Research Article

Autonomous Cognitive Model and Analysis for Survivable System

Yiwei Liao,¹ Guosheng Zhao ,¹ and Jian Wang²

¹College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China

²School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150001, China

Correspondence should be addressed to Guosheng Zhao; zgswj@163.com

Received 12 June 2020; Revised 31 July 2020; Accepted 7 August 2020; Published 27 August 2020

Academic Editor: Frederic Kratz

Copyright © 2020 Yiwei Liao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The research on autonomous recognition mechanism for survivability has vigorously been growing up. A method of autonomous cognitive model and quantitative analysis for survivable system was proposed based on cognitive computing technology. Firstly, a cognitive model for survivable system with cross-layer perception ability was established, a self-feedback evolution mode of cognitive unit based on monitor-decide-execute loop structure was improved, and a self-configuration of cognitive unit is realized. Then, combined with the cognitive state transition graph, the analysis of cognitive performance for survivable systems based on dynamic cognitive behavioral changes was constructed. Finally, the cognitive processes of survivable system were described by using formal modeling. Simulation validated the influence degree of test parameters on system survivability from two perspectives of the probability of intrusion detection systems vulnerability and attacks detected. Results show that enhancing the rate of monitoring actions change and the rate of performing actions change obviously improved the cognitive performance of survivable system.

1. Introduction

Survivability is a hot topic in the research on the next-generation Internet security. According to Westmark [1] and Ellison [2] definition, survivability can be illustrated from three properties: resistance, recognition, and recovery. Among them, recognition reflects the system's autonomous cognition of its own survival situations and securities of the scene and environment. Current research focuses more on the definition of survivability [1, 2], quantitative and qualitative evaluation [3–5], formal description [6–8], trusted protection [9], recovery [10], and other topics in resistance and recovery. But the research on recognition has just begun and is growing.

At present, consensus on the research of survivability mechanism have been achieved at home and abroad as follows. Recognition refers to the ability that the system possesses to “know” and “feel” the current system's survival situation [11]. Survivability-oriented recognition gives priority to the perception and cognition of the security status of the whole system environment, which can be regarded as the identification of basic key services' decline in survivability

and of the attack and intrusion event sets [12]. Recognition means the system's response and adaptability when systems face malicious intrusion [13], which can reflect systems' ability to assess its own security status and surrounding working environment, which can be analyzed from its recognition rate of security incidents and the recognition time of nonsecurity incidents. Recognition can be achieved by constraining reference thresholds of cognition parameters, while autonomy can be achieved by the central control process of the autonomous recognition unit [14]. Recognition can be obtained by establishing a hierarchical perception model and making the policy library drive the self-management mode of the monitor-decide-execute (MDE) loop structure [15]. Cognitive Computing is a summary of the characteristics of the next-generation intelligent Internet's core concepts [16]. Cognitive computing in the era of big data is approaching cognitive science, with the abilities of self-learning, self-adaptation, and self-perception to realize the human-brain-like recognition and judgment. In this paper, based on previous survivability researches, an autonomous cognitive model of survivable system is raised, and the model is formalized by using semi-Markov

stochastic process algebra [17, 18], which provides theoretical guidance for the study of survivable system's cognitive ability.

2. Autonomous Cognitive Model

The system's cognitive needs are mapped to the dynamic selection of multiobjective cognitive results at multiple cognitive levels. Meanwhile, the cross-layer perception is used to obtain the autonomous reasoning, dynamic decision-making, and resource reallocation of survivable systems, and to realize the self-adaptation to dynamic changes of the cognitive needs and environment security. In addition, cognitive model should reach a balance between formal description and cognitive abstraction, so it can not only accurately describe and reflect the system's recognition, but also facilitate reasoning, thus providing theoretical support for the study of cognitive ability of survivable system.

2.1. Cross-Layer Recognition. According to different emphasis on cognitive process, cognitive needs, and cognitive elements, the survivable system can be divided into three cognitive layers, namely, access cognitive layer, network cognitive layer, and service cognitive layer, as shown in Figure 1.

Access cognition layer reflects the recognition of the communication capability of available transmission channels, which supports protocol conversion and adaptation of various available channels, and achieves high reliable information transmission through the recognition of channels' communication capability.

The network cognitive layer shows the unified cognition of cognitive specifications in the cognitive process and goals of cognitive needs. It can realize dynamic reconfiguration and planning constraints of cognitive network resources.

The service cognitive layer reflects the recognition of the matching ability of providing Internet resources required for applications and users. It can serve high QoS service in complex environment, where massive, incomplete, or even malicious service scenarios exist.

2.2. Self-Feedback Mode of Cognitive Units. The cognitive unit structure is similar to Agent in the traditional sense, the basic unit of the realization of cognitive model [19–21], which is also the symbol of the autonomous cognitive ability of survivable system. With the self-feedback ability added on the basis of the existing cognitive unit structure, an improved cognitive loop structure is achieved as shown in Figure 2. This structure is a self-feedback evolving structure driven by the self-configuring strategy library of cognitive elements (M-D-E: Monitor-Decide-Execute), which can adjust behaviors, topology, and service parameters with the changes of working environment and task objectives inside and outside the survivable system. Apart from the function of perceiving contexts of normal network events, the structure can also deal with internal and external security threats to enable survivable systems to independently adapt to environment and demand changes.

The self-feedback mechanism of cognitive unit is shown in Figure 3, which includes local, domain-level, and global feedbacks. Each layer is composed of several cognitive units to achieve global, domain, and local cognition of the system's cognitive behaviors. Results of local feedback can obtain the local optimal solution to the goal of cognitive needs; global feedback can coordinate the feedback results at domain and local levels and obtain the global optimal or suboptimal solution.

Cognitive units can obtain self-configuration of cognitive elements with a self-feedback mechanism. There are two cases:

2.2.1. Preset Self-Configuration. When matchable strategies are found in existing strategy libraries, the configuration strategy in the preset cognitive rule set will analyze and reason the system, as shown in Figure 4.

2.2.2. Acquired Self-Configuration. When matchable strategies cannot be found in strategy library, effective rules achieved after acquisition will be stored as acquired rules in the configuration strategy library, as shown in Figure 5.

3. Cognitive Process of Formal Modeling

In order to formally describe the transition between different states of the system under attacks, faults, or accidental failures, and to better understand the dynamic evolution process of the survivable system's survival situations, a cognitive survival state transition diagram [14] is introduced, as shown in Figure 6.

The tool Version v25 of the PEPA Eclipse Plugin [22] of the Computer Science Foundation Laboratory of the University of Edinburgh is used to simplify the calculation process. The formal description of the cognitive survival state for the survivable system in Figure 6 is as follows:

- (i) Intruder: = (searching, h). Attack;
- (ii) Attack: = (starck_attack, p). (attack, k). Attack + (starck_attack, g). Intruder;
- (iii) General: = (attack, z1). Compromised + (failing, z2). Compromised + (error, z3). Compromised;
- (iv) Compromised: = (probe, w1). Detection + (mask, w2). General;
- (v) Detection: = (start_probe, L1). (emergency1, p1). SelfDestruction; + (start_probe, L2). (healing, L3). SelfHealing;
- (vi) SelfHealing: = (start_healing, L3). (sealheaking, s1). General + (start_healing, L4). (emergency2, p2). SealDestruction + (start_healing, L5). (selfhealing, s2). SelfHealing;
- (vii) SelfDestruction: = (start_destroy, L6). (destroy, delta: L (s)). (backup, s3). General

The parameters and their meanings are shown in Table 1.

The cognitive model of survivable systems can be formalized as a quintuple form (Mde, Objects, Domain), where $Mde = Mde_1, Mde_2, \dots, Mde_m$ represents the resource constraint sets of m cognitive units, $C = C(Mde_i)$ represents the

FIGURE 1: The cross-layer sensing cognitive model.

FIGURE 2: The improved cognitive loop structure.

cognitive sublayer at layer j , including i cognitive unit resources; $\text{Objects} = \{\text{Object}_1, \text{Object}_2, \dots, \text{Object}_n\}$ is a set of cognitive needs' objectives. The single objective Object_k is associated with the i^{th} cognitive sublayer $C_j = C(\text{layer}_i)$ and satisfies the mapping function $\rho: \text{Object}_k \rightarrow C_j$. If there are multiple cognitive needs objectives in a cognitive sublayer, it can be expressed by a union set: $\text{Object}_{s_1} \cup \text{Object}_{s_2} \cup \dots \cup \text{Object}_{s_q}$; Domain represents the

set of cognitive domains, and each subnet i is regarded as Domain_i ; $\Sigma = \{\zeta_1, \zeta_2, L, \zeta_n\}$ is the set of action decision result functions.

Survivable systems provide key services to the outside world, and users request services. Therefore, from the perspective of service supply, the survivable system is modeled as two ends: User and Server. The User end can be represented as User_j^k process. The formal description of user end is

- (i) Monitor = (monitor, m). Decide;
- (ii) Decide = (decide, r1). Execute + (uncertain, r2). Learn;
- (iii) Learn = (learning, r3). Decide;
- (iv) Execute = (execute, r4). Monitor + (service1, s1). Monitor + (service2, s2). Monitor + ... + (servcet, st). Monitor

And a model for Server end is made, objective $k \in \text{Objects}$, and the process of Server end is represented as Server_{ji}^k , where $i \in \text{Domain}$, $j \in C(\text{layer}_i)$, which satisfy $\rho: k \rightarrow j$. For different Domain_i processes, the rate and number of action changes are different. Therefore, the cognitive process can be shown as

$$\text{User}_j^k || \{\text{service}^k\} \text{Server}_{ji}^k$$

FIGURE 3: The self-feedback model based on the cognitive loop.

FIGURE 4: Self-configuration in preset cognitive rule sets.

FIGURE 5: Self-configuration in acquisition cognitive rule sets.

FIGURE 6: The transition diagram of cognitive survival state.

TABLE 1: The explanation of parameters in SM-PEPA.

Parameters	Meanings
H	Probability of finding system flaws
P	Probability of persistent attack
K	Duration of attacks
G	Rate of invaders abandoning attacks
Z_1	Probability of detecting intrusions
Z_2	Probability of stochastic failures entering compromise state
Z_3	Probability of entering compromise state because of faulty operations
W_1	Probability of compromise state entering detection state
W_2	Probability of blocking flaws
L_1	Probability of detecting state entering self-destructive state
P_1	The duration of the detected state to the self-destructive state
L_2	Probability of calling recovering updates from detection state
h_1	Duration of calling recovery updates
L_3	Probability from self-recovery state to normal state
S_1	Duration of self-recovery state to normal state
L_4	Probability of self-recovery state to self-destruction state
P_2	Duration of self-recovery state to self-destruction state
L_5	Probability of self-restoring state to self-restoring state
S_2	Duration of self-recovery state to self-recovery state
L_6	Repair probability from self-destructive state to normal state
S_3	Repair duration from self-destructive state to normal state

And service is the collection of all service interactions. For all the cognitive needs of the system, the formal description of server end is

Modelcognitive = { | (k) = User || {monitork, executek}
Sever }, kobjects, j{C (layer1), C (layer2), ..., C (layerp), i
Domain

4. Quantitative Analysis

The PEPA Workbench and Eclipse Plugin cognitive tool is used to quantitatively analyze the model of survivable

system, and the quantitative analysis results are obtained from the perspective of steady-state probability.

4.1. Solution of Steady-State Probability. For SM-PEPA, if any component P satisfies the formula: $P \xrightarrow{(a_1, r_1)} \dots \xrightarrow{(a_n, r_n)} P'$, P' will be called as the derivation of P , and the collection $d_s(P)$ will be the collection of all derivations of P . The state space X_s is the collection of all nodes of the derivative graph of SM-PEPA, and SMP corresponding to SM-PEPA is built: $\{X, T\} = \{X_n, T_n, n = 0, 1, 2, \dots\}$, where, $X_n \in X_s$, and when $m = n = l = 1$, we get

Attack', Failure', Accident', General' are derivations of component Attack, Failure, Accident, General, respectively. $Q(t)$ of SMP satisfies

$$Q_{ij}(t) = P\{X_{n+1} = j, T_{n+1} - T_n \leq t \mid X_n = i\} = P_{ij}H_{ij}, \quad (1)$$

$p_{ij} = P\{X_{n+1} = j, X_n = i\}$ represents the state transition rate between i, j ; $H_{ij} = P\{T_{n+1} - T_n \leq t \mid j = X_{n+1}, i = X_n\}$ represents the distribution probability obeyed by action change rates between i and j .

The stable-state probability of Markov can be obtained after the following calculations [23]:

$$\pi_i = \frac{v_i E[T_i]}{\sum_j v_j E[T_j]} = i, j \in X_S. \quad (2)$$

Let X_S be any state space and let the corresponding Markov Chain, $P = (p_{ij})$ be a state transition matrix:

$$p_{ij} = \lim_{t \rightarrow +\infty} Q_{ij}(t). \quad (3)$$

After doing reduction of the model, when the delay of action obeys the exponential distribution, the probability of transition from state α to state l is $p_{\alpha l} = r_{\alpha l} / \sum_j r_{\alpha j}$, where $r_{\alpha j}$ is the delay parameter of actions. And when the delay time parameter obeys the general distribution of action d , because its priority is higher than other actions, the probability of transition to the determined state q is 1, and the probability of transition to the rest is 0.

Therefore, the steady-state rate of embedded semi-Markov Chain satisfies

$$\begin{cases} \vec{v} = \vec{v}P, \\ \sum_i v_i = 1, \end{cases} \quad (4)$$

$\vec{v} = (V_G, V_C, V_D, V_{SH}, V_{SD})$ is a stationary probability vector embedded in semi-Markov Chain.

When the duration of behaviors in SM-PEPA model obeys exponential distribution, the solution of the model can be transformed into solving the duration Markov Chain corresponding to PEPA. Assuming that the steady-state probability distribution of duration Markov chains is $\pi(\cdot)$, so

$$\begin{cases} \pi Q = 0, \\ \sum_{i=1}^n \pi_i = 1, \end{cases} \quad (5)$$

$\pi = \{\pi_1, \pi_2, \dots\}$ is the steady-state probability vector.

4.1.1. State Transition Matrix. Because a survivable system application scenario for the corresponding goal is different, its internal and external environment are also different; at the same time, it is limited by many constraints, etc., so according to different application scenario for the conditions for survival systems can be divided into five states: normal survival state (general), compromise survival state (compromised), cognitive detection state (detection), the recovery state (selfhealing), and self-destructive state

(selfdestruction). From the state set, the state space $X = \{G, V, D, SH, SD\}$ can be obtained, and then the DTMC chain, just an example, can be obtained, as shown in Figure 7.

The above-mentioned parameters' probability values are shown in Table 2.

$$P = \begin{matrix} & \begin{matrix} G & C & D & SH & SD \end{matrix} \\ \begin{matrix} G \\ C \\ D \\ SH \\ SD \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ p1 & 0 & \overline{p1} & 0 & 0 \\ 0 & 0 & 0 & p2 & \overline{p2} \\ VG & 0 & 0 & p3 & p4 \\ p5 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad (6)$$

4.1.2. Quantification of Evaluation Indicators. Based on the state transition matrix P , the corresponding relationship between the evaluation index and the state transition probability is established [15]:

Recognition: $p1, TC \rightarrow G$

Resistance: $p1 + (1-p1) p2, TC \rightarrow G, TC \rightarrow D$

Recovery: $1-p3-p4, TSH \rightarrow G$

Reliability: $1-\pi SD$

Among them, $TC \rightarrow G$ means the time interval between threat detection and threat processing; $TC \rightarrow D$ means the time interval of resisting invasion or attack; $TSH \rightarrow G$ is the time interval of system self-recover; πSD is the steady-state probability of system in self-destructive state; $T_C \rightarrow G, T_C \rightarrow D, T_{SH} \rightarrow G$ can be obtained from the actual operation of survivable systems through bypass network monitoring tools.

4.1.3. Solution of Approximate Steady-State Probability. According to the steady-state distribution value of the steady-state rate v_i embedded in semi-Markov Chain, the five calculating formulas of steady states are as follows: $V_G + V_C + V_D + V_{SH} + V_{SD} = 1$

$$\begin{aligned} V_G &= \frac{1 - p_3}{2(1 - p_3)(2 - p_1) + p_2(1 - p_1)(p_3 + p_4)}, \\ V_C &= \frac{1 - p_3}{2(1 - p_3)(2 - p_1) + p_2(1 - p_1)(p_3 + p_4)}, \\ V_D &= \frac{(1 - p_1)(1 - p_3)}{2(1 - p_3)(2 - p_1) + p_2(1 - p_1)(p_3 + p_4)}, \\ V_{SH} &= \frac{p_2(1 - p_1)}{2(1 - p_3)(2 - p_1) + p_2(1 - p_1)(p_3 + p_4)}, \\ V_{SD} &= \frac{(1 - p_1)(1 - p_2 - p_3 + p_2 p_3 + p_2 p_4)}{2(1 - p_3)(2 - p_1) + p_2(1 - p_1)(p_3 + p_4)}. \end{aligned} \quad (7)$$

Here, we make the average staying time of self-destructive SD obey subexponential distribution, distribution parameters λ_1 and λ_2 , while the average staying time of other states obeys exponential distribution, which is also consistent with the actual network situation, then the average staying time of five states is shown as formula (7).

FIGURE 7: DTMC corresponding to survival situations.

TABLE 2: Parameters and probability values.

$p1 = W2$	$1-p1 = W1$
$p2 = L2$	$1-p2 = L1$
$p3 = L5$	$p4 = L4$
$1-p3-p4 = L3$	$p5 = L6$

The state transition matrix P.

$$\begin{aligned}
 t_G &= \frac{1}{\lambda_G}, \\
 t_C &= \frac{1}{\lambda_C}, \\
 t_D &= \frac{1}{\lambda_D}, \\
 t_{SH} &= \frac{1}{\lambda_{SH}}, \\
 t_{SD} &= \frac{1}{\lambda_1} + \frac{1}{\lambda_2}.
 \end{aligned} \tag{8}$$

The formula to get the steady-state probability based on the semi-Markov process is

$$\pi_i = \frac{v_i t_i}{\sum_j v_j t_j} \quad i, j \in \{G, C, D, SH, SD\}. \tag{9}$$

The steady-state probability of semi-Markov process can be solved finally. To simplify the analysis process, a global cognitive unit is assumed to consist of two domain cognitive units, Domain_1 and Domain_2. The approximate steady-state probabilities derived from each cognitive unit are shown in Table 3.

4.2. Quantitative Analysis and Simulation. In this paper, PEPA Workbench is used to process data files, and the tool, Version v25 of the PEPA Eclipse Plugin of the Computer Science Foundation Laboratory of Edinburgh University, is adopted to quantitatively analyze the performance of the proposed cognitive model in terms of resistance, recognition, and recovery.

TABLE 3: The approximate steady-state probabilities.

Module type	Module name	Approximate steady-state probabilities
Global cognition	Monitor	0.35971
	Decide	0.03452
	Execute	0.46896
	Monitor_1	0.31506
	Decide_1	0.03447
Domain cognition	Execute_1	0.03569
	Monitor_2	0.31958
	Decide_2	0.03452
	Execute_2	0.03689

Due to the addition of cognitive computing features in the model, state space X_S can be further divided into collection X_1 and X_2 to represent cognitive and noncognitive survivable state collections. Each local derivation in X_2 contains noncognitive survivable state and indefinite state in the following form: $X_1 = \{x|x = \text{DeGradation}||\dots\}$. Similarly, the steady-state probability collection, $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$, can also be divided into two parts, corresponding to the subcollection C_D in X_1 and the subcollection C_{UD} in X_2 , respectively.

The test parameters are listed in Table 4.

In order to better measure the impact of the selected index parameters on the cognitive performance of survivable systems, the resistance parameter h and the cognitive parameter z_1 are first examined. And then the values of h and z_1 are adjusted to maintain the rest of the parameters unchanged. The experimental results are shown in Figures 8 and 9.

In Figure 8, parameter h means the probability of attackers finding system flaws and, correspondingly, means the system's resistance to attacks. The smaller the value of h is, the stronger the anti-attack ability of the system becomes. With h decreasing, the survivability index of the system increases gradually. But the resistance of the system is not endless. When the value of h reaches $1e-09$, the survivability index of the system approaches 1.0 and gradually becomes stable. No matter how strong the attack defense is, it is possible to be invaded. The curve shows the defense trend that it will return to the origin and start a new round of survivability evolution process. As long as new flaws are added to the system and the flaws recognition rate of attackers are increased in unit time, the survival index curve will always show a trend similar to Figure 8.

Figure 9 shows the curve of system survivability index. z_1 represents the probability of attacks being recognized by the system. When the initial recognition rate is close to zero, the survivability index of the system is about 0.08, and the local cognitive units begin to update the acquisition rules independently. With the recognition rate increasing, the system keeps adjusting its state and updates the results of self-feedback behavior transitions to the global cognitive level, and the survivability index gradually increases, which improves the fact that the self-configuration mechanism in the cross-layer cognitive network further strengthens the system's survivability. When z_1 increases to 0.7, the

TABLE 4: Test parameters.

Parameters	Values
P	0.6000
k	100.0
g	0.4000
Z_2	0.2460
Z_3	0.0500
W_1	0.9100
W_2	0.0900
L_1	0.2700
P_1	0.0015
L_2	0.7300
H_1	0.0140
S_1	0.0100
P_2	0.1500
S_2	0.2600
L_6	0.9500
S_3	0.0152

FIGURE 8: The impact of (h) on survivable parameters.

FIGURE 9: z_1 's effects on survivability indexes.

survivability index begins to climb rapidly, which shows that improving the system's attack recognition rate delivers better effects on enhancing the system's survivability, rather than strengthening its resistance.

From the DTMC corresponding to the cognitive survival state collections, we can see that there are three possible states of self-recovery actions, L_3 , L_4 and L_5 as assumed. And the self-recovery rate $V = L_3/L_3 + L_4 + L_5$ in Figure 10 shows the changes of system survivability indexes when the self-recovery rates are 0.532, 0.758, 0.914, and 0.997, respectively. It also unveils the fact that, with the increase of the interval time of self-recoveries, the survivability index curve declines steadily. When the intervals are the same, the larger the value of self-recovery rate V is, the higher the survivability index of the system becomes. When the value of V is 0.997, the survivability index is close to the highest, 1.0, the system performs the best self-recovery ability. It can be seen that improving the system recovery is one of the most feasible ways to improve the system survivability.

For survivable systems, different indicators affecting cognitive performance are tested. The main parameters and their implications are shown in Table 5. In view of the cognitive model in this paper, the relationship between the above parameters and the cognitive ability of survivable system is analyzed and tested accordingly.

Reliability is one of the important indicators affecting the cognitive ability of survivable systems. Failure of cognitive units has great impacts on the cognitive performance of systems. The relationship between E_{SD} and reliability is shown in Figure 11. Parameters of E_{SD} decline along the transverse axis, and the height of the histogram decreases as well, which proves that the reliability of the system gets weakened as the interval of failure time decreases; that is, the higher the failure frequency is, the weaker the reliability becomes. When E_{SD} is $1/50 \times E_{SD}$, the reliability is still above 0.9, while when E_{SD} is reduced to $1/100 \times E_{SD}$, the reliability drops sharply to less than 0.1. That is, because the number of cognitive units that provide normal service decreases with the increase of failure frequency, the reliability of the system is weakened dramatically, thus causing significant impacts on the system's cognitive ability.

Recovery is an important indicator to measure the system's cognitive ability. Figure 12 demonstrates the relationship between E_{SH} and recovery. The system's recovery falls with E_{SH} growing, which shows that the longer the recovery time is, the more poor the recovery performance will be. In particular, when the E_{SH} value is $100 \times E_{SH}$, the system's recovery decreases to about 0.2. The survivable system cannot avoid attacks, faults, or other accidents under such complex working environment. If the self-recovery time is too long, the duration of staying in unsafe states will be longer, thus affecting the cognitive survivable system's cognitive ability.

The relationship between the rate of monitor behaviors' transitions (m represents different rates) and recognition is shown in Figure 13. From the figure, we can see that every curve climbs upwards, demonstrating that the system's recognition gets stronger as t increases. At first, the four curves rise significantly and then tend to grow steadily and slowly. That is because the time t starts to advance from 0,

FIGURE 10: Recovery rate's effects on survivability indexes.

TABLE 5: Parameters and implications of cognitive performance indicators.

Parameter	Value
E_{SD}	Expected value of interval time between system failures
E_{SH}	Expected time for system self-recovery
M	Rate of monitoring behaviors' transitions
E	Rate of executing behaviors' transitions

FIGURE 11: E_{SD} 's impacts on reliability.

meaning that the system begins to work from nonworking states. Then, the system's recognition increases rapidly from 0. And when t advances to a certain value, the recognition ability will also remain at a stable state. When m is 1.0, the curve of recognition stays at the lowest level, while when m is 5.0, the curve is at the highest level, which shows that the bigger the m value is, or the faster the execution rate of transition behaviors is, the stronger the recognition of the system will be. Because the time delay of executing

FIGURE 12: E_{SH} 's effects on recovery. FIGURE 13: m 's effects on the system's recognition.

monitoring behavior decreases, the number of monitoring units in working states increases, which improves the efficiency of perception and detection of the internal and external environment of the system, so the system's cognitive ability gets stronger.

The transition rate of monitoring behaviors, namely, the relationship between e and recognition, is shown in Figure 14. We can see that every curve climbs upwards along the transverse axis, demonstrating that the system's recognition gets stronger as t increases. When the value of t is relatively small, the four curves rise rapidly and then tend to grow steadily and slowly; that is because the time t starts to advance from 0, meaning that the system begins to work from nonworking states. Then, the system's recognition increases rapidly from 0. And when t advances to a certain value, the recognition ability will also remain at a stable state. The four curves are obtained when e is 0.2, 0.4, 1.5, and 2.0, respectively. When $e=0.2$, the corresponding curve is at the

FIGURE 14: e 's effects on the system's recognition.

lowest level, and when $e = 2.0$, the corresponding curve is at the highest level, which means that the bigger the value of e is, the stronger the system's recognition ability becomes. Because the time delay of executing monitoring behavior decreases, the number of monitoring units in working states increases, which improves the efficiency of perception and detection of the internal and external environment of the system, so the system's cognitive ability gets stronger.

5. Conclusion

Cognitive model of survivable system is the abstraction of cognitive ability of survivable system and the key to enhance the system's cognitive ability.

This paper studies the autonomous cognitive model and analysis method of survivable systems. The self-feedback structure of cognitive unit is improved, and the formal modeling of cognitive process is carried out by describing the transition map of cognitive survival state. In addition, the paper has obtained standardized results with the application of PEPA Workbench model tool. Next, we will further improve the cognitive structure and formal model of survivable systems and conduct research on the enhanced design of survivable system with autonomous cognitive model.

Data Availability

The data set can be obtained free of charge from <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This present research work was supported by the National Natural Science Foundation of China (Nos. 61202458 and

61403109); the Natural Science Foundation of Heilongjiang Province of China (No. F2017021); the Harbin Science and Technology Innovation Research Funds (No. 2016RAQXJ036).

References

- [1] V. R. Westmark, "A definition for information system survivability," in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pp. 2086–2096, IEEE Computer Society Press, Big Island, HI, USA, January 2004.
- [2] R. J. Ellison, R. C. Linger, T. Longstaff, and N. R. Mead, "Survivable network system analysis: a case study," *IEEE Software*, vol. 16, no. 4, pp. 70–77, 1999.
- [3] G. S. Zhao, H. Q. Wang, and J. Wang, "Study on situation evaluation for network survivability based on grey relation analysis," *Mini-Micro Systems*, vol. 27, no. 10, pp. 1861–1864, 2006.
- [4] J. Wang, H. Q. Wang, and G. S. Zhao, "Situation tracking assessment for network survivability based on sequential Monte Carlo," *Journal of Harbin Institute of Technology*, vol. 40, no. 5, pp. 802–806, 2008.
- [5] J. Wang, H. Q. Wang, and G. S. Zhao, "A situation assessment method for network survivability," *Wuhan University Journal of Natural Sciences*, vol. 11, no. 6, pp. 1785–1788, 2006.
- [6] J. Wang, H.-Q. Wang, and G.-S. Zhao, "Formal modeling and quantitative evaluation for information system survivability based on PEPA," *The Journal of China Universities of Posts and Telecommunications*, vol. 15, no. 2, pp. 88–113, 2008.
- [7] J. Wang, H. Q. Wang, and G. S. Zhao, "Automated analysis and validation for survivability of distributed mission-critical systems," *Chinese High Technology Letters*, vol. 19, no. 6, pp. 572–579, 2009.
- [8] G. S. Zhao, H. Q. Wang, and J. Wang, "A novel formal analysis method of network survivability based on stochastic process algebra," *Tsinghua Science & Technology*, vol. 12, no. 1, pp. 175–179, 2007.
- [9] J. A. Zinky, D. E. Bakken, and R. E. Schantz, "Architectural support for quality of service for CORBA objects," *Theory and Practice of Object Systems*, vol. 3, no. 1, pp. 55–73, 1997.
- [10] G. S. Zhao, J. Wang, and Z. X. Li, "A method of autonomous emergency rejuvenation for survivable system," *Mini-Micro Systems*, vol. 35, no. 10, pp. 2284–2289, 2014.
- [11] Z. H. Yu, S. M. Lin, and H. Q. Chen, *Network Security-Research on Survivability and Network Modeling*, Beijing, China, 2012.
- [12] Report on Software Survivability[EB/OL]. <http://www.doc88.com/p-147660727345.html>. Beijing, 2012.
- [13] G. S. Zhao, W. D. Wang, and W. Zhang, "Research of cloud survivability," *Telecommunications Science*, vol. 9, pp. 52–59, 2011.
- [14] G. S. Zhao, H. L. Liu, and J. Wang, "Study on the autonomous recognition mechanism for survivable systems," *Chinese High Technology Letters*, vol. 24, no. 10, pp. 999–1006, 2014.
- [15] J. Wang and G. S. Zhao, "Cognitive model and quantitative analysis for survivable system based on SM-PEPA," *Journal of Huazhong University of Science and Technology (Nature Science Edition)*, vol. 43, no. 5, pp. 99–103, 2015.
- [16] D. S. Modha, R. Ananthanarayanan, S. K. Esser, A. Ndirango, A. J. Sherbondy, and R. Singh, "Cognitive computing," *Communications of the ACM*, vol. 54, no. 8, pp. 62–71, 2011.
- [17] J. T. Bradley, "Semi-Markov PEPA: modelling with generally distributed actions," *International Journal of Simulation*, vol. 6, no. 3-4, pp. 43–51, 2005.

- [18] H.-Q. Wang, H.-W. Lü, Q. Zhao, X.-K. Dong, and G.-S. Feng, "Model and quantification of autonomic dependability of mission-critical systems," *Journal of Software*, vol. 21, no. 2, pp. 344–358, 2010.
- [19] R. Thomas, D. Friend, L. Dasilva, and A. Mackenzie, "Cognitive networks: adaptation and learning to achieve end-to-end performance objectives," *IEEE Communications Magazine*, vol. 44, no. 12, pp. 51–57, 2006.
- [20] P. Balamuralidhar and R. Prasad, "A context driven architecture for cognitive radio nodes," *Wireless Personal Communications*, vol. 45, no. 3, pp. 423–434, 2008.
- [21] C. Fortuna and M. Mohorcic, "Trends in the development of communication networks: cognitive networks," *Computer Networks*, vol. 53, no. 9, pp. 1354–1376, 2009.
- [22] Laboratory For Foundations of Computer Science: Version v25 of the PEPA Eclipse Plugin[EB/OL].<http://www.dcs.ed.ac.uk/pepa/downloads>.
- [23] J. D. Nicholas, *Parallel Computation of Response Time Densities and Quantiles in Large Markov and Semi-markov models*, p. 10, Imperial College, London, UK, 2004.