Extraction Methods for Real-time Misuse Detection Systems. In W*orkshop on Information Assurance, United States Military Academy, West Point, NY*, (2004).

[9] Lee, W. and Stolfo, S.J., A Framework for Constructing features and models for Intrusion Detection Systems. In *ACM Transactions on Information and System Security*, (2000), 227 - 261.

[10] Levinson, S.E., Rabiner, L.R. and Sondhi, M.M., An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. In *Bell System Tech. J.*, (1983), 1035 - 1047.

[11] Phoha, V.V. *The Springer Dictionary of Internet Security*, Springer, Verlag, New York, June 2002.

[12] Rabiner, L., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, (February 1989).

[13] Rena Hixon and Gruenbacher, D.M., Markov Chains in Network Intrusion Detection. In *Workshop on Information Assurance, United States Military Academy, West Point, NY*, (2004).

[14] Roy A. Maxion and Tan, K.M.C., Benchmarking Anomaly-Based Detection Systems. In *1st International Conference on Dependable Systems &Networks: New York, New York, USA*, (2000).

[15] Stolcke, A. and Omohundro, S., Hidden Markov Model Induction by Bayesian Model Merging. In *Advances in Neural Information Processing Systems*, (1993), 11 - 18.