CIT. Journal of Computing and Information Technology, Vol. 28, No. 2, June 2020, 89–100
doi: 10.20532/cit.2020.1005178

A Markov-Based Intrusion Tolerance Finite Automaton

Fengquan Li

Xi'an International University, Xi'an, China

It is inevitable for networks to be invaded during operation. The intrusion tolerance technology comes into being to enable invaded networks to provide the necessary network services. This paper introduces an automatic learning mechanism of the intrusion tolerance system to update network security strategy, and derives an intrusion tolerance finite automaton model from an existing intrusion tolerance model. The proposed model was quantified by the Markov theory to compute the stable probability of each state. The calculated stable probabilities provide the theoretical guidance and basis for administrators to better safeguard network security. Verification results show that it is feasible, effective, and convenient to integrate the Markov model to the intrusion tolerance finite automaton.

ACM CCS (2012) Classification: Security and privacy → Network security → Network security

Keywords: finite automaton, intrusion tolerance, Markov, network security

1. Introduction

Network security is commonly maintained by preventing network intrusions and repairing system loopholes through all means in a timely and comprehensive manner. This common method is a passive strategy, owing to the difficulty in predicting network intrusions. Network intrusion is largely inevitable, because the intrusion methods are constantly updated, and unknown system vulnerabilities are hard to find. In this context, network administrators must develop a mechanism to ensure that the network server can continue to provide network services after being invaded. The mechanism that enables the invaded network server to repair itself while providing necessary services is called intrusion tolerance technology.

A system that adopts intrusion tolerance technology is known as a tolerance system. To design such a system, the premise is to recognize the presence of network intrusions in the system. The system design needs to establish a sound mechanism that guarantees the continued operations of the core system functions after the invasion, provides timely remedies for invasion losses, and curbs the invasion as soon as possible, such that the system could resume the normal sate. Currently, the intrusion tolerance system is the third-generation core technology of network security maintenance, serving as the last line of defense for network security. Therefore, it is theoretically and practically significant to study the relevant mechanism of network intrusion tolerance system.

To date, many scholars have tried to implement intrusion tolerance system at home and abroad. Based on hidden semi-Markov model, Bang *et al*. [1] presented a detection scheme for Long-Term Evolution (LTE) signaling attacks on wireless sensor and actuator networks. The scheme could effectively differentiate between attack nodes and common nodes, and enhance the maintenance of intrusion tolerance. Cha and Kang [2] introduced a sequence classification method based on the hidden Markov model to misuse-based intrusion detection. Specifically, the system was treated as a statistical Markov model containing a set of observable states and a set of hidden states, and the model was used to effectively detect the intrusion behavior in the network, highlighting the keys in the maintenance of the tolerance system. To prevent illegal intrusion into the network of large non-res-

idential organizations, Harang and Kott [3] proposed a hidden Markov model with restricted hidden state, which couples Markov chain with Monte-Carlo simulation. By analyzing the combination of intrusion time series, the proposed model parses and predicts the network risks and provides the defense measures.

Drawing on a new feature extraction technology, Khreich *et al*. [4] designed an anomaly detection system that reduces the false alarm rate of intrusion detection. Their system, which integrates the frequency with the time information of the system call trajectory, was trained on single-class support vector machine (SVM) to make accurate forecast of the key points in the tolerance system. Rmayti *et al*. [5] developed a fully decentralized intrusion tolerance prediction mechanism, based on the Bernoulli-Bayesian model for node behavior classification and the Markov chain model based on behavior evolution tracking. Through NS2 simulation, the prediction mechanism was found to accurately detect the key nodes that had been invaded in the tolerant system, which ensures the reliable and safe data packet forwarding between network nodes. Sandhu *et al*. put forward a Markov-based intrusion detection framework that detects the network intrusions in cloud computing [6, 7]. By a two-stage Markov model, the edge devices were effectively divided into four levels, so as to pinpoint the malicious edge devices in cloud computing. The framework was later improved by adding virtual honeypots, and subject to actual attack tests in a virtual environment created by OpenStack and Microsoft Azure. Test results show that the improved framework can identify malicious devices effectively, while reducing the false alarm rate.

Marchang *et al*. [8] presented two Markov-based anomaly detection schemes for intrusion tolerance systems, including a lightweight intrusion detection scheme based on the frequency of statistical data, and another scheme based on Markov chain of data orderliness. The two schemes were proved valid in anomaly detection for intrusion tolerance systems, through simulation and theoretical analysis. Entezari-Maleki *et al*. [9] proposed a two-class random detection model to evaluate the mean time for detecting intrusion nodes in intrusion tolerance systems. Their model simulates each attack with two different continuous-time Markov chains, and gives the calculation method for mean attack detection time. Sadreazami *et al*. [10] created a novel detection framework for distributed blind intrusions into intrusion tolerance systems. Under the framework, the values measured by sensors were treated as target image signals, and their statistical features were used to detect intrusions. Experimental results show that their framework outperforms other schemes of the same type in detecting the attacks on intrusion tolerance systems.

In view of the lack of an intelligent detection algorithm for intrusion tolerance systems, Hajisalem and Babaie [11] combined artificial bee colony (ABC) [12, 13] and artificial fish swarm (AFS) [14, 15] into a hybrid classification method. First, the training dataset was segmented by fuzzy mean clustering and correlation-based feature selection, respectively; the Classification and Regression Tree (CART) was implemented to generate If-Then rules based on the selected features, so as to distinguish between normal records and abnormal intrusion records. Simulation results show that the ABC-AFS hybrid method outshined traditional methods, and achieved the detection rate of 99% and the false alarm rate of 0.01%. Allen *et al*. [16] proposed an intrusion tolerance detection model based on Bayesian reinforcement learning for preventive maintenance related to network security in colleges. In the model, the median estimation learning time metric was introduced to evaluate the speed of the intrusion tolerance system in eliminating parameter uncertainty with probability concentrated in a single scenario. Compared with the alternatives in numerical research, the model was found to have faster learning speed and better detection efficiency. Based on multiple-detector anomaly detection system (ADS), Khreich *et al*. [17] proposed an intrusion detection model for intrusion tolerance systems. Using the Boolean combination in the working feature space of the receiver, the model effectively combines the classification results of different detectors to reduce false alarm rate. Then, the proposed model was verified on two large system call datasets generated on Linux and Windows. The results showed that the model consistently outperformed the single best detector and the homogeneous detector in intrusion detection.

Diddigi *et al*. [18] put forward a reinforcement learning algorithm that tracks intrusions to the tolerance system through upper confidence tree search. With the aid of Markov decision process, the algorithm optimizes the state space and the action space by accurately calculating sensor data, and pinpoints the intrusion nodes in the tolerance system, thereby maintaining the network security at a high speed. Ahmadian Ramaki *et al*. [19] proposed an intrusion tolerance Markov detection model based on machine learning: an isomorphic model was set up to quantify the tolerance system and quickly identify the key nodes being intruded, laying the basis for decision-making on network security. Miehling *et al*. [20] proposed an intrusion tolerance detection model based on the conditional dependency graph. To accurately predict the key nodes and enhance maintenance of the tolerance system, their model simulates the dependency between security conditions (attacker capabilities) and attacks, quantifies the state space and sets up a predictable Markov decision-making process, and calls a scalable online defense algorithm to track the defense behaviors.

Mengistu *et al*. [21] developed a machine learning algorithm for load monitoring of the online tolerance system. Specifically, event-based unsupervised profiling and Markov chain technology were combined with Markov chain technology for system modeling; an additional factor Markov model was used to decompose the generated parameters of equipment model online, and to extract the target nodes, so as to optimize the tolerance system. Sethuraman *et al*. [22] proposed a passive intrusion tolerance detection model for wireless networks that are vulnerable to network intrusions. The model relies on hidden Markov technology to quantify the initial probabilities, identifies the threat nodes in the system through statistical and probabilistic analyses on feedback series, and makes accurate prediction and maintenance to enhance the anti-intrusion capability of the wireless network. To cope with Advanced Persistent Threats (APTs), Brogi and Bernardino [23] derived a hidden Markov model based on APT evolution. An accurate modeling was achieved by reconstructing the evolution process of attack activities. Expert system and artificial intelligence (AI) algorithm were integrated to timely pinpoint potential vulnerabilities of the system. The system will be more capable of resisting the APTs if the identified vulnerabilities are solved.

To safeguard cloud computing networks, Narwal *et al*. [24] combined Markov model with Markov game into a hybrid detection scheme for tolerance systems: the relevant state nodes were trained by training sequences, and the resource trajectories in the system were established with Wireshark network analyzer. In this way, the suspicious nodes were discovered to improve the system's tolerance. Husák *et al*. [25] presented a continuous Markov-based recognition method to recognize the system intrusion intentions: the state nodes of the system were modeled by machine learning and data mining; the nodes that are most likely to be invaded were forecasted, and maintained to improve the overall security of the system. Singhal *et al*. [26] designed an intrusion tolerance detection method based on deep learning, which detects the intrusion nodes of the tolerance system through deep dictionary learning and deep transform learning, using the multi-label classification of deep learning.

To overcome the difficulty in optimizing the training parameters of the Markov model, Chadza *et al*. [27] developed an AI intrusion tolerance detection model. By custom or default rules, the model gives off an alarm during the inspection of the DARPA 2000 MSA dataset. Experiments show that the AI model improves the detection rate of the intrusion tolerance system by 44.95%. Sikder *et al*. [28] proposed a context-aware intrusion tolerance detection system. The system observes changes in a user's sensor data indifferent tasks, and creates a context model to distinguish benign from malicious behaviors of sensors, thereby improving the security of smart devices. To enhance connected vehicles against multiple types of network attacks, Katragadda *et al*. [29] put forward a sequence mining method to detect low-speed injection attacks of the tolerance system in the control area network (CAN), using four types of replay attacks. Their method evaluates the effectiveness of each attack with different attack features and computing performance. Experimental data indicate that the proposed method can effectively detect the low-speed injection attacks on the tolerance system, and greatly improve security of the CAN.

In this paper, the Scalable Intrusion-Tolerant Architecture (SITAR) intrusion tolerance model is optimized by updating the network security strategy, producing an intrusion tolerance finite automaton model. Since every node in the model carries Markov features, the proposed model was quantified by the Markov theory. The stable probability of each state was calculated, and used to determine the key nodes of the system. Then, the maintenance time of each key node was extended, making the system harder to be invaded. The research results provide the direction and theoretical basis for administrators to effectively maintain the network.

2. Optimization of Finite Automata Model for Intrusion Tolerance System

An intrusion tolerance system is an automatic protection system that maintains a network server after the network has been invaded, such that the network server can continue to provide services, while repairing its vulnerabilities. Currently, there is a wide array of network servers and intrusion techniques. Hence, the architecture, security strategy and self-repair algorithm of the intrusion tolerance system are highly flexible and diverse. To effectively describe the function of each node in the system, this paper optimizes the SITAR intrusion tolerance model by updating the network security strategy, and thus proposes an intrusion tolerance finite automaton model. The overall architecture of the proposed model is shown in Figure 1.

The state transition model covers the following basic states: health state *B*, dangerous state *D*, intrusion state *I*, damage avoidance state *A*, tolerance excitation state *E*, reduced service state *R*, safe shutdown state *C*, out of control state *O*, and strategy update state *U*.

Among them, states *A*, *R*, and *C* are special states. Under any of these states, the system needs to learn and analyze the intrusion method automatically or manually by network administrators, and then update the network security strategy to resume the normal state, laying a new basis for system protection in the future. When the system is in an unhealthy state, it can sometimes restore to state *B* automatically, and sometimes require manual restoration.

The state transition model can handle any unknown attack, under which the system service is similar to any known state. This means that the model can deal with unknown forms of attack. By this model, the system operation is divided into several states. For each state, the corresponding security strategy is available to safeguard healthy operation of the system. Therefore, the state transition model is both flexible and secure.

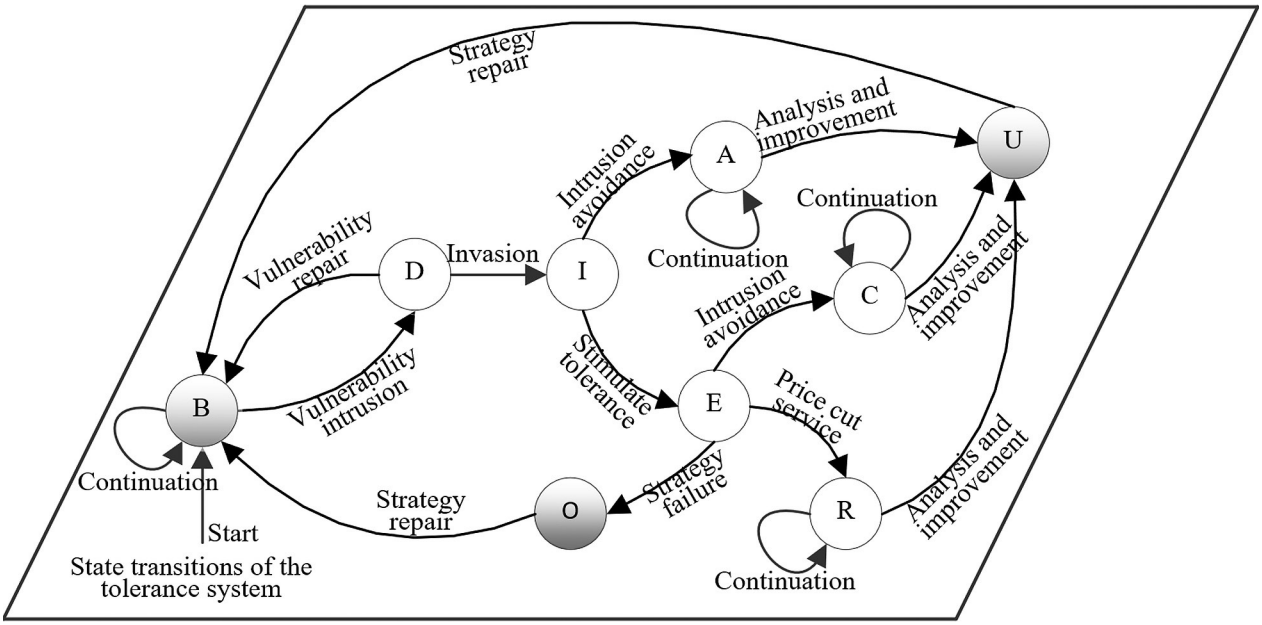

Figure 1. The state transition process of the intrusion tolerance system model.

3. Finite Automaton Conversion of Intrusion Tolerance System

According to the state transition of each node in the intrusion tolerance system (Figure 1), the state transitions of system nodes reflect that the system belongs to the normal state of service provision or to the tolerance state after being invaded. Therefore, each system state in Figure 1 represents a unique working state of the system. Since the system provides a limited number of limited services and working conditions, the operation process of the intrusion tolerance system could be described by finite automaton. Whereas the intrusion into the system is stochastic and unpredictable, the system model is non-deterministic: Even if the working state and conditional symbols are fixed, the model cannot be converted to a unique deterministic state. Hence, the Non-Deterministic Finite Automaton (NDFA) was chosen to formally depict the state transition of the intrusion tolerance system.

3.1 The NDFA of Intrusion Tolerance System

The NDFA of intrusion tolerance system can be formalized as a five-tuple:

$$(Q_0, Ж, Ü, f, F),$$

where, $Q_0 \subseteq Ü$ is the set of nodes with non-empty initial state; $Ж$ is a finite non-empty set of input characters, each of which represents a possible transition condition of the model; $Ü$ is a finite set of nodes with non-empty state, in which each element is a state; f is a multi-valued mapping and a subset of $Ü \times Ж \to Ü$; $F \subseteq Ü$ is the set of nodes with termination states, which can be null.

According to the mapping rules of finite automaton, the mapping $f(Q, \delta) = Q'$ can be expressed as: when the finite automaton is at the state node Q, after the system has received the input condition of character δ, the intrusion tolerance system model will switch to the state node Q'.

The operation process of the NDFA of the intrusion tolerance system can be described by the state transition graph and the state transition table. Suppose there are k state nodes in the NDFA, and these nodes have t characters about the input conditions of transition. Then, the state transition graph of the NDFA will contain k circular nodes, each of which has a maximum of t directed arcs pointing towards other state nodes. The input condition characters are marked on the directed arcs. The state transition graph has one and only one initial state node, but multiple termination state nodes. Each termination state node represents a possible termination state of the intrusion tolerance system.

As shown in Figure 1, the intrusion tolerance system can be abstracted as an NDFA = ($Ü$, $Ж$, f, Q_0, F), where, $Ü = \{B, D, I, A, E, R, C, U, O\}$; $Q_0 = \{B\}$; $F = \{B\}$; $Ж = \{0, \xi, 1\}$. In the finite non-empty set $Ж$, if the input condition character of a state node in the system equals 1, then the security strategy configured at that node has been successfully activated; if the character equals zero, the strategy has not been activated, *i.e.*, the strategy has failed; if the character equals ξ, the state node has been empty-shifted.

The mapping f: $Ü \times Ж \to Ü$ can be expressed as:

$$f(B, 0) = D, f(B, 1) = B;$$
$$f(D, 0) = I, f(D, 1) = B;$$
$$f(I, 0) = A, f(I, \xi) = E;$$
$$f(A, 0) = U, f(A, 1) = [A, U];$$
$$f(E, 0) = O, f(E, 1) = [R, C];$$
$$f(R, 0) = U, f(R, 1) = [R, U];$$
$$f(C, 0) = U, f(C, 1) = [C, U];$$
$$f(O, 1) = B; f(U, 1) = B.$$

Table 1 sums up the state node transitions under different input condition characters.

Table 1. The state transition table of the NDFA of intrusion tolerance system.

State Q	Input condition character $Ж$		
	0	**1**	**ξ**
B	D	B	
D	I	B	
I	A		E
A	U	$[A, U]$	
E	O	$[R, C]$	
R	U	$[R, U]$	
C	U	$[C, U]$	
O		B	
U		B	

3.2. Workflow of the NDFA of Intrusion Tolerance System

The initial state node of the NDFA is *B*. At this time, the model of intrusion tolerance system has just started, and the system belongs to the original state of service provision. Intruders will manipulate the intrinsic vulnerabilities of the system. Once an intrusion takes place, the NDFA will shift from state node *B* to state node *D*, *i.e.*, to the dangerous state.

When the system arrives at the state node *D*, the intruder has just succeeded in launching the invasion, without causing serious damages to the application services or to security strategy of the system. If the administrator identifies the intrusion and takes measures to contain it timely, the system will soon return to the health state node *B*. Otherwise, the system will move from state node *D* to the intrusion sate node *I*.

When the system arrives at the state node *I*, some applications or functions of the network will be damaged by the intruder. The damages might be slight or serious. The degree of damage directly bears on the subsequent operations of the system. If service abnormalities of the system are not noticed by the administrator, but detected and controlled/eliminated by the fault tolerance mechanism in the system, the system will switch into the damage avoidance state node *A*; If service abnormalities of the system are detected, the system will switch into the tolerance excitation state node *E*, kicking off the tolerance mechanism automatically. Based on the judgment of the mechanism, the system will automatically move to the reduced service state node *R* or safe shutdown state node *C*.

If service abnormalities of the system are detected without triggering the tolerance mechanism, the system will switch to the out of control state node *O*. Under this state, the administrator should be notified to take manual measures of maintenance and manually intervene in the system operations.

When the system returns to health state node *B* from *A*, *R* or *C*, the strategy update state node U will learn about the intrusion and update the security strategy, thereby enhancing the anti-intrusion capability of the system and making the system better prepared against future intrusions.

4. Markov Quantification of the NDFA

4.1. State Node Transform Model

Since each node in the NDFA of the intrusion tolerance system carries Markov features, the system state could be quantified by Markov theory to facilitate further analysis. The state of the NDFA can be further determined as

$$\ddot{U} = \{B, D, I, A, E, R, C, U, O\},$$
$$Q_0 = \{B\}, \text{ and}$$
$$F = \{B\}.$$

The state transition probability can be expressed as ρ_i, where $i \in \ddot{U}$. Then, the input condition character table $\ddot{Z}\ddot{K}$ of the NDFA can be further quantified from $\{0, 1, \xi\}$ into

$$\{\rho_b, \rho_{bd}, \rho_{db}, \rho_{di}, \rho_{ie}, \rho_{ia}, \rho_{eo}, \rho_{ec}, \rho_{er}, \rho_a, \rho_r, \rho_c, \rho_{ru}, \rho_{cu}, \rho_{au}, \rho_{ub}, \rho_{ob}\}.$$

The isomorphic process of NDFA state transition before and after quantification is shown in Figure 2.

According to Markov theory, the state transition probability matrix ρ of the DNFA can be expressed as:

$$\rho = \begin{array}{c} \\ B \\ D \\ I \\ E \\ A \\ R \\ C \\ O \\ U \end{array} \begin{array}{c} \begin{array}{ccccccccc} B & D & I & E & A & R & C & O & U \end{array} \\ \begin{bmatrix} \rho_b & \rho_{bd} & - & - & - & - & - & - & - \\ \rho_{db} & - & \rho_{di} & - & - & - & - & - & - \\ - & - & - & \rho_{ie} & \rho_{ia} & - & - & - & - \\ - & - & - & - & - & \rho_{er} & \rho_{ec} & \rho_{eo} & - \\ - & - & - & - & \rho_a & - & - & - & \rho_{au} \\ - & - & - & - & - & \rho_r & - & - & \rho_{ru} \\ - & - & - & - & - & - & \rho_c & - & \rho_{cu} \\ \rho_{ob} & - & - & - & - & - & - & - & - \\ \rho_{ub} & - & - & - & - & - & - & - & - \end{bmatrix} \end{array}$$

According to the actual situation and Markov theory, there exist the following relationships in matrix ρ: $\rho_{db} = 1 - \rho_{di}$, $\rho_{ie} = 1 - \rho_{ia}$, and $\rho_{eo} = 1 - \rho_{er} - \rho_{ec}$. After introducing the state transition probability matrix ρ, the NDFA model satisfies the relationship:

$$f(Q, \rho_i) = \ddot{U}\rho,$$

where, $Q \in \ddot{U}$, and $\rho_i \in \ddot{Z}\ddot{K}$.

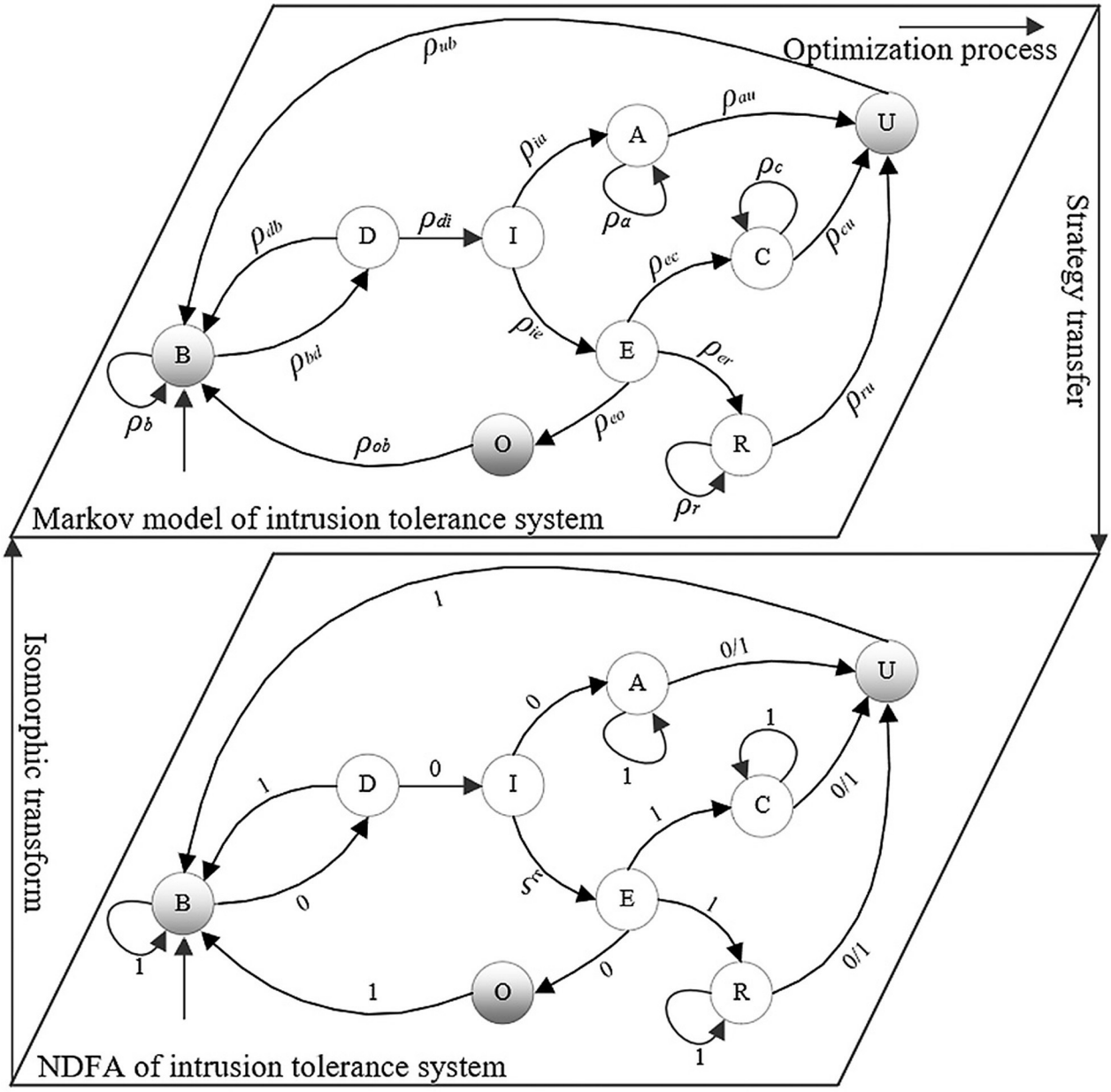

Figure 2. The isomorphic process of NDFA state transition before and after quantification.

4.2. Stable Probability of NDFA

In the NDFA of the intrusion tolerance system, the stable probability refers to the probability that each state is in stable operation. Let P_i be the probability that the state transition model of the system stabilizes at state i; ∂_I be the stabile probability of state i in the state transition model; ρ be the matrix of state transition probabilities of system states. Then,

$$\sum P_i = 1,\ i \in \ddot{U};$$

$$\overline{\partial} = [\partial_B, \partial_D, \partial_I, \partial_E, \partial_A, \partial_R, \partial_C, \partial_O, \partial_U].$$

The P_i value can be computed by:

$$P_i = \frac{\partial_i t_i}{\sum_j \partial_j t_j} \tag{1}$$

The ∂_I value satisfies:

$$\begin{cases} \overline{\partial} = \overline{\partial}\rho \\ \sum_i \partial_i = 1 \end{cases} \tag{2}$$

The risk of intrusion faced by a system, which is characterized by intrusion duration and prob-

ability of successful intrusion, depends on the intrusion ability and technical level of the intruder, and on the defensive measures taken by the administrator. As a result, it is a complex process to model the network intrusion process. To reduce the complexity, the intrusion risk was described by the mean maintenance time of each system state. Let $\{t_B, t_D, t_I, t_E, t_A, t_R, t_C, t_O, t_U\}$ be the mean maintenance time of each state. After introducing parameter T, the stable probability of the NDFA at each state can be calculated by:

$$\begin{cases} P_B = (\rho_{ob} + \rho_{ub} + \rho_{bd}\rho_{db})t_B / T \\ P_D = \rho_{bd} t_D / T \\ P_I = \rho_{bd}\rho_{di} t_I / T \\ P_E = \rho_{bd}\rho_{di}\rho_{ie} t_E / T \\ P_A = \rho_{bd}\rho_{di}\rho_{ia} t_A / T \\ P_C = \rho_{bd}\rho_{di}\rho_{ie}\rho_{eo} t_C / T \\ P_R = \rho_{bd}\rho_{di}\rho_{ie}\rho_{er} t_R / T \\ P_C = \rho_{bd}\rho_{di}\rho_{ie}\rho_{ec} t_C / T \\ P_U = \left(\rho_{bd}\rho_{di}\rho_{ia}\rho_{au} + \rho_{bd}\rho_{di}\rho_{ie}\rho_{er}\rho_{ru} + \right. \\ \quad \left. + \rho_{bd}\rho_{di}\rho_{ie}\rho_{ec}\rho_{cu}\right) t_U / T \\ T = \left(\rho_{ob} + \rho_{ub}\right) t_H + \rho_{bd}\left(t_D + \rho_{db} t_H + \right. \\ \quad + \rho_{di}\left(t_I + \rho_{ia}\left(t_A + \rho_{au} t_U\right) + \right. \\ \quad + \rho_{ie}\left(t_E + \rho_{eo} t_O + \rho_{ec}\left(t_C + \rho_{cu} t_U\right) + \right. \\ \quad \left. \left. \left. + \rho_{er}\left(t_R + \rho_{ru} t_U\right)\right)\right)\right) \end{cases}$$

The greater the stable probability of each state node, the longer the intrusion tolerance system remains in safe operation, the higher the cost of intrusion, and the safer the system.

5. Simulation and Results Analysis

5.1. Environment Construction

Based on the features of the state transition model for intrusion tolerance system, the network topology was plotted as shown in Figure 3, where server S_1 (E-mail server), S_2 (Web server), and S_3 (database server) constitute a complete intrusion tolerance system. The system adopts a unified security strategy to provide network services.

The main servers in the network environment (Figure 3) were subject to vulnerability scan. Then, the vulnerabilities of each server were identified (Table 2).

Table 2. Vulnerabilities of the devices in the tolerance system.

Device number	Operating system	Vulnerability
S_1	Windows Server 2003	CVE-2007-0038
		CVE-2004-0893
S_2	Windows Server 2003	CVE-2008-0702
		CVE-2004-2575
S_3	Windows Server 2003	CVE-2006-2379
		CVE-2002-0364

5.2. Test on System Model and Data Analysis

Through active attacks, this paper obtains relevant data on intrusion into the tolerance system model and organizes these data into the state transition probability matrix ρ of the NDFA. The system can operate at state nodes B, R, C, and A, that is,

$$\rho_b = \rho_a = \rho_r = \rho_c = 1.$$

For simplicity, these state nodes were ignored, *i.e.*, it was assumed that

$$\rho_b = \rho_d = \rho_r = \rho_c = 0.$$

Moreover, the system will resume operation after being maintained by the administrator. Hence,

$$\rho_{ub} = \rho_{ob} = \rho_{au} = \rho_{ru} = \rho_{cu} = 1.$$

Since the system inevitably has vulnerabilities and the vulnerabilities will eventually be discovered by the intruder, we have

$$\rho_{bd} = 1.$$

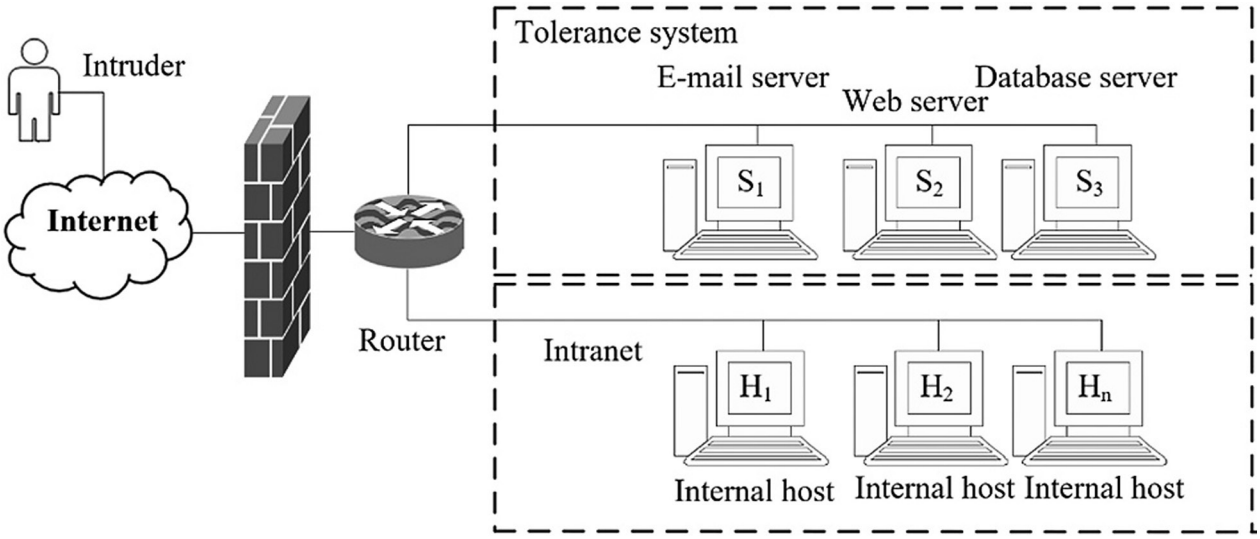

Figure 3. Network topology of the state transition model for intrusion tolerance system.

As shown in Figure 3, each server in the tolerance system contains lots of vulnerabilities. The probability that these vulnerabilities are successfully utilized by the intruder is

$$\rho_{di} = 0.5;$$

the probability that these vulnerabilities are detected and repaired timely by the system is

$$\rho_{db} = 1 - \rho_{di} = 0.5;$$

the probability that the intrusion is detected and avoided by the system is

$$\rho_{ia} = 0.4.$$

According to Markov theory, the probability that the intrusion tolerance system detects the intrusion and triggers the security strategy is

$$\rho_{ie} = 1 - \rho_{ia} = 0.6;$$

the probability that the system detects the intrusion and adopts a reduced service strategy is

$$\rho_{er} = 0.5;$$

the probability that the system detects the intrusion and terminates the provision of network services is

$$\rho_{ec} = 0.4;$$

the probability that the system is damaged by the intrusion and forced to terminate service provision is

$$\rho_{eo} = 1 - \rho_{er} - \rho_{ec} = 0.1.$$

Survey results show that, among all states, the model spends a relatively long time at state *T*. Hence, it was assumed that $t_B = 1.0$, and $t_D = 1.8$. The intrusion was detected after $t_I = 0.4$. Then, the system state switched from *I* to *A*. State *A* was noticed after $t_A = 0.5$. Only through strategy learning could state *A* move back to state *B*. The strategy learning consumed $t_U = 0.5$. Under state *E*, the transition direction was determined by the tolerance strategy. Hence, it was assumed that $t_E = 0.2$, $t_R = 4.0$, $t_C = 1.5$, and $t_O = 2.5$. Note that all $t_i (i \in Ü)$ variables are time units. The maintenance time of each state node is illustrated in Figure 4.

According to the maintenance time ti of the system (Figure 4), distribution of the stable probability P_i for each state in the tolerance system can be obtained by formulas (1)-(3) (Figure 5).

As shown in Figure 5, the stable probabilities of different states can be ranked in descending order as $\{B, D, R, U, I, C, A, O, E\}$. Among them, at states *U* and *O*, the system cannot return to state *B* without manual intervention. Hence, the stable probabilities of the two states have nothing to do with the maintenance of network security, and could therefore be neglected. Hence, the key state nodes of the system could be defined as $\{B, D, R\}$. By extending the maintenance time t of these state nodes, the invasion tolerance system could be more reliable. In this way, the invasion will be more difficult to implement, and more likely to be prevented.

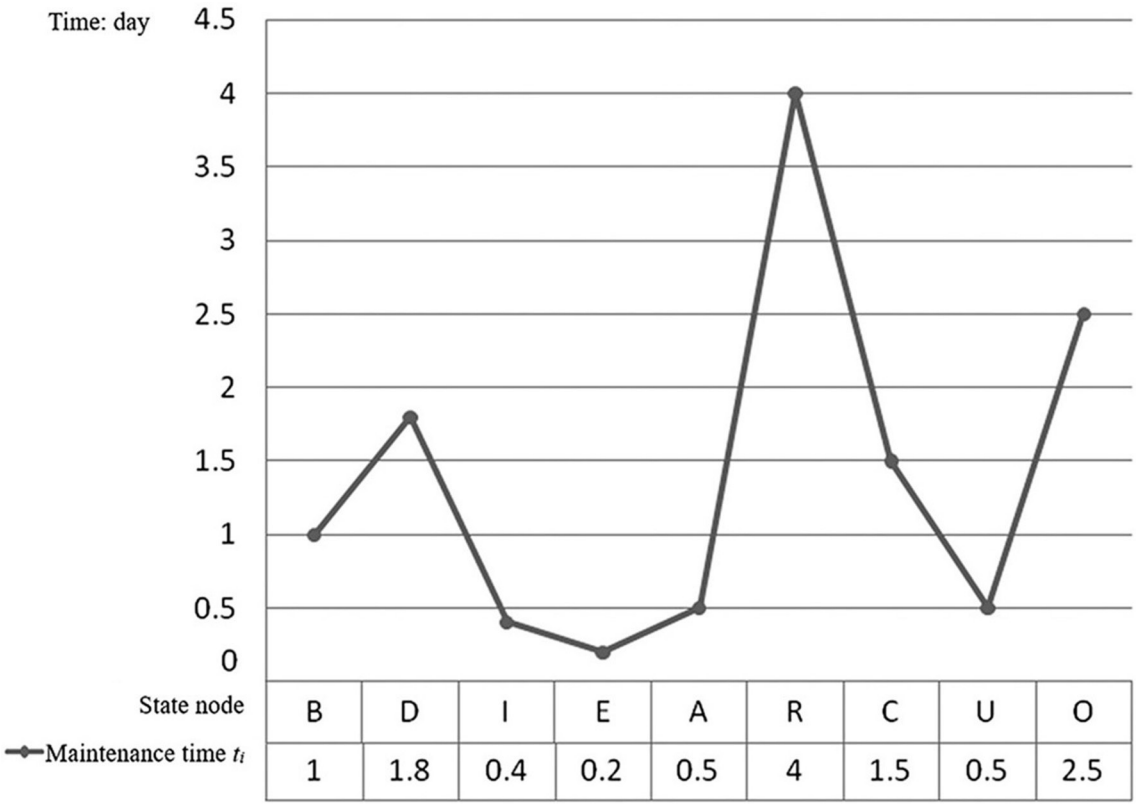

Figure 4. Distribution of the maintenance time t_i of the system.

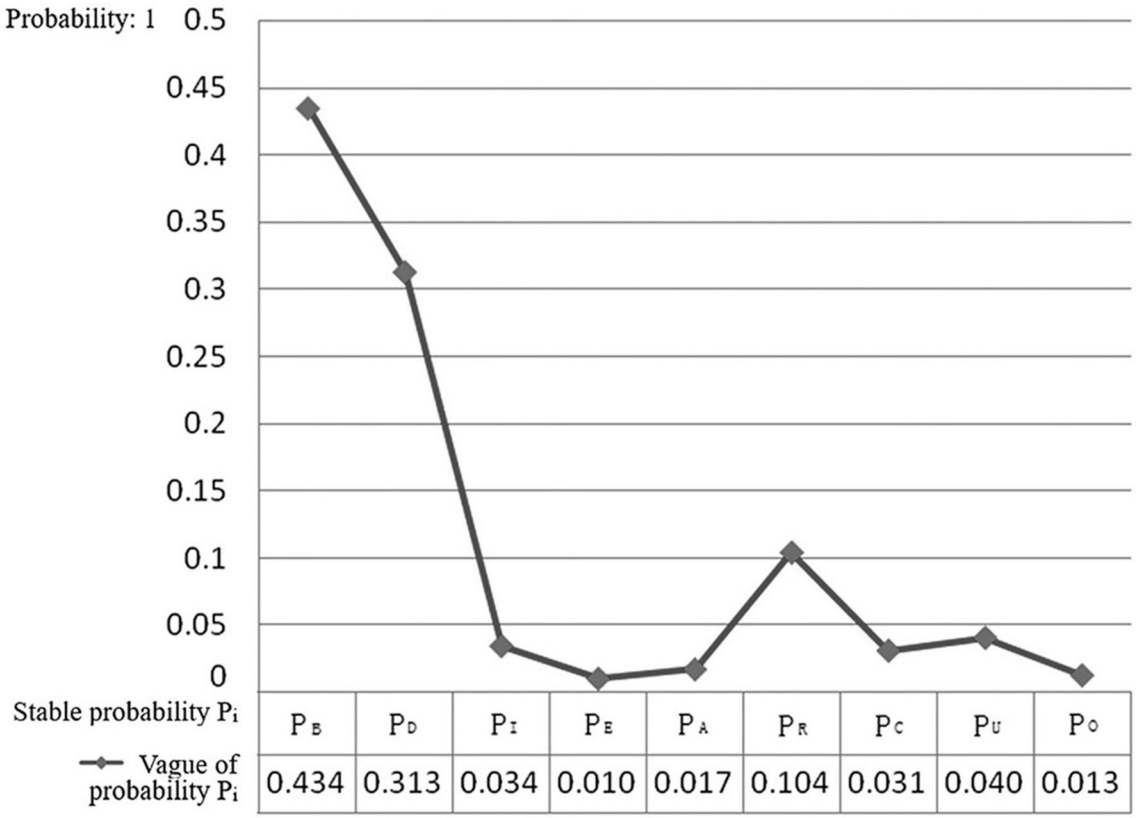

Figure 5. Distribution of the stable probability P_i for each state in the tolerance system.

6. Conclusion

The intrusion tolerance system is the third-generation core technology of network security maintenance, serving as the last line of defense for network security. The system plays an important role in network security management and attracts much attention from network security experts. By updating the security strategy, this paper optimizes the existing SITAR intrusion tolerance model and proposes an intrusion tolerance finite automaton. The proposed model was quantified by Markov theory, the stability probability was calculated for each state, and the key state nodes of the tolerance system were determined as {*B*, *D*, R}. If the maintenance time at key state nodes is extended, it will be more difficult to invade the system, and the system model will become more secure. The future research will further improve the feasibility and reduce maintenance time of the system model by dynamically maintaining vulnerabilities and updating security strategy in the intrusion tolerance system.

References

[1] J. H. Bang *et al.*, "Anomaly Detection Of Network-Initiated LTE Signaling Traffic in Wireless Sensor and Actuator Networks Based on A Hidden Semi-Markov Model", *Computers & Security*, vol. 65, pp. 108–120, 2017. https://doi.org/10.1016/j.cose.2016.11.008

[2] [2] K. H. Cha and D. K. Kang, "Experimental Analysis of Hidden Markov Model Based Secure Misuse Intrusion Trace Classification and Hacking Detection", *Journal of Computer Virology and Hacking Techniques*, vol. 13, no. 3, pp. 233–238, 2017. https://doi.org/10.1007/s11416-017-0293-7

[3] R. Harang and A. Kott, "Burstiness of Intrusion Detection Process: Empirical Evidence and a Modeling Approach", *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2348–2359, 2017. https://doi.org/10.1109/TIFS.2017.2705629

[4] W. Khreich *et al.*, "An Anomaly Detection System Based on Variable N-Gram Features and One-Class SVM", Information and Software Technology, vol. 91, pp. 186–197, 2017. https://doi.org/10.1016/j.infsof.2017.07.009

[5] M. Rmayti *et al.*, "A Stochastic Approach for Packet Dropping Attacks Detection in Mobile AD HOC Networks", *Computer Networks*, vol. 121, pp. 53–64, 2017. https://doi.org/10.1016/j.comnet.2017.04.027

[6] R. Sandhu *et al.*, "Identification of Malicious Edge Devices in Fog Computing Environments", *Information Security Journal: A Global Perspective*, vol. 26, no. 5, pp. 213–228, 2017. https://doi.org/10.1080/19393555.2017.1334843

[7] A. S. Sohal *et al.*,"A Cybersecurity Framework to Identify Malicious Edge Device in Fog Computing and Cloud-of-Things Environments", *Computers & Security*, vol. 74, pp. 340–354, 2018. https://doi.org/10.1016/j.cose.2017.08.016

[8] N. Marchang *et al.*, "Detecting Byzantine Attack in Cognitive Radio Networks by Exploiting Frequency and Ordering Properties", *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 816–824, 2018. https://doi.org/10.1109/TCCN.2018.2845382

[9] R. Entezari-Maleki *et al.*, "IDS Modelling and Evaluation in WANETs Against Black/Grey-Hole Attacks Using Stochastic Models", *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 27, no. 3, pp. 171–186, 2018. https://doi.org/10.1504/IJAHUC.2015.10001797

[10] H. Sadreazami *et al.*, "Distributed-Graph-Based Statistical Approach for Intrusion Detection in Cyber-Physical System", *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 137–147, 2017. https://doi.org/10.1109/TSIPN.2017.2749976

[11] V. Hajisalem and S. Babaie, "A Hybrid Intrusion Detection System Based on ABC-AFS Algorithm for Misuse and Anomaly Detection", *Computer Networks*, vol. 136, pp. 37–50, 2018. https://doi.org/10.1016/j.comnet.2018.02.028

[12] S. Harifi *et al.*, "Using Metaheuristic Algorithms to Improve K-Means Clustering: A Comparative Study", *Revue d'Intelligence Artificielle*, vol. 34, no. 3, pp. 297–305, 2020. https://doi.org/10.18280/ria.340307

[13] A. Arshaghi *et al.*, "Detection of Skin Cancer Image by Feature Selection Methods Using New Buzzard Optimization (BUZO) Algorithm", *Traitement du Signal*, vol. 37, no. 2, pp. 181–194, 2020. https://doi.org/10.18280/ts.370204

[14] D. S. Zhang *et al.*, "Aquifer Parameter Inversion by Artificial Fish Swarm Algorithm Based on Quantum Theory", *Ingenierie des Systemes d'Information*, vol. 24, no. 1, pp. 29–33, 2019. https://doi.org/10.18280/isi.240103

[15] L. Kaddouri *et al.*, "Design of Two-Dimensional Recursive Digital Filter Using Multi Particle Swarm Optimization Algorithm", *Journal Eu-*

ropéen des Systèmes Automatisés, vol. 53, no. 4, pp. 559–566, 2020. https://doi.org/10.18280/jesa.530415

[16] T. T. Allen *et al.*, "Reward-Based Monte Carlo-Bayesian Reinforcement Learning for Cyber Preventive Maintenance", *Computers & Industrial Engineering*, vol. 126, pp. 578–594, 2018. https://doi.org/10.1016/j.cie.2018.09.051

[17] W. Khreich *et al.*, "Combining Heterogeneous Anomaly Detectors for Improved Software Security", *Journal of Systems and Software*, vol. 137, pp. 415–429, 2018. https://doi.org/10.1016/j.jss.2017.02.050

[18] R. B. Diddigi *et al.*, "Novel Sensor Scheduling Scheme for Intruder Tracking in Energy Efficient Sensor Networks", *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 712–715, 2018. https://doi.org/10.1109/LWC.2018.2814576

[19] A. Ahmadian Ramaki *et al.*, "A Systematic Review on Intrusion Detection Based on the Hidden Markov Model", *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 11, no. 3, pp. 111–134, 2018. https://doi.org/10.1002/sam.11377

[20] E. Miehling *et al.*, "A POMDP Approach to The Dynamic Defense of Large-Scale Cyber Networks", *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2490–2505, 2018. https://doi.org/10.1109/TIFS.2018.2819967

[21] M. A. Mengistu *et al.*, "A Cloud-Based On-Line Disaggregation Algorithm for Home Appliance Loads", *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3430–3439, 2018. https://doi.org/10.1109/TSG.2018.2826844

[22] S. C. Sethuraman *et al.*, "Intrusion Detection System for Detecting Wireless Attacks in IEEE 802.11 Networks", *IET networks*, vol. 8, no. 4, pp. 219–232, 2018. https://doi.org/10.1049/iet-net.2018.5050

[23] G. Brogi and E. D. Bernardino, "Hidden Markov Models for Advanced Persistent Threats", *International Journal of Security and Networks*, vol. 14, no. 4, pp. 181–190, 2019. https://doi.org/10.1504/IJSN.2019.103147

[24] P. Narwal *et al.*, "A Hidden Markov Model Combined With Markov Games for Intrusion Detection in Cloud", *Journal of Cases on Information Technology (JCIT)*, vol. 21, no. 4, pp. 14–26, 2019. https://doi.org/10.4018/JCIT.2019100102

[25] M. Husák *et al.*, "Survey of Attack Projection, Prediction, and Forecasting in Cyber Security", *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 640–660, 2018. https://doi.org/10.1109/COMST.2018.2871866

[26] V. Singhal *et al.*, "Simultaneous Detection of Multiple Appliances from Smart-Meter Measurements via Multi-Label Consistent Deep Dictionary Learning and Deep Transform Learning", *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2969–2978, 2018. https://doi.org/10.1109/TSG.2018.2815763

[27] T. Chadza *et al.*, "Analysis of Hidden Markov Model Learning Algorithms for the Detection and Prediction of Multi-Stage Network Attacks", *Future Generation Computer Systems*, vol. 108, pp. 636–649, 2020. https://doi.org/10.1016/j.future.2020.03.014

[28] A. K. Sikder *et al.*, "A Context-Aware Framework for Detecting Sensor-Based Threats on Smart Devices", *IEEE Transactions on Mobile Computing*, vol. 19, no. 2, pp. 245–261, 2019. https://doi.org/10.1109/TMC.2019.2893253

[29] S. Katragadda *et al.*, "Detecting Low-Rate Replay-Based Injection Attacks on In-Vehicle Networks", *IEEE Access*, vol. 8, pp. 54979–54993, 2020. https://doi.org/10.1109/ACCESS.2020.2980523

Received: January 2021
Revised: February 2021
Accepted: March 2021

Contact address:
Fengquan Li
Xi'an International University
Xi'an
China
e-mail: lifengquanedu@126.com

Fengquan Li graduated from the Northwest University and has obtained a MSc degree in computer software and theory. He is now working in Xi'an International University as an associate professor. His main research fields are computer network security, image processing, and higher education research.