Journal of the Balkan Tribological Association Vol. 22, No 3-I, 2400–2416 (*2016*)

Computer simulations – security risk forecast

NETWORK SECURITY RISK FORECAST BASED ON MARKOV TIME-VARYING MODEL

NI ZHEN[a], LI QIANMUI[a*], GANG LIU[b]

[a] *School of Computer Science and Engineering,*
Nanjing University of Science and Technology, 210094 Nanjing, China
[b] *School of Computer Engineering,*
Suzhou Vocational University, 215104 Suzhou, China
E-mail: liqianmu@126.com

ABSTRACT

Network security risk forecast is a new research focus in the network security field. Though some research has been done on the forecast. Research of single intrusion attack incidents, of which some methods could predict impending single intrusion incidents or partial composite intrusion and attach behavior to some extent has been conducted.
The security risk of networks has the memorylessness property of the Markov model. That indicates the network risk level of future times which is only related to the current state. Past states have no impact on future risk predictions.
The traditional Markov model assumes that the transmission matrix does not vary along with time. In this work, we present a time-varying Markov model through updating the transmission matrix in real-time. In addition, we give the formulation of risk probability calculation. Based on this model, we are able to predict the incurring probability of network risks at a specific future time.

Keywords: computer application, security risk forecast, Markov time-varying model, network attack.

AIMS AND BACKGROUND

Currently, there are two aspects of network security forecast. One is the forecast of singe intrusion attacks or composite attacks. An intrusion attack forecast model based on fuzzy neural network is proposed by Zhang Guiling and others[1], to

* For correspondence.

forecast the intrusion attack intention. A security threat forecast statistics model based on local area network is proposed by Bhattachaya and others[2], to predict the possible security threats that a system may suffer through the statistical analysis of historic vulnerability data. A network security Bayes forecast model based on an Agent is proposed by Pikoulas and others[3]. The model is a multivariate linear statistical model, and the security condition forecast is done by conducting statistical analysis of network users. An attack plan recognition and forecast method based on a causal network is proposed by Qin Xinzhou and others[4]. The method focuses on the correlation of singe attack scenes, the recognition of attacker strategies and intentions, and the forecast for potential attacking behaviors. An attack prediction algorithm based on incident relevance in intrusion response is proposed by Wang Zuli and others[5]. The final purpose of the attack is predicted by analyzing the relevance of attacking incidents. A Bayesian reasoning based intrusion attack forecast method is proposed by Ishida and others[6]. The cycle and quantity of each kind of attack could be found by calculating the conditional probability of each historic attack incident.The increase or decrease of intrusion attack incidents could be effectively predicated. A security alarming method based on the statistical pattern of intrusion incidents is proposed by Zhangfeng and others[7], and the intrusion attack behaviors in the future is predicated through clustering analysis, cyclical analysis, and trend predication. The work above could only forecast single intrusion attack incidents, but couldn't provide security risk forecast for the whole network.

The other aspect is to comprehensively consider each factor and indicator that could affect a systems security, combine network security risk evaluation system, and adopt appropriate forecast methods to predict the security risk trend of the network system. Currently the research methods and models in this field are still not mature. A network security risk forecast method based on RBF neural network is proposed by Renwei and others[8], and the RBF neural network model of network security risk forecast is established after a lot of experiments and training. However, the method comes with disadvantages like too large data volume and basis function selection difficulty. A network attack risk forecast technique of support vector machine is proposed by Zhang Xiang and others[9], and the time series prediction of network attack risk evaluation indicators is conducted with the support vector regression predication method. However, as it is difficult to select a kernel function, this method is not quite practical. Lai Jibao and others propose to use simple weight fusion method to calculate current network security risk[10], and then, utilize gray theory GM(1,1) model to predict future network security risk, however this model is not suitable for conditions when the risk value accumulation curve change is not in line with exponential growth.

In this work, we propose a time-varying Markov model for real-time network risk forecasting. Through a series of real-time updates on the transmission matrix,

our model can effectively predict the incurring probability of network risks at a specific future time.

EXPERIMENTAL

Markov time-varying model. Definition 1. Markov chain is a sequence of random variables X1, X2, X3…. The range of these random variables is called state space. If X_{t+k} is only a function of X_t for the conditional probability distribution of past states, then

$$P(X_{t+k}=i_{t+k}\mid X_1=i_1, X_2=i_2,\ldots, X_t=i_t)=P(X_{t+k}=i_{t+k}\mid X_t=i_t)$$

Here $X_{t+k}=i_{t+k}$ at $t+k$ point of time is in i_{t+k} state.

The identical equation above could be seen to have Markov nature. At time of $t+k$, the probability distribution of system state $X_{t+k}=i_{t+k}$ is only dependent on the state at time t, and has nothing to do with states before t.

Definition 2. A Markov chain model could be expressed as $(S,\mathbf{P},\pi)$ [11], of which:

(1) S is the non-empty state set composed by all the possible states in the system, in other words S is the state space of the system. For example, a state space of network could be $S=\{1,2,\ldots,n\}$, of which 1 denotes primary state and n denotes final state.

(2) $\mathbf{P}=[p_{ij}(t,t+k)]_{m\times n}$ is the system's state transition probability matrix, $p_{ij}(t,t+k)=P\{X_{t+k}=j|X_t=i\}$, $i, j\in S$ denotes the probability of being at state j after k times of state transitions from state i at time t. For the chain, any state at time t would be transitioned to a state in state space after $t+k$, so for any $i\in S$,

$$\sum_{j=1}^{n} p_{ij}\left(t,t+k\right)=1,\ \ 0\le p_{ij}\left(t\right)\le 1, i, j\in S$$

,

(3) $\pi=(\pi_1, \pi_2,\ldots, \pi_n)$ is the initial probability distribution of the system, π_i is the probability of being in state i at the initial time, meeting

$$\sum_{i=1}^{n}\pi_i=1,\ \sum_{j=1}^{n} p_{ij}\left(t,t+k\right)=1,\ \ 0\le p_{ij}\left(t\right)\le 1,\ i,\ j\in S.$$

For $p_{ij}(t,t+k)=P\{X_{t+k}=j|X_t=i\}$, $i, j\in S$, when $k=1$, then $p_{ij}(t,t+1)=p_{ij}(1)$ is the one step state transition probability at time t, so P is the one step state transition probability matrix.

Theorem 1. Let $\{X_t, t=1,2,\ldots,n\}$ be a Markov chain, then for any times u,v,

$$P_{ij}(u+v)=\sum_{k=1}^{n}P_{ik}(u)P_{kj}(v), i, j \in S \tag{1}$$

Proof: for any fixed times s, u, v and states i, j, k, based on conditional probability definition and multiplication theorem,

$$P\{X_{s+u+v}=j, X_{s+u}=k \mid X_s=i\}$$
$$=P\{X_{s+u}=k \mid X_s=i\} \times P\{X_{s+u+v}=j \mid X_{s+u}=k , X_s=i\}=P_{ik}(u)P_{kj}(v) \tag{2}$$

Because of event group $X_{s+u}=k$" ， $k=1,2,\ldots,n$ forms a partition, then

$$P_{ij}(u+v)=P\{X_{s+u+v}=j \mid X_s=i\}=\sum_{k=1}^{n}P\{X_{s+u+v}=j, X_{s+u}=k \mid X_s=i\}.$$

Substitute formula (2) into the formula above, and the theorem is proven.

for1mula (1) in matrix form

$$\mathbf{P}_{u+v}=\mathbf{P}_u\mathbf{P}_v \tag{3}$$

Deduction 1 k step transition probability matrix is the k power of one step transition probability matrix, that is $\mathbf{P}_k=\mathbf{P}\times\mathbf{P}_{k-1}=\mathbf{P}^k$

Proof: use formula (3) and set u=1, v=k-1, get recurrence relation

$$\mathbf{P}_k=\mathbf{P}\mathbf{P}_{k-1}=\mathbf{P}\mathbf{P}\mathbf{P}_{k-2}=\ldots=\mathbf{P}^k$$

Row vector $\boldsymbol{\pi}(k)=(\pi_1(k), \pi_2(k), \ldots, \pi_n(k))$, of which $\pi_j(k)$ denotes the probability of event an being in state j at k time after k times of state transitions from initial ($k=0$) state, and

$$\sum_{j=1}^{n}\pi_j(k)=1 \tag{4}$$

According to the non-after effect property of Markov chain, Bayes conditional probability formula and deduction,

$$\begin{cases}\boldsymbol{\pi}(1)=\boldsymbol{\pi}(0)\mathbf{P}_1=\boldsymbol{\pi}(0)\mathbf{P}\\ \boldsymbol{\pi}(2)=\boldsymbol{\pi}(0)\mathbf{P}_2=\boldsymbol{\pi}(0)\mathbf{P}^2\\ \ldots\\ \boldsymbol{\pi}(k)=\boldsymbol{\pi}(0)\mathbf{P}_k=\boldsymbol{\pi}(0)\mathbf{P}^k\end{cases} \tag{5}$$

In the formula, $\boldsymbol{\pi}(0) = (\pi_1(0), \pi_2(0), \ldots, \pi_n(0))$ is initial state probability vector. $\mathbf{P}_i(i = 1,2,\ldots,k)$ is the i step state transition probability matrix for the system.

It could be seen from the reasoning above that, the traditional Markov forecast model is based on the assumption that a system state transition probability matrix is not changed with time. However, in many practical conditions, especially in a network attack environment, a Markov time-varying model's accuracy is improved by constantly updating the state transition probability matrix. It could be obtained from Formula (5)

$$\begin{cases} \boldsymbol{\pi}(1) = \boldsymbol{\pi}(0)\mathbf{P}_{(0)} \\ \boldsymbol{\pi}(2) = \boldsymbol{\pi}(1)\mathbf{P}_{(1)} \\ \boldsymbol{\pi}(3) = \boldsymbol{\pi}(2)\mathbf{P}_{(2)} \\ \cdots \\ \boldsymbol{\pi}(k) = \boldsymbol{\pi}(k-1)\mathbf{P}_{(k-1)} \end{cases} \tag{6}$$

In the formula, $\mathbf{P}_{(i)}$ denotes the state transition probability matrix at time i.

It could known from the analysis above that, if the initial state of an event at time 0 is known, in other words $\boldsymbol{\pi}(0)$ is known, by using recurrence formula (6), the probability of the event being in various possible states at time k after k times of state transitions could be obtained, i.e $\boldsymbol{\pi}(k)$, therefore the state probability forecast for the event at time k could be obtained. So the key is how to determine the state transition probability matrix $\mathbf{P}_{(i)}$ at different points of time.

The security risks of network mainly come from the hostile attacks. A network attack behavior could be divided into three stages: information gathering stage, attack stage, and attack completion stage. Based on the different stages of attacks, network security risks could be divided into: no security risk L_0(network is not targeted), slight security risk L_1 (network is being targeted), serious security risk L_2(network being attacked), and severe security risk L_3(network has been captured). These different security risk levels constitute the state space of Markov time-varying forecast model, **i.e.** $S = \{ L_0, L_1, L_2, L_3\}$, then the state transitions of network security risks are shown as Fig. 1.

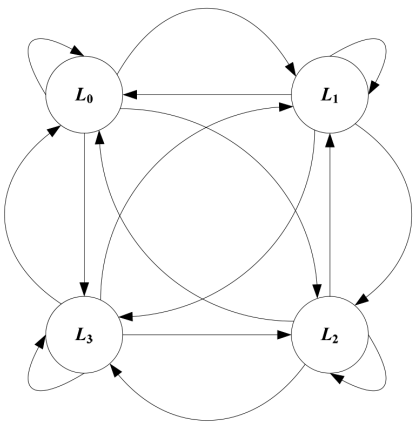

Fig. 1. State transitions of network security risks

then the state transition probability matrix of network security risks is:

$$P = \begin{pmatrix} P_{l_0 l_0} & P_{l_0 l_1} & P_{l_0 l_2} & P_{l_0 l_3} \\ P_{l_1 l_0} & P_{l_1 l_1} & P_{l_1 l_2} & P_{l_1 l_3} \\ P_{l_2 l_0} & P_{l_2 l_1} & P_{l_2 l_2} & P_{l_2 l_3} \\ P_{l_3 l_0} & P_{l_3 l_1} & P_{l_3 l_2} & P_{l_3 l_3} \end{pmatrix} \tag{7}$$

To calculate state transition probability matrix P is to calculate the state transition probability of each state to any other state. The calculation of state transition probability generally adopts the thought of frequency being similar to probability.

$$p_{ij} = \frac{n_{ij}}{\sum_j n_{ij}} \tag{8}$$

in the formula: n_{ij} is the number of samples transitioned to state j from state i;

The real-time iterative algorithm for updating state transition probability matrix P is as follows:

Step 1: initialize history sample data. Input data object set X, input designated cluster number N, and randomly select N objects as initial cluster centers from X. Set iteration ending condition, like max cycle indexes or cluster center convergence error margin;

Step 2: perform iteration. Assign data objects to the closest cluster centers according to similarity criteria, to form a cluster. Set the average vector of each cluster as the new cluster center and reassign data objects.

Step 3: execute Step 2 repeatedly until the end condition is met.

Step 4: when a new sample arrives, calculate cohesion, and get the network risk state cluster that the new sample belongs to. Taking the risk state cluster of the last sample into consideration and perform the statistics of risk state transition numbers.

Step 5: utilize Formula(8) to recalculate the state transition probability, and update original state transition probability matrix.

Cohesion calculation steps are as follows:

Step 1: select Euclidean distance as the cluster partition standard. See new sample records as a n dimension vector, the distance of any two n dimension vectors i and j could be calculated with Euclidean distance, and set $d(\boldsymbol{i}, \boldsymbol{j})$ to be the distance of any two vectors in cluster G,

i.e. $d(\boldsymbol{i}, \boldsymbol{j}) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$

Step 2: calculate the average distance of the cluster $AVG\_dis = \frac{2}{m(m+1)} \sum_{i=1}^{m} \sum_{j=i+1}^{m} d(\boldsymbol{i}, \boldsymbol{j})$ and biggest distance $MAX\_dis = \max(d(\boldsymbol{i}, \boldsymbol{j}))$. Get the cluster cohesion $iner(G) = \frac{MAX\_dis}{AVG\_dis}$.

The higher the cohesion value, the similar the cluster elements will be. Later based on the calculating method of state transition probability matrix, get the new state transition probability matrix. Under the condition of initial state probability known, utilize Formula (6) to get the probability value of each security risk state of network at a certain point of time in the future.

RESULTS AND DISCUSSION

To verify the effectiveness of Markov time-varying model of real-time risk probability forecast, KDD CUP 1999 data set is adopted in the paper for a simulation experiment[12]. Taking the depiction of different attacks by MIT Lincoln Lab intrusion detection experiment into consideration[13], the network security risk level dividing, which corresponds to each attack and is based on the attack stage, is shown in Tab. 1. 4. The network security risk rating is divided into four levels, namely L0, L1, L2 and L3.

Table 1. Classification of attacks

L_0	L_1	L_2	L_3
Normal	Portsweep, Insidesniffer, ipsweep, Nmap, Satan	Guess_passwd, Phf, back, Land, Neptune, Pod,Smurf, Teardrop, buffer_overflow, Imap, Multihop	ftp_write, rootkit, Spy, Warezclient, Warezmaster, Perl, Loadmodule

As the KDD data set is too huge, select one kind of attack data from each attack stage in the KDD data packet for the experiment. The attack combination selected in the paper is nomal, satan, guess_password and rootkit. The experiment extract 2 % normal data (2195 records), all satan data (1589 records), all guess_ password data (53 records) and all rootkit data (10 records) as the training data of the experiment.

There are 41 qualitative and quantitative attribute characters in the KDD CUP 1999 data set, of which there are 8 discrete attribute variables and 33 continuous attribute variables. As the concentration attributes in the data set are excess, the analysis complexity is increased. To ease the analysis burden and increase risk

Table 2. Result of principal component analysis

Component	Initial characteristic value			Extract quadratic sum loading		
	characteristic value	variance contribution %	Accumulated variance contribution %	characteristic value	variance contribution %	Accumulated variance contribution %
1	14.259	52.811	52.811	14.259	52.811	52.811
2	2.269	8.404	61.216	2.269	8.404	61.216
3	1.809	6.701	67.917	1.809	6.701	67.917
4	1.565	5.797	73.714	1.565	5.797	73.714
5	1.071	3.966	77.679	1.071	3.966	77.679
6	1.015	3.758	81.437	1.015	3.758	81.437
7	.969	3.590	85.027			
8	.895	3.316	88.344			
9	.862	3.192	91.536			
10	.761	2.818	94.354			
11	.619	2.291	96.645			
12	.395	1.464	98.109			
13	.142	.526	98.635			
14	.121	.448	99.083			
15	.077	.284	99.367			
16	.057	.211	99.578			
17	.026	.097	99.675			
18	.021	.078	99.753			
19	.018	.065	99.819			
20	.010	.037	99.856			
21	.009	.033	99.889			
22	.008	.029	99.917			
23	.007	.025	99.942			
24	.006	.022	99.964			
25	.005	.019	99.983			
26	.003	.011	99.993			
27	.002	.007	100.000			

forecast efficiency, principal component analysis is adopted in the experiment, to conduct principal component extraction on original attributes, and decrease feature vector dimension numbers. On the premise of guaranteeing the least original data information loss, use less attributes to replace original excess attributes, and most of the original information could be still reflected.

After removing all the character type attribute variables and the attribute variables with the value of zero, the experiment conducts feature extraction on the remaining 26 attributes, and extracts the un-rotated principal components with characteristic value over 1. Table 2 shows the result of the principal component analysis.

It could be seen from Table 2 that the accumulated contribution rates of top 6 principal components reach 81.437 %, the 6 principal components could basically reflect the information of original attributes, and by doing so the number of variables is decreased, 20/26 = 85 % calculation work is saved, and it is better for the overall analysis and research on the issue.

After the principal component analysis, further divides the training data into 4 clusters: C_1, C_2, C_3 and C_4 to represent 4 security level states L_0, L_1, L_2 and L_3. Fig. 2 shows the cluster distribution, Table 3 shows the cluster analysis result, and Fig. 3 shows the distribution trend of original samples and samples in clusters.

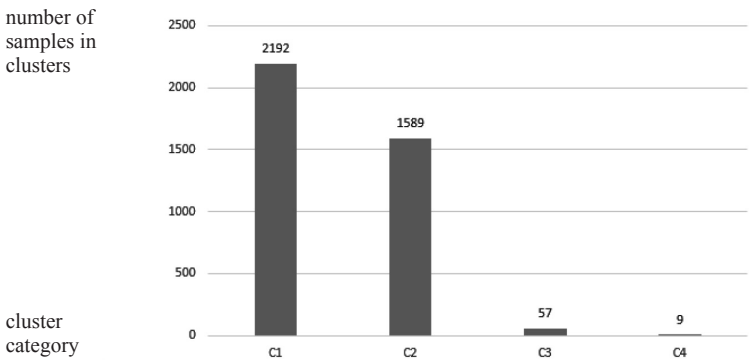

Fig. 2. Cluster distribution of original samples

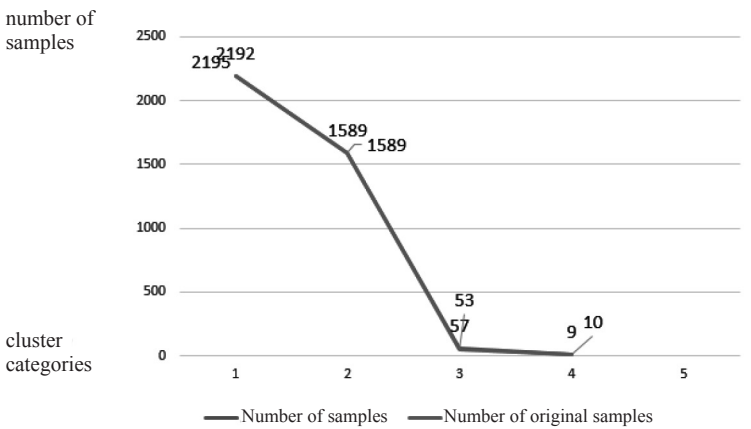

Fig. 3. Cluster distribution trend of original samples

Table 3. Cluster analysis result

Cluster	Number of samples	Number of original samples	Accuracy
C_1	2192	2195	99.86%
C_2	1589	1589	100%
C_3	57	53	92.98%
C_4	9	10	90%

When new sample data arrives, add the new sample data into the 4 clusters, calculate the cohesion of each cluster according to the definition of cohesion, and compare the values of cohesion. The higher the cohesion value is, the new sample will be more similar to the cluster, so we could add new sample data to the cluster with the greatest cohesion.

The statistics of transition times (number of transitions from a state at a time to the next state at the next time) for each state in the training sample data have been conducted, and the statistics result is shown in Table 4.

Table 4. Statistics result of state transition numbers

Next State / State	L_0	L_1	L_2	L_3	Total
L_0	2182	12	1	0	2195
L_1	9	1577	3	0	1589
L_2	3	0	49	1	53
L_3	1	0	0	9	10

Utilize Formula (8) to calculate the transition probability of each state, and the following initial risk state transition probability matrix is got:

$$\mathbf{P} = \begin{pmatrix} 0.99408 & 0.00547 & 0.00045 & 0 \\ 0.00566 & 0.99245 & 0.00189 & 0 \\ 0.05661 & 0 & 0.92452 & 0.01887 \\ 0.1 & 0 & 0 & 0.9 \end{pmatrix} \quad (9)$$

At first the network is normal, so its initial state probability is $\pi(0)$= (1, 0, 0, 0). Multiply $\pi(0)$ with Formula (9), then we have $\pi(1) = \pi(0)\times\mathbf{P}$ = (0.99408, 0.00547, 0.00045, 0) being the current state probability of the network. In other words, the probability of the network being in the 'no security risk' state

is 0.99408, the probability of the network being in the 'slight security risk' state is 0.00547, the probability of the network being in the serious security risk state is 0.00045, and the probability of the network being in the severe security risk state is 0. It could be seen that the possibility of the network being in the no security risk state is the highest, followed by the slight security risk state, and the result is in line with the practical sample.

From the rest 90 % data of kddcup.data.txt, the experiment extracts a sample data set composed by normal, satan, guess_password and rootkit. The sample data set is the test sample, and is divided into four groups.

Time T_1: input the first group test sample, which includes 1000 normal sample data ;

Time T_2: then input second group test sample,which includes 1000 satan sample data;

Time T_3: then input third group test sample, which includes 50 guess_passwd sample data;

Time T_4: at last input forth group test sample, which includes 100 rootkit sample data.

Whenever a new group of test samples arrives, utilize state transition probability matrix update algorithm. First perform cluster classification on the samples, then utilize the calculating method of the risk state transition probability matrix to recalculate state transition probability matrix. At last, utilize Formula (6) to forecast the probabilities of the network in different security risk levels.

Table 5 to Table 8 show the state transition number statistics of the first, the second, the third and the forth group test samples.

Table 5. State transition number statistics result of first group test sample

Next State State	L_1	L_2	L_3	Total
L_0	1	0	0	966
L_1	3	0	0	4
L_2	0	0	0	0
L_3	0	0	0	0

Table 6. State transition number statistics result of second group test sample

Next State State	L_0	L_1	L_2	L_3	Total
L_0	0	1	0	0	1
L_1	0	999	0	0	999
L_2	0	0	0	0	0
L_3	0	0	0	0	0

Table 7. State transition number statistics result of third group test sample

Next State State	L_0	L_1	L_2	L_3	Total
L_0	0	0	0	0	0
L_1	0	0	2	0	2
L_2	0	1	45	1	47
L_3	0	0	1	0	1

Table 8. State transition number statistics result of forth group test sample

Next State State	L_0	L_1	L_2	L_3	Total
L_0	0	0	0	0	0
L_1	0	0	0	0	0
L_2	0	0	0	2	2
L_3	0	0	1	7	8

Then we could get the new state transition probability matrices of the first group, the second group, the third group and the forth group.

$$\mathbf{P}_{(1)} = \begin{pmatrix} 0.999 & 0.001 & 0 & 0 \\ 0.25 & 0.75 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{P}_{(2)} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{P}_{(3)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0.02127 & 0.95746 & 0.02127 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{P}_{(4)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0.125 & 0.875 \end{pmatrix}$$

Based on Formula (6), Table 9 gives the forecast result of the Markov time-varying model.

Based on Formula (5), Table 10 gives the forecast result of the traditional Markov time-varying model.

Fig. 4 is the forecast result of the time-varying Markov model, and Fig. 5 is the forecast result of traditional Markov model.

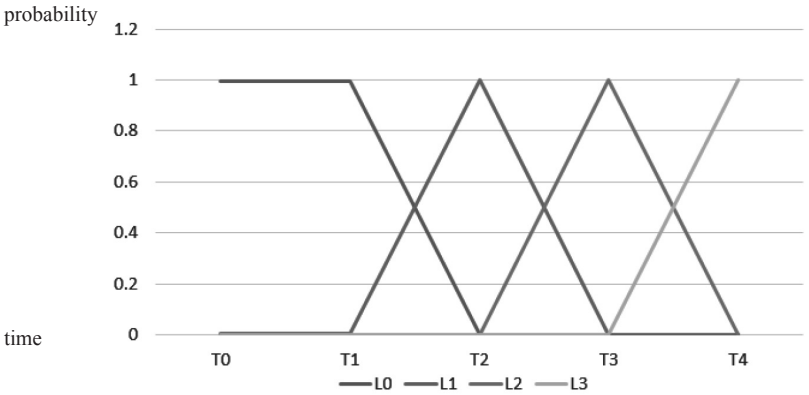

Fig. 4. Forecast result of the time-varying Markov model

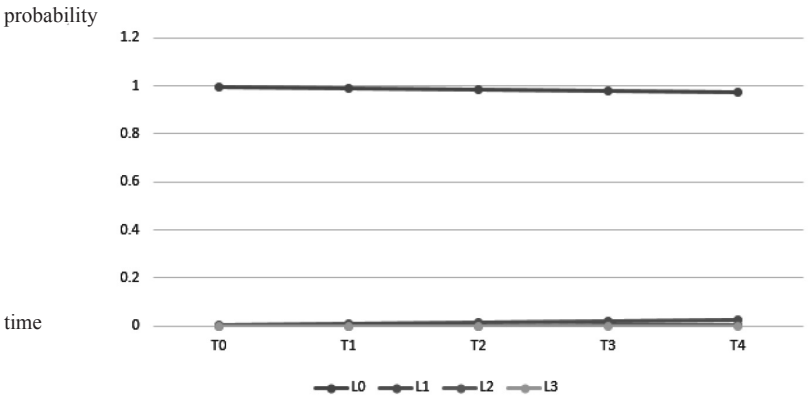

Fig. 5. Forecast result of traditional Markov model

Observed from the experimental results, the transmission rate to high-level risk states of our model is larger. It could be seen from the test data input that the test data at each point of time is of the same kind, so the risk state at each point of time should be L_0, L_1, L_2 and L_3. As the state transition probability matrix of Markov time-varying model is updated with the inclusion of new samples in real time, the network state probability each time is different, which makes the forecast for risks more objective. The simulation experiment indicates that, for the Markov time-varying model, at T_1 the probability of the networking being in L_0 state is the highest; at T_2 the probability of the networking being in L_1 state is the highest; at T_3 the probability of the networking being in L_2state is the highest; at T_4 the probability of the networking being in L_3state is the highest, and the forecast results are in line

Table 9. Forecast result of Markov time-varying model

state probability at last point of time	state transition probability matrix after updating	risk probability forecast			
		L_0	L_1	L_2	L_3
(1,0,0,0)	$\begin{pmatrix} 0.99408 & 0.00547 & 0.00045 & 0 \\ 0.00566 & 0.99245 & 0.00189 & 0 \\ 0.05661 & 0 & 0.92452 & 0.01887 \\ 0.1 & 0 & 0 & 0.9 \end{pmatrix}$	0.99408	0.00547	0.00045	0
(0.99408,0.00547, 0.00045,0)	$\begin{pmatrix} 0.999 & 0.001 & 0 & 0 \\ 0.25 & 0.75 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	0.99445	0.0051	0	0
(0.99445,0.0051,0,0)	$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	0	0.99955	0	0
(0,0.99955,0,0)	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0.02127 & 0.95746 & 0.02127 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	0	0	0. 99955	0
(0,0,0.99955,0)	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0.125 & 0.875 \end{pmatrix}$	0	0	0	0. 99955

Table 10. Forecast result of the traditional Markov time-varying model

state probability at last point of time	state transition probability matrix	risk probability forecast			
		L_0	L_1	L_2	L_3
(1, 0, 0, 0)	$\begin{pmatrix} 0.99408 & 0.00547 & 0.00045 & 0 \\ 0.00566 & 0.99245 & 0.00189 & 0 \\ 0.05661 & 0 & 0.92452 & 0.01887 \\ 0.1 & 0 & 0 & 0.9 \end{pmatrix}$	0.99408	0.00547	0.00045	0
(0.99408, 0.00547, 0.00045, 0)		0.98825	0.01087	0.00087	0.00001
(0.98825, 0.01087, 0.00087, 0.00001)		0.98251	0.01619	0.00127	0.00003
(0.98251, 0.01619, 0.00127, 0.00003)		0.97686	0.02144	0.00165	0.00005
(0.97686, 0.02144, 0.00165, 0.00005)		0.97130	0.02662	0.002	0.00008

with practical test samples. However, about the forecast by the traditional Markov model the state transition probability does not change with time. The forecast results would always show that the probability of the networking being in L_0 state is the highest, which is far from the truth.

CONCLUSIONS

The paper first introduces the relevant research outcomes on forecast issues in the field of network security, and points out that there is a lack of security risk forecast research addressing the overall network, and the forecast methods and models are not mature.

To accurately forecast network security risks, a network-oriented real-time risk forecast Markov time-varying model is proposed in the paper. Using the Markov time-varying model, the risk probability of L0 state, L1 state, L2 state and L3 state is close to 100%, at different moments, and the prediction result is consistent with the ground truth.The model abandons the assumption that in a traditional Markov forecast model the state transition probability matrix does not change with time, and predict the probabilities of security risk states that the network is in in the future by updating the state transition probability matrix in real time. To verify the effectiveness and feasibility of the model on network security risk forecast, in the experiment the model is applied in a network attack environment, DARPA's intrusion test data is utilized, feature extraction and statistical learning is combined, and the probabilities of the network being in different security risk levels are forecast. Compared with traditional Markov forecast model, this model is more real-time, objective and accurate.

ACKNOWLEDGEMENTS

The Fundamental Research Funds for the Central Universities (No. 30916015104); National key research and development program:key projects of international scientific and technological innovation cooperation between governments (No. S2016G9070).

REFERENCES

1. G. ZHANG, J. SUN: A Novel Network Intrusion Attempts Prediction Model Based on Fuzzy Neural Network. Proceedings of the 6th International Conference on Computational Science, Springer Berlin Heidelberg, 419 (2006).
2. S. BHATTACHARYA, S. K.GHOSH: Security Threat Prediction in a Local Area Network Using Statistical Model. Proceedings of IEEE International Conference on Parallel and Distributed Proceeding Symposium, IEEE, 1 (2007).

3. J. PIKOULAS, W. J. BUCHANAN, M. MANNION, et al.: An Agent-Based Bayesian Forecasting Model for Enhanced Network Security. Proceedings of the 8th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems, IEEE, 247 (2001).
4. X. QIN, W. LEE: Attack Plan Recognition and Prediction Using Causal Networks. Proceedings of International Conference on Computer Security Applications, Atlanta, IEEE, 370 (2004).
5. Z. WANG, X. P. CHENG: An Attack Predictive Algorithm Based on the Correlation of Intrusions Alerts in Intrusion Response. J., **32**(4), 144 (2005).
6. C. ISHIDA, Y. ARAKAWA, I. SASASE, ET AL.: Forecast Techniques for Predicting Increase Or Decrease Of Attacks Using Bayesian Inference. Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and signal Processing, IEEE, 450 (2005).
7. F. ZHANG, Z. G. QIN, J. D. LIU: Intrusion Event Based Early Warning Method for Network Security. Computer Science. J., **31**(11),77 (2004).
8. W. REN, X. H. JIANG, T. F. SUN: An RBF Neural Network Based Network Security Situation Forecast Method. Computer Engineering and Applications. J., **42**(31), 136 (2006).
9. X. ZHANG, C. Z. HU, S. H. LIU and others: Research on Network Attack Situation Forecast Technique Based on Support Vector Machine. Computer Engineering. J., **33**(11),10 (2007).
10. J. B LAI, H. Q.W, L. ZHU: Study of Network Security Situation Awareness Model Based on Simple Additive Weight and Grey Theory. Proceedings of 2006 International Conference on Computational Intelligence and Security, IEEE. **2**, 1545 (2006).
11. Q. B. YIN, R. B. ZHANG, X. Y. LI and others: Research on Technology of Intrusion Detection Based on Linear Prediction and Markov Model. Chinese Journal of Computers, J., **28**(5), 900 (2005).
12. KDD CUP 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
13. DARPA. Training data attack description. http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/docs/attacks.html.

Received 27 July 2016
Revised 28 September 2016