Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 374260, 7 pages
http://dx.doi.org/10.1155/2014/374260

Research Article

The Application of Baum-Welch Algorithm in Multistep Attack

Yanxue Zhang,[1] Dongmei Zhao,[2] and Jinxing Liu[3]

[1] College of Mathematics and Information Science, Hebei Normal University, Shijiazhuang 050000, China
[2] College of Information Technology, Hebei Normal University, Shijiazhuang 050000, China
[3] The First Aeronautics College of PLAAF, Xinyang 464000, China

Correspondence should be addressed to Dongmei Zhao; zhaodongmei666@126.com

Received 8 April 2014; Accepted 6 May 2014; Published 28 May 2014

Academic Editor: Yuxin Mao

The biggest difficulty of hidden Markov model applied to multistep attack is the determination of observations. Now the research of the determination of observations is still lacking, and it shows a certain degree of subjectivity. In this regard, we integrate the attack intentions and hidden Markov model (HMM) and support a method to forecasting multistep attack based on hidden Markov model. Firstly, we train the existing hidden Markov model(s) by the Baum-Welch algorithm of HMM. Then we recognize the alert belonging to attack scenarios with the Forward algorithm of HMM. Finally, we forecast the next possible attack sequence with the Viterbi algorithm of HMM. The results of simulation experiments show that the hidden Markov models which have been trained are better than the untrained in recognition and prediction.

1. Introduction

Currently, the network security situation is increasingly sophisticated and the multistep network attack has become the mainstream of network attack. 2012 Chinese Internet network security reports released by the National Computer Network Emergency Response Technical Team Coordination Center of China (CNCERT/CC) show that the two typical multistep attacks: warms and distributed denial of service (DDOS) [1] account for 60% of overall network attacks. Multistep attack [2] means that the attacks apply multiple attack steps to attack the security holes of the target itself and achieve the devastating blow to the target. There are three features of attack steps of multistep attack. (1) In the multistep attack, there is a casual relationship between multiple attack steps. (2) The attack steps of multistep attack have the property of time sequence [3]. (3) The attack steps of multistep attack have the characteristics of uncertainty [4].

Multistep attack is one of the main forms of network attack behaviors, recognizing and predicting multistep attack that laid the foundation of active defense, which is still one of the hot spots nowadays. Literature (application of hidden Markov models to detect multistep network attacks) proposed a method to recognize multistep attack based on hidden Markov model.

Markov model literature (improving the quality of alerts and predicting intruder's next goal with hidden colored Petri-net) introduced the concept of attack "observation," but both stayed in the specific attack behaviors, which have some limitations. Current research on the approaches to forecast multistep attack behaviors mainly includes four types: (1) the approach to forecasting multistep attack based on the antecedents and consequences of the attack [5]. It applies the precursor subsequent relationship of the event, to forecast the attacker wants to implement attacks in the near future. Because of the complexity and the diversity of the attack behaviors, this approach is difficult to achieve. (2) The approach to forecasting multistep attack based on hierarchical colored Petri-nets (HCPN) applies the raw alerts by Petri-nets and considers that the attack intention is inferred by raw alerts [4]. But this approach focuses on the intrusion detection of multistep attack behaviors. (3) The approach to forecasting multistep attack based on Bayes game theory could forecast the probability that the attackers choose to

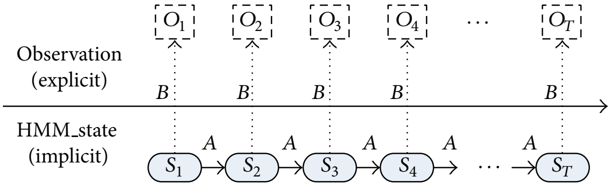


Figure 1: Model of recognizing and forecasting multistep attack based on hidden Markov model.

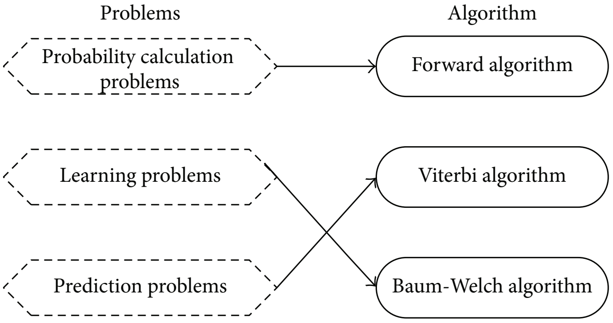


Figure 2: Correspondence between the problems and algorithms of hidden Markov model.

attack and the probability that the defenders choose to defend in the next stage rationally [6, 7]. However, in current study, only two-person game model is established, so this approach has some limitations. (4) The approach to forecasting multistep attack based on attack intention [3, 8] uses extended-directed graph to describe the logical relationship between attack behaviors and forecasts the next stage by logical relationship. The shortcoming of this approach is that it is difficult to determine the matching degree of the multistep attack. At the same time, there exists a certain degree of subjectivity in recognizing and forecasting multistep attack. In this regard, we integrate the attack intentions and hidden Markov model and propose a method to forecast multistep attack based on hidden Markov model. Firstly, we train the existing hidden Markov model(s) by the Baum-Welch algorithm of HMM. Then we recognize the alert belonging to attack scenarios with the Forward algorithm of HMM. Finally, we forecast the next possible attack sequence with the Viterbi algorithm of HMM. Simulation experiments results show that the hidden Markov models which have been trained are better than the untrained in recognition and prediction.

2. Hidden Markov Model

Hidden Markov model was first proposed by Baum and Petrie in 1966. It is a statistical model, which is used to describe a Markov process which contains a hidden parameter [9]. The research object of this model is a data sequence; each value of this data sequence is called an observation. Hidden Markov model assumes that there still exists another sequence which hides behind this data sequence; the other sequence consists of a series of states. Each observation occurs in a state, the state cannot be observed directly, and the features of the state can only be inferred from the observations.

A complete hidden Markov model (HMM) is usually represented by a triple $\lambda = (A, B, \pi)$, which includes the following five elements:

(1) a finite state, which is represented by the set S, where $S = \{s_1, s_2, \ldots, s_N\}$ and, at time t, the state is denoted by q_t;

(2) the set of observations, which is represented by the set O, where $O = \{o_1, o_2, \ldots, o_T\}$;

(3) the state transition matrix, which is represented by the matrix A, where $a_{ij} = p[q_{t+1} = s_j \mid q_t = s_j]$ and $1 \le i, j \le N$;

(4) the probability distribution of matrix A, which is represented by the matrix B, where $b_j(k) = p[o_k \mid q_t = s_j]$ and $1 \le j \le N$, $1 \le k \le T$;

(5) the set of initial state probability distribution of HMM, which is represented by the set π, where $\pi_i = p[q_1 = s_i]$ and $1 \le i \le N$.

The model of recognizing and forecasting multistep attack based on hidden Markov model is shown in Figure 1.

There are three problems which can be solved by hidden Markov model well.

(1) *Probability Calculation Problems*. Calculate the probability $p(O \mid \lambda)$ under a given hidden Markov model $\lambda = (A, B, \pi)$ and the observation sequence $O = \{o_1, o_2, \ldots, o_T\}$.

(2) *Learning Problems*. Estimate the parameters of $\lambda = (A, B, \pi)$ when the observation sequence $O = \{o_1, o_2, \ldots, o_T\}$ is known, to maximize the probability $p(O \mid \lambda)$.

(3) *Prediction Problems*. Calculate the state sequence $I = \{i_1, i_2, \ldots, i_T\}$ under the maximum probability, when the hidden Markov model $\lambda = (A, B, \pi)$ and observation sequence $O = \{o_1, o_2, \ldots, o_T\}$ are given.

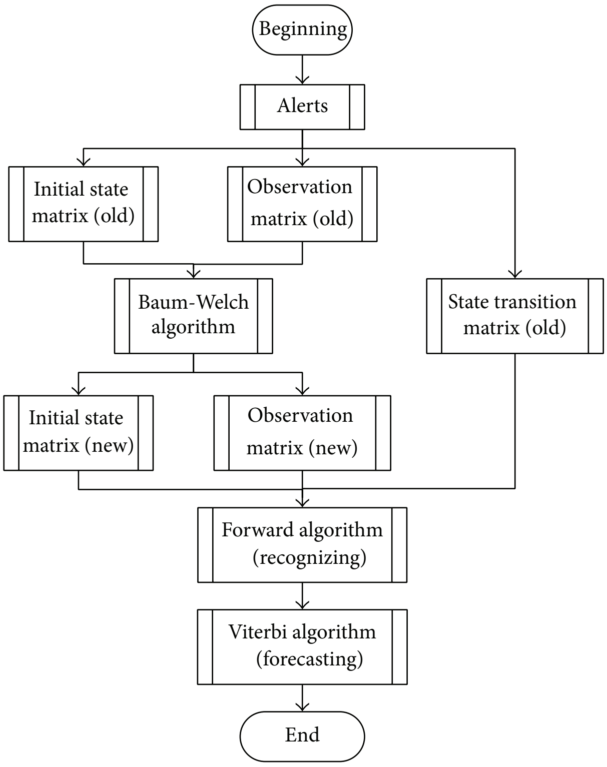

Beginning
Alerts
Initial state matrix (old)
Observation matrix (old)
Baum-Welch algorithm
State transition matrix (old)
Initial state matrix (new)
Observation matrix (new)
Forward algorithm (recognizing)
Viterbi algorithm (forecasting)
End

Figure 3: Flow chart of recognizing and forecasting multistep attack.

```
Input: alert sequence.
        $O = \{o_1, o_2, \ldots, o_T\}$;
Output: the parameters of hidden Markov model.
        $\lambda^{(n+1)} = (A^{(n+1)}, B^{(n+1)}, \pi^{(n+1)})$.
Step 1. Initialization.
        for $n = 0$, select $a_{ij}^{(0)}, b_j(k)^{(0)}, \pi_i^{(0)}$, we can obtain the initial model $\lambda^{(0)} = (A^{(0)}, B^{(0)}, \pi^{(0)})$.
Step 2. Iterative calculation.
for $n = 1, 2, \ldots$,
$a_{ij}^{(n+1)} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$;
$b_j(k)^{(n+1)} = \frac{\sum_{t=1, o_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$;
$\pi_i^{(n+1)} = \gamma_1(i)$.
where $\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{p(O \mid \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)\sum}$;
        $\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{p(O \mid \lambda)} = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}$.
Step 3. Termination. We can obtain the parameters of hidden Markov model.
$\lambda^{(n+1)} = (A^{(n+1)}, B^{(n+1)}, \pi^{(n+1)})$.
```

Algorithm 1

```
Forward_Algorithm $(\lambda, O)$:
Input: (1) alert sequence $O = \{alert_1, alert_2, \ldots, alert_T\}$;
        (2) hidden Markov model (HMM) $\lambda$.
Output: the probability $p(O \mid \lambda)$ generated by alert sequence $O = \{alert_1, alert_2, \ldots, alert_T\}$ of hidden Markov model.
Begin:
    (1) $\forall \operatorname{int} ent_i \in \lambda, 1 \le i \le N$.
        // $N$ is the number of attack intentions.
        calculate the probability of $alert_1$ generated by $\operatorname{int} ent_i$: $\alpha_1(i) = \pi_i b_i(alert_1)$
    (2) calculate the probability of alert sequence $\{alert_1, alert_2, \ldots, alert_T\}$ and $q_{t+1} = \operatorname{int} ent_j$.
        (a) at time $t$, calculate the probability of alert sequence $\{alert_1, alert_2, \ldots, alert_T\}$ and $q_t = \operatorname{int} ent_j$: $\alpha_t(j)$.
        (b) at time $t + 1$, calculate the probability of intent sequence $\{alert_1, alert_2, \ldots, alert_T\}$ generated by hidden Markov
        model (HMM): $\lambda$ and
        $q_t = \operatorname{int} ent_j$: $\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(alert_{t+1})$ where $1 \le t \le T-1; 1 \le j \le N$.
    (3) calculate the probability of the intent sequence $O = \{alert_1, alert_2, \ldots, alert_T\}$ generated by hidden Markov
        model (HMM): $\lambda$.
        $p(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$.
    (4) Return $p(O \mid \lambda)$.
End;
```

Algorithm 2

```
Viterbi_Algorithm$(\lambda, O)$:
Input: alert sequence $O = \{alert_1, alert_2, \ldots, alert_T\}$;
Output: (1) intent sequence: $Q = \{\operatorname{int} ent_1, \operatorname{int} ent_2, \ldots, \operatorname{int} ent_T\}$.
        (2) the completed intent sequence and the next likely intent.
Begin:
    for $i$ = 1 to HMM_$m$
    // HMM_$m$ is the number of hidden Markov model(s)
    {
        Prob = Forward_Algorithm(hmm_$i$, $O$);
        // calculate the probability of alert sequence generated by each hidden Markov
        // model(s)
    }
    Most_likely_multi-step_attack_intention = maximum(Prob);
    Q = Viterbi_Algorithm(hmm_$i'$, $O$);
    // $Q$ is the completed intent sequence
    // hmm_$i'$ is the maximum(Prob) of hmm_$i$
    $Q' = S - Q$    // the next likely intent
    // $S$ is the intent sequence of hmm_$i'$
End;
```

Algorithm 3

Correspondence between the problems and algorithms of hidden Markov model are shown in Figure 2.

Hidden Markov model is usually used to deal with the problems related to the time sequence and it has been widely used in speech recognition, signal processing, bioinformation, and other fields. Based on the characteristics of the attack steps of hidden Markov model and the problems that hidden Markov model can be solved, we apply the hidden Markov model to the field of recognizing and forecasting multistep attack. Firstly, the improved Baum-Welch algorithm is used to train the hidden Markov model λ, and we get a new hidden Markov model λ'. Then we recognize the alert belonging to attack scenarios with the Forward algorithm of hidden Markov model. Finally, we forecast the next possible attack sequence with the Viterbi algorithm of hidden Markov model.

3. The Approach to Recognizing and Forecasting Multistep Attack

The steps of the approach to recognizing and forecasting multistep attack are as follows.

Step 1. Obtain the initial state matrix (old), state transition matrix (old), and observation matrix (old) of HMM (λ).

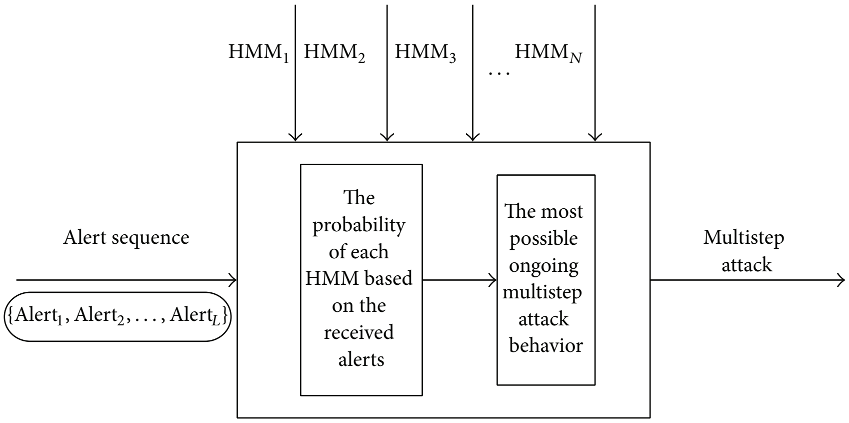

HMM$_1$
HMM$_2$
HMM$_3$
...
HMM$_N$
Alert sequence
{Alert$_1$, Alert$_2$, ..., Alert$_L$}
The probability of each HMM based on the received alerts
The most possible ongoing multistep attack behavior
Multistep attack

FIGURE 4: The structure of recognizing multistep attack with Forward algorithm.

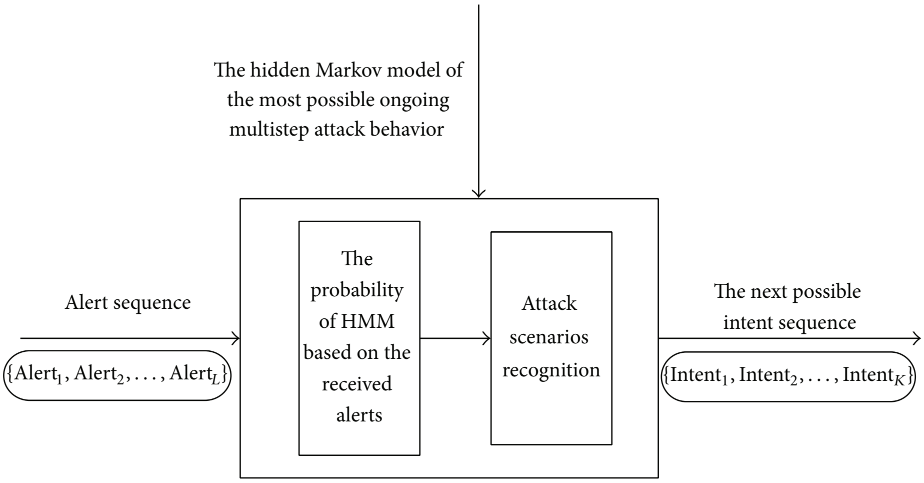

The hidden Markov model of the most possible ongoing multistep attack behavior
Alert sequence
{Alert$_1$, Alert$_2$, ..., Alert$_L$}
The probability of HMM based on the received alerts
Attack scenarios recognition
The next possible intent sequence
{Intent$_1$, Intent$_2$, ..., Intent$_K$}

FIGURE 5: Forecasting multistep attack with Viterbi algorithm.

TABLE 1: The initial state matrix of DDoS_HMM.

State$_1$	State$_2$	State$_3$	State$_4$	State$_5$
0.250	0.750	0.000	0.000	0.000

Step 2. Use the improved Baum-Welch algorithm to train the initial state matrix (old) and observation matrix (old), and we get an initial state matrix (new), observation matrix (new), and a new HMM (λ').

Step 3. Recognize the alert belonging to attack scenarios with the Forward algorithm.

Step 4. Forecast the next possible attack sequence with the Viterbi algorithm.

The flow chart is shown in Figure 3.

3.1. The Introduction of Baum-Welch Algorithm. If we want to apply the hidden Markov model to the multistep attack, the biggest problem is to determine the observations of HMM. A better parameter can improve the efficiency of calculation. Meanwhile, if the selection of observation is improper, this may result in a longer training time and even not complete the training. In this regard, we apply the Baum-Welch algorithm to train the given hidden Markov model. From the result of literature (accurate Baum-Welch algorithm free from overflow), we can learn that the most reliable algorithm to train the HMM is Baum-Welch algorithm. Baum-Welch algorithm can train the given hidden Markov model (λ) by an observation sequence and generate a new hidden Markov model (λ') for detection.

The steps of Baum-Welch algorithm are as in Algorithm 1.

TABLE 2: The state transition matrix of DDoS_HMM.

	State$_1$	State$_2$	State$_3$	State$_4$	State$_5$
State$_1$	0.000	1.000	0.000	0.000	0.000
State$_2$	0.000	0.177	0.823	0.000	0.000
State$_3$	0.000	0.228	0.688	0.028	0.056
State$_4$	0.000	0.000	0.000	0.750	0.250
State$_5$	0.000	0.000	0.000	0.000	0.000

Table 3: The observation matrix of DDoS_HMM.

	$A1$	$A2$	$A3$	$A4$	$A5$	$A6$	$A7$	$A8$	$A9$	$A10$	$A11$	$A12$	$A13$
$S1$	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$S2$	0.000	0.490	0.490	0.020	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$S3$	0.000	0.000	0.000	0.000	0.200	0.200	0.200	0.200	0.200	0.000	0.000	0.000	0.000
$S4$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
$S5$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.660	0.170	0.170

Table 4: The initial state matrix of DDoS_HMM$'$.

State$_1$	State$_2$	State$_3$	State$_4$	State$_5$
0.599	0.401	0.000	0.000	0.000

Table 5: The observation matrix of DDoS_HMM$'$.

	$A1$	$A2$	$A3$	$A4$	$A5$	$A6$	$A7$	$A8$	$A9$	$A10$	$A11$	$A12$	$A13$
$S1$	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$S2$	0.000	0.499	0.499	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$S3$	0.000	0.000	0.000	0.000	0.387	0.000	0.387	0.000	0.226	0.000	0.000	0.000	0.000
$S4$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
$S5$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.998	0.001	0.001

Table 6: DDoS_HMM.

STATE	ALERT
State$_1$	{Alert$_1$}
State$_2$	{Alert$_2$, Alert$_3$, Alert$_4$}
State$_3$	{Alert$_5$, Alert$_6$, Alert$_7$, Alert$_8$, Alert$_9$}
State$_4$	{Alert$_{10}$}
State$_5$	{Alert$_{11}$, Alert$_{12}$, Alert$_{13}$}

Table 7: FTP Bounce_HMM.

State	Alert
State$_1$	{Alert$_1{}'$, Alert$_2{}'$}
State$_2$	{Alert$_3{}'$, Alert$_4{}'$}
State$_3$	{Alert$_5{}'$, Alert$_6{}'$, Alert$_7{}'$}
State$_4$	{Alert$_8{}'$}
State$_5$	{Alert$_9{}'$, Alert$_{10}{}'$}

3.2. Forward Algorithm. The pseudocode of Forward algorithm is as in Algorithm 2.

Recognizing multistep attack is mainly based on the alert sequence. First, we calculate the probability of alert sequence generated by the given HMM(s). Then we decide that the attack which has the maximum is likely to be the ongoing attack. The structure of recognizing multistep attack with Forward algorithm is shown in Figure 4.

3.3. Viterbi Algorithm. The pseudocode of Viterbi algorithm is as in Algorithm 3.

Predicting the behavior of multistep attack is mainly to determine the intentions that the attackers have been completed and forecast the next possible attack intentions. The structure of forecasting multistep attack with Viterbi algorithm is shown in Figure 5.

4. The Simulation Experiment and Analysis

4.1. Baum-Welch Algorithm: Train the Given HMM(s). Based on the literature (approach to forecast multistep attack based on fuzzy hidden Markov model), we can obtain the initial state matrix, state transition matrix, and observation of DDoS_HMM, as is shown from Tables 1, 2, and 3.

The data set which is used in the simulation experiment is an attack scenario testing data set LLDOS1.0 (inside) provided by DARPA (Defense Advanced Research Projects Agency) in 2000. We extract two kinds of multistep attack from it; they are DDoS multistep attack and FTP Bounce multistep attack. While the calculation of the state transition matrix is completely the statistical calculations on data, we only train the initial state matrix and observation matrix of HMM. We can see that there are a large number of zeros in observation matrix clearly and the observation matrix is the sparse matrix. So we train the matrix(s) by block. We suppose that the number of observation sequences is S and the length of S is 32, where S multiplied by 32 equals the number of training data. And there is no corresponding sequence of state. In this regard, we can obtain the initial state matrix (new) and the observation matrix (new) of the DDoS_HMM$'$ (λ'), as is shown in Tables 4 and 5.

4.2. Forward Algorithm: Recognize the Alert Belonging to Attack Scenarios. The attack intentions and alerts of DDoS_HMM and FTP Bounce_HMM are shown in Tables 6 and 7, respectively.

When the alerts "Alert$_1$" and "Alert$_3$" were received, according to the Forward algorithm of hidden Markov model,

Table 8: The comparison of results.

	$p(\text{alerts} \mid \text{DDoS\_HMM})$	$p(\text{alerts} \mid \text{FTP Bounce\_HMM})$	$\frac{p(\text{alerts} \mid \text{DDoS\_HMM})}{p(\text{alerts} \mid \text{FTP Bounce\_HMM})}$
Before training	0.1225	0.0079	15.5
After training	0.2989	0.0036	83.0

we will obtain the probability based on DDoS_HMM$'$ and FTP Bounce_HMM$'$, respectively:

$$p(\text{alerts} \mid \text{DDoS\_HMM}) = 0.2989,$$
$$p(\text{alerts} \mid \text{FTP Bounce}) = 0.0036.$$

We can see from the above results, $p(\text{alerts} \mid \text{DDoS\_HMM}) > p(\text{alerts} \mid \text{FTP Bounce})$. That is to say, the ongoing multistep attack behavior is likely to be DDoS_HMM.

4.3. Viterbi Algorithm: Forecast the Next Possible Attack Sequence. When the alert sequence $\{\text{Alert}_1, \text{Alert}_3, \text{Alert}_7, \text{Alert}_8, \text{Alert}_{10}\}$ was received by the console, we can obtain the completed intent sequence $\{\text{State}_1, \text{State}_2, \text{State}_3, \text{State}_4\}$. That is to say, now completed intentions are the previous four attack intentions; the next intention will be state$_5$.

4.4. Comparison of Results. We compare the results between the untrained HMM(s) and the trained HMM(s) by Baum-Welch algorithm; the comparison of results are shown in Table 8.

5. Conclusion

The biggest difficulty of hidden Markov model applied in multistep attack is the determination of observations. Now the research of the determination of observations is still lacking, and it shows a certain degree of subjectivity. In this regard, we train the existing hidden Markov model(s) by the Baum-Welch algorithm of HMM based on several groups of observation sequence. And we can obtain a new hidden Markov model which is more objectively. Simulation experiments results show that the hidden Markov models which have been trained are better than the untrained in recognition and prediction.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped in improving the quality of this paper. This work was supported by the National Natural Science Foundation of China no. 60573036, Hebei Science Fund under Grant no. F2013205193, and Hebei Science Supported Planning Projects no. 12213514D.

References

[1] B. L. Xie, S. Y. Jiang, and Q. S. Zhang, “Application-ialer DDoS attack detection based on request keywords,” *Computer Science*, vol. 40, no. 7, pp. 121–125, 2013.

[2] C. Yuan, *Research on Multi-Step Attack Detection Method Based on GCT*, Jilin University, Jilin, China, 2010.

[3] C. Chen and B. Q. Yan, “Network attack forecast algorithm for multi-step attack,” *Computer Engineering*, vol. 5, no. 37, pp. 172–174, 2011.

[4] G. Q. Zhai and S. Y. Zhou, “Construction and implementation of multistep attacks alert correlation model,” *Journal of Computer Applications*, vol. 31, no. 5, pp. 1276–1279, 2011.

[5] Z. L. Wang and X. P. Cheng, “An Attack predictive algorithm based on the correlation of intrusion alerts in intrusion response,” *Computer Science*, vol. 32, no. 4, pp. 144–146, 2005.

[6] H. Cao, Q. Q. Wang, Z. Y. Ma et al., “Attack Predition model based on dynamic bayesian games,” *Computer Applications*, vol. 27, no. 6, pp. 1545–1547, 2007.

[7] H. Cao, Q. Q. Wang, Z. Y. Ma et al., “Attack predition model based on static Bayesian game,” *Application Research of Computers*, vol. 24, no. 10, pp. 122–124, 2007.

[8] J.-W. Zhuge, X.-H. Han, Z.-Y. Ye, and W. Zou, “Network attack plan recognition algorithm based on the extended goal graph,” *Chinese Journal of Computers*, vol. 29, no. 8, pp. 1356–1366, 2006.

[9] S. H. Zhang, *Research on Network Security Early Warning Technology Based on Hidden Markov Model*, PLA Information Engineering University, Henan, China, 2007.

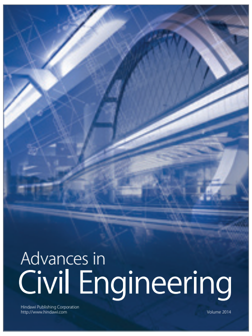
Journal of
Engineering

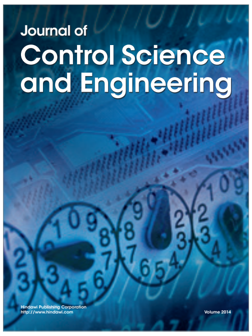
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

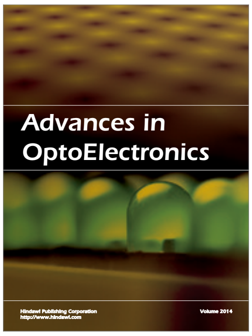
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

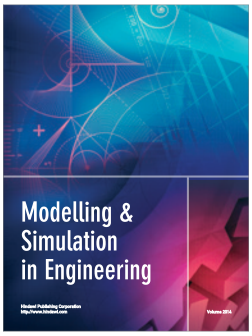
Journal of
Robotics

Journal of
Electrical and Computer Engineering

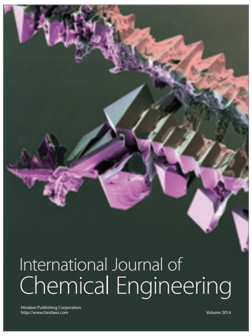

Advances in
OptoElectronics

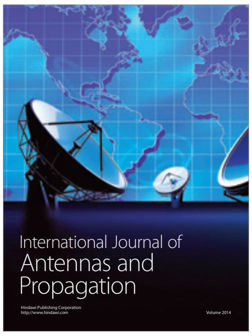
VLSI Design

International Journal of
Navigation and Observation

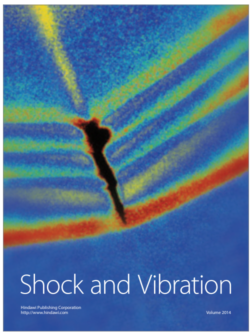

Modelling &
Simulation
in Engineering

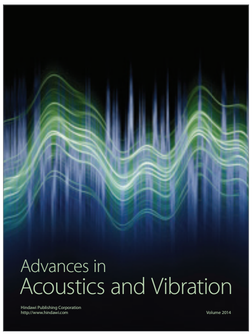
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration