A Detection Approach of User Behaviors Based on HsMM*

XIE Yi¹ and YU Shunzheng²

Department of Electrical and Communication Engineering
Sun Yat-Sen University, Guangzhou 510275, China.
1 xieyicn@163.com
2 syu@zsu.edu.cn

Abstract: It is difficult for the existing anomaly detection methods to distinguish the burst of normal traffic from the anomalous traffic in a large-scale web site. This paper extends the current Hidden semi-Markov Model (HsMM) that is used for single sequence of observations to the HsMM with multiple sequences of observations, and proposes a detection approach on user behaviors based on this extended HsMM. Two new on-line algorithms are proposed in this approach to solve the problems that we have incomplete training data and need an algorithm for on-line updating of the model parameters. By conducting an experiment with a real traffic data, this approach shows that it is not only suitable for describing the characteristics of the access behaviors of users, but also effective in measuring the degree of normality of the user behaviors.

Key Words: HsMM, User behaviors, Anomaly detection

1. INTRODUCTION

With the development of information networks, the number of network attacks is increasing. The attack methods become more and more complicated. People have to pay more attention to the network security and develop intrusion detection systems. The characteristics of a large-scale web site are different from those of a general web site, and its traffic volume is very huge and quite bursty. These characteristics cause the large-scale web sites to be much easier to be attacked by the Flooding Attack of Distribute Denial of Service (DDoS) than the general web sites [1]. It is difficult for the general anomaly intrusion detection methods [2] to distinguish the burst and huge volume of the normal traffic from the flooding traffic of DDoS in a large-scale web site. Because of this reason, the fault alarm rate is very high when the general anomaly detection methods are used for such web site. Therefore, the security of the large-scale web sites becomes a new challenge for the network security technologies.

*This work was supported by Natural Science Foundation of China under grant No. 90304011 and partially by Natural Science Foundation of Guangdong Province under grant No.04009747.
There exist many anomaly detection methods in the literature, which are based on Data Mining [3], Neural Network [4], Markov chains [5] and etc. However, we cannot find many methods in the literature that emphasize the large-scale web sites [6-7]. This paper uses the Hidden semi-Markov Model (HsMM) [8-9] to describe the user behaviors that may change with the time. Hidden Markov Model (HMM), which is widely applied in Speech Recognition, Character Recognition and DNA sequences clustering [8], is not widely used in the application of network security [10-12]. Some researches [13] have presented that network traffic has the property of second order self-similarity and long-range dependence. Yu et al. [14] prove that the HsMM is better than the HMM in describing the unstable distribution and can describe the second order self-similarity and long-range dependence of network traffic which may change with the time. Because of these advantages, the HsMM can be used to detect the anomalous user behaviors.

The rest of the paper is organized as follows. In Section 2, we introduce a parameter re-estimation algorithm of HsMM for multiple observation sequences. We then, in Section 3, propose two new on-line detection algorithms on user behaviors. In Section 4, we conduct an experiment using three sets of real traffic data to validate our detection algorithms. Finally, we conclude our work.

2. A PARAMETER RE-ESTIMATION ALGORITHM OF HSMM

The main difference between HsMM and HMM is that the state duration is not a constant or exponentially distributed. Ferguson [15] is the first to investigate estimation algorithms for the HsMM. However, Ferguson’s algorithm is computationally too expensive to be of practical use in many applications. Yu et al [9] proposed a new forward-backward algorithm that reduces the computational complexity from $O((MD^2+M^2)T)$ to $O((MD+M^2)T)$, where $M$ is the number of states; $D$ the maximum possible interval between state transitions; and $T$ the period of the observations used to estimate the model parameters. This new algorithm improves the computational efficiency of HsMM and promotes its applications. The algorithm is briefly reviewed as follows.

The parameters of HsMM are denoted as: $\lambda = \{a_{mn}, \{\pi_m\}, \{b_m(v_k)\}, \{p_m(d)\}\}$, where $a_{mn}$ is the state transition probabilities; $\pi_m$ the initial state probabilities; $b_m(v_k)$ the observation element probabilities; $p_m(d)$ the probabilities of states’ duration; $o_t$ the observation sequence; $V = \{v_1, v_2, \ldots, v_k\}$ the set of observation elements; $S = \{1, 2, \ldots, M\}$ the set of states; and $\{1, 2, \ldots, D\}$ the set of state duration. Two forward-backward variables and other three variables denoted $\alpha_t(m,d)=\Pr(o_1^t, q_t, \tau_t) = (s_m,d)$, $\beta_t(m,d)$, $\zeta_t(m,n)$, $\eta_t(m,d)$ and $\gamma_t(m)$, are defined as follows:

\[
\alpha_t(m,d) = \Pr(o_1^t, q_t, \tau_t) = (s_m,d) = \alpha_{t-1}(m,d+1)b_m(o_t) + \sum_{n,s} \alpha_{t-1}(n,l)a_{mn}b_m(o_t)p_m(d),
\]

\[
\beta_t(m,d) = \Pr(o_1^T, q_t, \tau_t) = (s_m,d)
\]
\[
\xi_t(m, n) = \Pr[\alpha^T_1, q_{t-1} = s_m, q_t = s_n] = \alpha_{t-1}(m, 1)a_{mn}b_\phi(o_t) \left\{ \sum_{d \in \mathcal{D}} p_n(d)\beta_n(n, d) \right\},
\]

\[
\eta_t(m, d) = \Pr[\alpha^T_1, q_{t-1} \neq s_m, q_t = s_m, \tau_t = d] = \sum_{n \in \mathcal{S}} \alpha_{t-1}(n, 1)a_{mn}b_m(o_t)p_m(d)\beta_m(m, d),
\]

\[
\gamma_t(m) = \Pr[\alpha^T_1, q_t = s_m] = \gamma_{t+1}(m) + \sum_{n \in \mathcal{S}} (\xi_{t+1}(m, n) - \xi_{t+1}(n, m)),
\]

Where \( m, n \in \mathcal{S}, d \in \{1, 2, \ldots, D\} \).

Therefore, we can obtain the following estimation formulas:

\[
\hat{q}_t = \arg\max_{1 \leq m \leq M} \Pr[q_t = s_m | \alpha^T_1] = \arg\max \gamma_t(m), \quad t = T, T-1, \ldots, 1
\]

\[
p_\lambda(o^T_1) = \sum_{m, d} \alpha_t(m, d), \quad m \in \mathcal{S}, d \in \{1, 2, \ldots, D\}
\]

\[
\hat{\pi}_i = \frac{\gamma_i(i)}{\sum_{i \in \mathcal{S}} \gamma_i(i)}
\]

\[
\hat{a}_{ij} = \sum_i \xi(i, j) / \sum_i \sum_j \xi(i, j), \quad i, j \in \mathcal{S}
\]

\[
\hat{p}_m(d) = \sum_i \eta_i(m, d) / \sum_i \sum_d \eta_i(m, d), \quad m \in \mathcal{S}, d \in \{1, 2, \ldots, D\}
\]

\[
\hat{b}_m(k) = \sum_i \gamma_i(m) \delta(o_i - v_k) / \sum_i \sum_k \gamma_i(m) \delta(o_i - v_k), \quad m \in \mathcal{S}, v_k \in \mathcal{V}
\]

This algorithm is based on single observation sequence. Our anomaly detection method will have multiple observation sequences on multiple user behaviors. Therefore, we need to extend the re-estimation algorithm of the HsMM for multiple observation sequences. Using the frequency [8], we can derive the new formulas from the above results easily. Assuming there are \( L \) observation sequences with different lengths \( T_l(l=1, 2, \ldots, L) \), we obtain the extended re-estimation formulas:
\[
\hat{\pi}_i = \frac{\sum_{l=1}^{L} \frac{1}{P(o_i^{(l)} \mid \lambda)} \cdot \gamma_i^{(l)}(i)}{\sum_{l=1}^{L} \frac{1}{P(o_i^{(l)} \mid \lambda)} \cdot \sum_{i} \gamma_i^{(l)}(i)},
\]

(12)

\[
\hat{a}_{ij} = \frac{\sum_{l=1}^{L} \frac{1}{P(o_i^{(l)} \mid \lambda)} \cdot \sum_{j} \xi_{ij}^{(l)}(i,j)}{\sum_{l=1}^{L} \frac{1}{P(o_i^{(l)} \mid \lambda)} \cdot \sum_{j} \sum_{i} \xi_{ij}^{(l)}(i,j)},
\]

(13)

\[
\hat{p}_m(d) = \frac{\sum_{l=1}^{L} \frac{1}{P(o_i^{(l)} \mid \lambda)} \cdot \sum_{d} \eta_d^{(l)}(m,d)}{\sum_{l=1}^{L} \frac{1}{P(o_i^{(l)} \mid \lambda)} \cdot \sum_{d} \sum_{i} \eta_d^{(l)}(m,d)},
\]

(14)

\[
\hat{b}_m(k) = \frac{\sum_{l=1}^{L} \frac{1}{P(o_i^{(l)} \mid \lambda)} \cdot \sum_{k} \gamma_k^{(l)}(m) \delta(o_i^{(l)} - v_k)}{\sum_{l=1}^{L} \frac{1}{P(o_i^{(l)} \mid \lambda)} \cdot \sum_{k} \sum_{i} \gamma_k^{(l)}(m) \delta(o_i^{(l)} - v_k)},
\]

(15)

\[
P(O \mid \lambda) = \prod_{l=1}^{L} P(o_i^{(l)} \mid \lambda) = \prod_{l=1}^{L} \sum_{m,d} \alpha_{ij}^{(l)}(m,d),
\]

(16)

Where \(i, j, m \in S, d \in \{1,2,\ldots, D\} \) and \(v_k \in V\). From the above re-estimation formulas, we have the HsMM with multiple observation sequences.

3. ANOMALY DETECTION ALGORITHM

Usually, we track a user access behavior to a server by observing the sequence of objects the user requests. “Think time” is another observable presenting the time spent by the user in browsing a special object. We use a two-dimensional random vector \(\bar{x} = (objectID, interval)\) to describe the user behavior in this paper, where \(objectID\) is the index of an object in the server, e.g., an HTML page or an image file, and \(interval\) is the “think time” of the user in browsing the object. Therefore, the user access behavior can be considered as a process \(\{\bar{x}_t : t = 1,2,\ldots,T\}\), where \(\bar{x}_t\) is the value of \(\bar{x}\) taken at time \(t\). For different users we can obtain different sample sequences of the stochastic processes from the access logs of the server.

We assume that the process \(\{\bar{x}_t\}\) is controlled by an underlying semi-Markov process. For an HsMM whose parameters are given, each state (called hidden state) of the HsMM can be used to describe a sequence of operations of the user. Transitions of the hidden states can be considered as the user’s browsing behavior from one web page to another following links between the pages. Therefore, likelihood of normal users’ access sequences computed by the given HsMM can be used to construct a distribution of the likelihoods. In considering that most of the normal users take the similar actions to access the server, i.e., with the similar likelihoods of the access process, we can define the normal degree of the user behaviors according to this distribution. Using this normal degree, we can judge a user who is normal or anomalous. As an application of this method, we propose an approach for anomaly detection as shown in Figure 1.
In this approach, a set of training data is used to construct an HsMM in the beginning. Then, we obtain the HsMM and the likelihood distribution of the training data set. We call this likelihood distribution Original Likelihood Distribution (OLD) in this paper. A set of preprocessed real data is then evaluated by the HsMM and the corresponding likelihood of every user is obtained. By comparing the likelihood of a user with the OLD, we can judge whether the user behavior embedded in the data set is normal or not. If the user behavior is normal, a switch will be set on, and then the real data will be saved into the training data module and sent to the single sequence HsMM module. Using the outputs of the single sequence HsMM, parameters of the main HsMM can be updated. If the user behavior is anomalous, the switch will be set off and other module (e.g. anomalous-process module) will be called.

Furthermore, two improved on-line algorithms are integrated in this approach to solve two basic problems to be solved for most statistical models. We introduce them as follows:

(1) Insufficient Training Data

Because a finite number of samples are used in training the model, the model has such a problem that an object ID that appears in the real data set may not appear in the training data set. Since the object ID is just a discrete symbol associated with pages, we can’t use the Interpolation or the Mixture Continuous Observation Densities [8] to solve this problem. This will result in that the HsMM algorithm cannot go through when a new object ID appears in the real data. Once the output probability, $b_m(v_k)$, of the object ID reaches zero it cannot update back to nonzero forever. In this paper, we use the linear-prediction [16] of Digital Signal Process (DSP) to solve this problem.

At first, we sort the elements of $V$, the subset of object IDs that are included in the training data set, by their frequency and denote it by $V = (v_1, v_2, ..., v_K)$, where $v_1$ is the symbol of the element with maximum frequency in the training data set; $v_2$ the second one and so on; $v_K$ the symbol of the element with minimum frequency in the training data set. All the elements which do not appear in the training data set are not included in $V$ and $V'$. Then, we can construct $b_m(v_k')$ from $b_m(v_k)$ . Using the linear-prediction, the output probability $b_m(v_k')$ of $v_K$ with minimum frequency can be approximated as follows:

$$b_m(v'_k) = \sum_{i=1}^{K} h_m(i) b_m(v_{K-i}) \quad 1 \leq m \leq M,$$  (17)
where $M$ is the number of the Markov states, and $h_m(i)$ are the coefficients that can be determined by Yule-Walker equations [16]. When a new observation element appears in the real data, we denote it as $v_{K+1}$, the output probability of which can be considered less than that of $v_K$. Therefore, we can use the linear-prediction to estimate the value of $b_m(v_{K+1})$ by

$$b_m(v_{K+1}) = -\sum_{i=1}^{M} h_m(i) h_m(v_{K+1-i}) \quad 1 \leq m \leq M,$$

(18)

After we obtain $b_m(v_{K+1})$ for each state $m$, we need to make the output probability matrix to be normalized (i.e., the sum of each row is 1). When all these are finished, we can use this temporarily updated HsMM to estimate the likelihood of real data that have the new observation element. In this algorithm, each state has $p$ coefficients, which determines the accuracy of the estimation. Because $V'$ has been sorted by the frequencies, $h_m(v_{K+1})$ of the new observation data is just related to the probabilities of several closest observation elements. Hence we only need to set $p$ to be a small number that can increase the efficiency of computation with a good estimation.

(2) On-line Algorithm of Updating Model Parameters

The above method can be used to estimate the probability of new observation elements which never appear in the training data set. However, the training data are finite and static, and user behaviors are changing with time. If the parameters of HsMM are also static without update, the model will become invalid gradually. For instance, if a new observation element $v_{K+1}$ appears with high frequency after some time (i.e. the majority of user behaviors are changing), and its accumulative total of frequency is more than that of $v_K$, the system may still consider $v_{K+1}$ to be the element with minimum frequency. Apparently, the results (likelihoods) computed by this HsMM will deviate from the normal probability distribution built by the finite set of training data and become worse and worse. At last, we maybe get a wrong judgment for a normal user with new access behavior.

In order to solve this problem, the HsMM must be able to update its parameters with time. One solution could be like this: when a real sequence is input to the HsMM, the model uses both the original training data and this real sequence to re-estimate the parameters. Because of its large amount of computations, this solution is unpractical for on-line use. In this paper, we improve the re-estimation algorithm of HsMM for updating its parameters based on [17]. The main recursion process becomes as follows:

Let $\lambda^L = (\{a_{mn}^L\}, \{\pi_m^L\}, \{b_m^L(v_k)\}, \{p_m^L(d)\})$ be the parameters of HsMM with $L$ training sequences and $\lambda^{(i)} = (\{a_{mn}^{(i)}\}, \{\pi_m^{(i)}\}, \{b_m^{(i)}(v_k)\}, \{p_m^{(i)}(d)\})$ be the parameters of HsMM which is trained by a single observation sequence $l$. Using the estimation algorithm of HsMM for multiple observation sequence described in section 2, we have:

$$a_{ij}^{L+1} = \frac{\sum_{j=1}^{L+1} \sum_{i=1}^{L+1} p(q_{t-1} = s_i, q_t = s_j | o_t^{(i)} \lambda)}{\sum_{j=1}^{L+1} \sum_{i=1}^{L+1} \sum_{j} p(q_{t-1} = s_i, q_t = s_j | o_t^{(i)} \lambda)} = \frac{\sum_{i=1}^{L+1} \text{trans}(i, j, l)}{\sum_{i=1}^{L+1} \text{states}(i, l)}$$
\begin{equation}
\approx \left( \sum_{i=1}^{L} \text{states}(i,l) \right) \left/ \sum_{l=1}^{l+1} \text{states}(i,l) \right) a_{ij}^L + \left( \sum_{i=1}^{L+1} \text{states}(i,l) \right) a_{ij}^{L+1},
\end{equation}

where \( \text{trans}(i,j,l) \) is the frequency of transitions from state \( i \) to state \( j \) in sequence \( l \); and \( \text{states}(i,l) \) is frequency of state \( i \) which appears in sequence \( l \).

From the above formula, we can conclude that the \( a_{ij}^{L+1} \) can be estimated by \( a_{ij}^L \) and \( a_{ij}^{(L+1)} \). Using the similar method, we can obtain the other parameters as follows:

\begin{equation}
b_{L+1}^L(v_k) \approx \left( \sum_{i=1}^{L} \text{states}(i,l) \right) \left/ \sum_{l=1}^{l+1} \text{states}(i,l) \right) b_{L}^L(v_k) + \left( \sum_{i=1}^{L+1} \text{states}(i,l) \right) b_{L+1}^L(v_k),
\end{equation}

\begin{equation}
p_{L+1}^L(d) \approx \frac{\sum_{i=1}^{L} \text{turnin}(q_{i-1} \neq i, q_i = i, l)}{\sum_{l=1}^{L} \text{turnin}(q_{i-1} \neq i, q_i = i, l)} p_{L}^L(d) + \frac{\sum_{i=1}^{L} \text{turnin}(q_{i-1} \neq i, q_i = i, l)}{\sum_{l=1}^{L} \text{turnin}(q_{i-1} \neq i, q_i = i, l)} p_{L+1}^L(d),
\end{equation}

\begin{equation}
\pi_{L+1}^L \approx \left( L/(L+1) \right) \pi_{L}^L + \left( L/(L+1) \right) \pi_{L+1}^L,
\end{equation}

Where \( \text{turnin}(q_{i-1} \neq i, q_i = i, l) \) is the frequency of transition from other states to state \( i \) in sequence \( l \).

These equations show that \( \lambda^{L+1} \) can be estimated by \( \lambda^L \) and \( \lambda^{(L+1)} \) if all other coefficients are given. Moreover, since all coefficients in the above equations are only related to the frequency of appearance of state \( i \) (i.e., \( \sum_{l=1}^{L} \text{states}(i,l), i=1,2,...,M \)) and the frequency of transition from other states to state \( i \) (i.e., \( \sum_{l=1}^{L} \text{turnin}(q_{i-1} \neq i, q_i = i, l), i=1,2,...,M \)), we just need to determine \( \sum_{l=1}^{L} \text{states}(i,l) \) and \( \sum_{l=1}^{L} \text{turnin}(q_{i-1} \neq i, q_i = i, l) \), where \( i = 1,2,...,M \), and then we can obtain all the coefficients and the updated parameters of HsMM. The algorithm is very simple, efficient and practical.

Using the improved algorithm of parameter estimation of HsMM above, we can update the parameters of HsMM with low computational complexity. However, in the real network environment, if we update the parameters of HsMM without judgment, the model may be trained by the attacker and lose its ability of anomaly detection. Therefore, in our detection approach, we use the HsMM to judge the normal degree of the real sequence before it is used in model training. In this way, it can prevent the HsMM from being trained by hostile attacker.

### 4. EXPERIMENT RESULTS AND ANALYSIS

In this section, we use the real traffic of WorldCup98 [18] as the original experiment data to validate our detection algorithms.

(1) Data Preprocess
We select three groups of data from [18] and denote them as dataset1, dataset2 and dataset3. dataset1 and dataset2 include those users who may be individual users with low traffic volume; dataset3 includes the users who may be proxy servers with the largest traffic volumes. Each user’s sequence consists of two-dimensional random vector $\tilde{x} = (objectID, interval)$.

In order to reduce the computational complexity, we map the two-dimensional vector into one-dimensional symbol.

(2) Checking the Linear-Prediction for Insufficient Training Data

We use dataset1 to train the HsMM and obtain the parameters; and then we use this HsMM to compute the likelihood of dataset2. For those data which appear in dataset2 but do not appear in dataset1, we use two methods to estimate their likelihoods respectively: (1) using the Linear-Prediction method introduced in the previous section and let $p=2$; (2) the HsMM uses both the input sequence and dataset1 to estimate new parameters when the input sequence includes new observation elements, and then the new HsMM is used to compute the likelihood of this sequence. After the experiment, we obtain two groups of likelihoods of dataset2; we denote them as loglik1 and loglik2. In order to analyze the difference of likelihood distribution between those two methods, we compute the distribution of $(\loglik1 - \loglik2)$, as shown in Figure 2. We can see that most of the values are close to the zero and the differences of the results obtained by these two methods are small. Therefore, we can use the Linear-Prediction method, whose computational complexity is more efficient than the other one in solving the Insufficient Training Data problem.

(3) Detection of Normal Degree

We select the dataset1 to train the HsMM and obtain the OLD, as shown in Figure 3. The OLD is within $[-14.3586, -2.9089]$, the mean of OLD is $\mu = -5.9653$, and the variance is $\sigma^2 = 4.7006$.

In order to check the validity of our algorithm, we use the HsMM that is trained by dataset1 to compute the likelihoods of both dataset2 and dataset3, and compare the differences between them. In Figure 4, we can see that the differences of the likelihood distributions between dataset1 and dataset2 are small. This result shows that the users’ behaviors in dataset1 and dataset2 are very similar. But in Figure 5, there exist significant differences...
differences of likelihood distributions between \textit{dataset1} and \textit{dataset3}. It indicates that the access behaviors between the users with higher traffic volumes and those with lower traffic volumes are unlike; and our detection approach can distinguish these differences. From these results, we have another conclusion: if we want to use HsMM to detect the user behaviors, we must build different HsMM for users with different traffic volumes and cannot use only one HsMM to detect all user behaviors; otherwise, it is possible that our judgments would be confused by the detection results. For example, the behavior of a proxy server differs from those of individuals.

It is well-known that one of the characteristics of a DDoS attack is that the attackers send out millions of requests per-second. In this case, the attackers are the super-high traffic volume users whose behaviors differ from either ordinary individuals or proxy servers, and hence we can use our detection approach to recognize those attack behaviors easily.

5. CONCLUSIONS AND DISCUSSION

In this paper, we extended an existing parameter re-estimation algorithm of HsMM that is suitable for single observation sequence to that with multiple observation sequences. Based on the extended HsMM, we proposed a detection approach on anomalous user access behaviors. Using our two reformulated on-line algorithms, this approach solved two basic problems: Insufficient Training Data and On-line Updating of Model Parameters. Finally, we used the real access traffic data sets to validate our detection algorithms. From the results of the experiments, we can conclude that this approach can be applied effectively to detect the anomalous user access behaviors.

REFERENCES