Malicious Code Propagation Model Based on the Alert Mechanism in Blockchain Network

Ruiqing Lu, Jianguo Ren*, Yonghong Xu

***Abstract*—The blockchain network has core advantages such as decentralization and is becoming a key infrastructure in the digital economy era. However, its open and shared nature makes it more vulnerable to malicious code attacks than other networks. This study provides a detailed analysis of the propagation mechanism of malicious code in blockchain networks. Considering the latent malicious code and the characteristics of real-time sharing of information transmission in the blockchain network, using the prevention and control information after the recovery of infected nodes, based on the traditional Susceptible-Exposed-Infected-Recovered (SEIR) model, a Susceptible-Exposed-Infected-Alert-Recovered (SEIRA) model with alert mechanism is proposed. Mathematical methods are employed to conduct stability analysis on the new model and verify its stability. Numerical simulation experiments demonstrate that, compared to the SEIR model, when the transmission threshold is less than 1, the biggest number of infected nodes drops by 4.69%, 36.94%, and 43.57%, respectively. The experimental results indicate that the SEIAR model can better contain malicious code with higher alert rates leading to better effects. This provides new theoretical and practical guidance for formulating blockchain security protection strategies.**

***Index Terms*—Blockchain network, Malicious code, Propagation model, Stability analysis**

I. Introduction

With the rapid development of information technology in recent years, cyberspace security is facing unprecedented challenges [1]. Malicious code is a major danger to network security. It has infectious, concealed, and latent features, making it difficult to identify and track on the network, significantly complicating network security defense. After infecting a computer system, latent malicious code might remain concealed in the network environment for a long period, only to be activated under specific conditions. As a result, detecting and removing latent malicious code in a timely manner has become a significant difficulty in network security [2].

Manuscript received September 7, 2024; received March 14, 2025.

This work is supported by the Jiangsu Normal University Graduate Research and Practice Innovation Program (2024XKT2617) and the National Science Foundation of Jiangsu Province under Grant (BK20241960).

Ruiqing Lu is a postgraduate student of the College of Computer Science, Jiangsu Normal University, Xuzhou, China (e-mail: lrq9809@163.com).

Jianguo Ren is an associate professor of the Research Center for Complex Networks and Swarm Intelligence, Jiangsu Normal University, Xuzhou, China (corresponding author to provide phone: +8613775848013; e-mail: jsnucs1119@163.com).

Yonghong Xu is a teacher of the School of Life Science, Jiangsu Normal University, Xuzhou, China (e-mail: xyh8810@126.com).

Blockchain, a decentralized and tamper-proof distributed ledger technology, has gained a lot of attention in the field of network security because of its encryption properties and decentralized process [3], [4]. Participants in the blockchain network communicate and trade using peer-to-peer (P2P) technology. This method not only improves data transmission efficiency, but also increases network reliability. Although the blockchain network offers several security advantages over traditional networks, its anonymity and decentralization allow malicious code to proliferate more easily. Malicious behaviors, such as malware and topological structural attacks, continue to have a significant impact on the security of blockchain networks [5]. The blockchain network architecture is shown in Figure 1.

The anonymity and real-time nature of blockchain networks provide ease and security to network participants, but they also provide circumstances for the rapid spread of malicious code, exposing several security problems [6]. The anonymity of blockchains may be used to conceal harmful behavior, making malicious code in the network more difficult to detect [7]. As a result, the prevention and control of malicious code cannot be based simply on detection technology. It is also critical to investigate and comprehend the spreading process of malicious programs [8]. Establishing a harmful code propagation model to investigate the propagation dynamics of malicious code, as well as more accurately simulate and anticipate its proliferation trend, is critical for developing positive and effective preventive measures and emergency response plans [9].

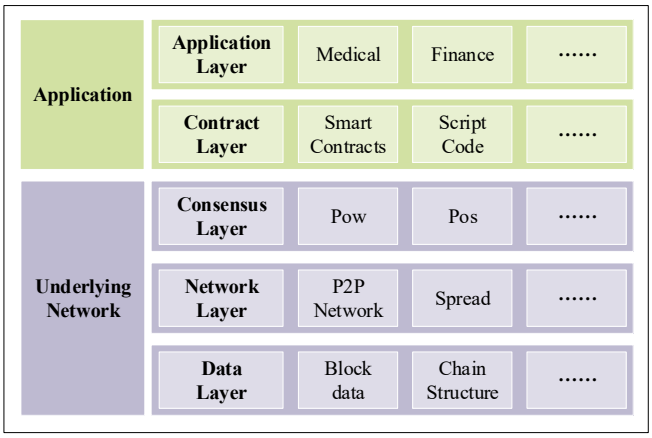

Fig. 1. Blockchain Network Architecture

II. Relate Work

Many scholars have done in-depth investigations of the transmission of harmful code. Kephart et al. [10] created the infectious illness model to explore the transmission of

harmful code in the network and proposed the first network virus compartment model SI. Mishra et al. [11] developed the SSSIR model to study the spread dynamics of three forms of bad behaviors in the network: viruses, Trojan horses, and worms, and investigated the role of antivirus software in preventing the distribution of dangerous code through experiments. The investigation solely looked at the influence of antivirus software and neglected the involvement of network nodes in network alerts and defenses. Toutonji et al. [12] proposed the VEISV model for studying the behavior of several malware worms. This model takes into account the influence of external defense tactics and changes in node status, but it does not thoroughly investigate the significance of internal network techniques in preventing the propagation of harmful code.

The dissemination behavior of malicious code is strongly influenced by network topology. As a result, when researching its dissemination characteristics, the occurrence of malicious code should be studied across various network architectures. Li et al. [13] studied the propagation mechanism of malicious code in software-defined networks, established a W-SIR model based on a feedback mechanism, and designed three modules to inhibit the spread of malicious code. The experimental results show that the module can effectively contain the spread of the virus at the technical level, showing high stability and application potential. However, it does not fully consider how to adjust security protection policies in the dynamic changing network environment. Ding et al. [14] investigated the propagation characteristics of worms in the mobile Internet and suggested the SEIQR model with time delay, which demonstrated the effect of the time delay when each state node changes to the recovery state on worm propagation. The study, however, only examined the influence of external strategy duration and did not go into detail on malicious code prevention techniques. Tang et al. [15] established the SLBRS model based on the characteristics of complex dynamic networks, investigated the changes in vulnerable and infected nodes, and calculated the security entropy derivative to examine the changing trend of network space security. However, this model merely evaluates the influence of node relationships on security trends and does not provide specific suppression tactics. Zhang et al. [16] proposed the SEIQRS-V model by combining wireless sensor network characteristics. This model takes into account the network's realistic features and incorporates the idea of latent latency to investigate the transmission of latent malicious code. The article just gives a basic prevention and control technique and does not go into detail about how to stop the spread of dangerous code. Liu et al. [17] suggested a SILRD model using wireless sensor networks. This model describes the node energy level and provides defense by refilling node energy and repairing vulnerabilities. However, this model transmits vulnerability patches via drones, which adds an undue burden on the working environment and costs.

The preceding research modeled the spread of malicious code in a specific network, but they do not applicable to the propagation dynamics of dangerous code in a blockchain network, and none of them addressed exploiting network participants' relationships to limit virus transmission. Nodes in a network typically utilize security measures such as antivirus software to cope with known threats. These technologies can detect internal and external threats by comparing virus databases and analyzing harmful network traffic, however they may not be very effective when dealing with unknown threats. When an infected node successfully removes malicious code using its anti-virus capability, it generates malicious code prevention information in real time and sends alert signals to adjacent susceptible nodes, quickly spreading the malicious code prevention strategy to the entire network in a one-to-ten, ten-to-hundred manner and enhancing the network's defense capability.

To investigate the spread of harmful code in the blockchain network, it is required to evaluate both the features of the blockchain network and those of bad code. Based on the classic SEIR paradigm, this work proposes the SEIAR model, which includes a dangerous code alert system. Alert nodes are special nodes that may create and transmit alert signals. Such nodes take advantage of the blockchain network's information-sharing capabilities to quickly broadcast prevention and control information to neighboring nodes in the network after recognizing and eliminating harmful code, hence assisting in gaining immunity against malicious code. Establishing an alert system can not only improve the node's security protection level, but it can also actively aid other nodes in increasing their malicious code defense capabilities, so improving network security overall. The positive impact of the alert mechanism allows the blockchain network to respond more effectively to the invasion of harmful code while also providing a new perspective for developing more extensive and effective malicious code prevention and control measures.

The main work of this article is:

1) Analyze the method by which latent harmful code propagates in the blockchain network and propose an SEIAR model with an alert mechanism. This model can accurately depict the spreading dynamics of latent malicious code in the blockchain network.

2) Convert the model into a system of dynamic equations, then determine the system's equilibrium point and propagation threshold. We use mathematical approaches, such as the Jacobian matrix and the Lyapunov function construction, to demonstrate the model's disease-free and viral equilibrium points are stable. In addition, we conduct a sensitivity analysis of the system's fundamental reproduction number for the relevant parameters.

3) This study uses numerical simulation experiments to explore the effect of alert methods and critical factors in the model on the number of infected nodes in the network. And suggest defense plans and methods that are suited to certain scenarios in order to effectively meet these issues. Experimental results suggest that the alert system has a considerable impact on preventing the propagation of harmful code, serving as an important reference for developing successful security solutions.

III. Model Formulate

Based on the SEIR model, this paper proposes a SEIAR model with an alert mechanism by combining the characteristics of malicious code and blockchain networks. The nodes in the model are defined as five states: susceptible, exposed, infected, alert, and recovered.

S denotes susceptible nodes, which are not infected with the virus at time t and have limited immunity to unknown malicious code in the model.

E denotes exposed nodes, which are infected with latent malicious code but do not show evidence of infection at time t. These nodes are not infectious.

I indicate infected nodes, which are nodes that have been infected with the virus and turned on at time t. These nodes are infectious.

A denotes alert nodes, which are nodes that created a defense at time t in the model. The infected nodes have removed the harmful malware using virus detection. The nodes share preventive and control information across the network to assist other nodes in developing defense capabilities against dangerous code.

R symbolizes recovered nodes, which are already immune to the virus at time t.

The transformation relationship between the various states of the SEIAR model is shown in Figure 2.

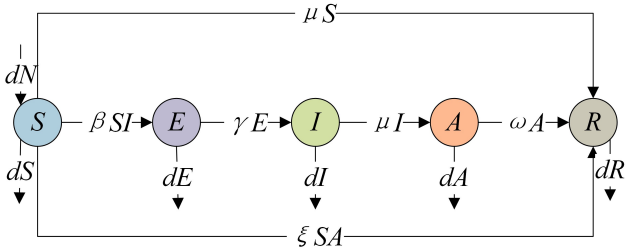

Fig. 2 State transformation relationship of SEIAR model

In the model, d indicates the probability of a new node joining the current network and of an existing node leaving the network per unit of time. N is the total number of nodes in the model $N(t) = S(t) + E(t) + I(t) + A(t) + R(t)$. When the blockchain network is invaded by malicious code, the node status will be converted according to the following rules:

1) Susceptible state S → Exposed state E. After infiltrating the nodes, the malicious malware will remain latent in the computer system. The malicious code is ready to be executed, and the nodes transition to the exposed state. The infection rate β is the likelihood of a vulnerable node becoming infected per unit time.

2) Exposed state E → Infected state I. When the latent harmful code's triggering circumstances are met, the malicious code in the nodes is activated, and the nodes become infected. The activation rate γ measures the likelihood of activating the latent node per unit time.

3) Infected state I → Alert state A. Because of the nodes' anti-virus capabilities, affected nodes have a chance to remove the invading bad code on their own. Afterward, the infected nodes will record the source, characteristics, and removal procedures of the malicious code in depth. After removing the harmful code, the nodes enter the alert state. The immunity rate μ indicates the likelihood that the infected node's harmful code will be cleansed per unit time.

4) Alert state A → Recovered state R. The alert nodes construct an alert signal based on the malicious code's characteristics, broadcast it to the susceptible nodes, and wait for a response. After receiving the information, the susceptible nodes act fast to construct defenses and gain immunity. When the alert nodes deliver the alert information to the vulnerable nodes and receive a response from the nodes and the administrator within T (the time it takes for the alert nodes to send the alarm signal for the first time), the nodes become immune. If the alert nodes do not receive a response from the susceptible nodes within T, it will transmit the alert information to the susceptible nodes again before directly transforming into the immune state. The probability of alert node conversion to the immune node in unit time is ω.

5) Susceptible state S → Immune state R. Because the nodes themselves have anti-virus capabilities, susceptible nodes have a chance of acquiring the capacity to defend against harmful code, and the nodes are immediately transformed to immune nodes. The immunity rate μ represents the unit time probability. Furthermore, the vulnerable nodes can get the malicious code prevention and control strategy by receiving the alert signal from the alert nodes, allowing them to develop a defense mechanism and transform themselves into immune nodes. The alert rate ξ is the likelihood of a node converting into a recovery node per unit of time.

According to the network node state transition model in Figure 2, it is transformed into the following differential dynamics equations:

$$\begin{cases} \dfrac{dS}{dt} = dN - dS - \mu S - \xi SA - \beta SI \\ \dfrac{dE}{dt} = \beta SI - \gamma E - dE \\ \dfrac{dI}{dt} = \gamma E - \mu I - dI \\ \dfrac{dA}{dt} = \mu I - \omega A - dA \\ \dfrac{dR}{dt} = \mu S + \xi SA + \omega A - dR \end{cases} \tag{1}$$

The first four equations in system (1) are independent of the fifth equation, hence system (1) can be represented as the following system of equations:

$$\begin{cases} \dfrac{dS}{dt} = dN - dS - \mu S - \xi SA - \beta SI \\ \dfrac{dE}{dt} = \beta SI - \gamma E - dE \\ \dfrac{dI}{dt} = \gamma E - \mu I - dI \\ \dfrac{dA}{dt} = \mu I - \omega A - dA \end{cases} \tag{2}$$

The feasible domain of system (2) is $\Omega = \left\{ (S, E, I, A) \in R_+^3 : 0 \le S + E + I + A \le N \right\}$.

The SEIAR model's essential component is the alert mechanism, which is divided into two parts. The first part identifies and detects malicious code in the system, cleans up the detected malicious code, and collects its characteristic information. The algorithm is shown in Table I.

The second part is based on the first part. When the malicious code of the infected nodes is cleared, the infected node is transformed into an alert node, and then the alert signal is generated and broadcast to the entire network. The algorithm is shown in Table II.

TABLE I
MALICIOUS CODE CLEANUP PROCESS

```
for each I in N:
  Update virus database
  Scan and detect malicious code
  if malicious code is detected
    Take security policy
    if There is a corresponding security policy
      Clean up malicious code
    else
      Generate a new security policy
end for
```

TABLE II
ALERT INFORMATION SENDING PROCESS

```
for each A in N
   Extract malicious code feature information
   Generate alert information
   Broadcast alert information to S
   Wait for a response, the waiting time is T
   if All nodes respond
      A→R
   else
      Broadcast alert information to unresponsive nodes
      A→R
end for
```

IV. STABILITY ANALYSIS

To analyze the model's stability, we first determine its equilibrium point. This system of formulae determines the model's equilibrium point:

$$\begin{cases} \dfrac{dS}{dt}=0 \\ \dfrac{dE}{dt}=0 \\ \dfrac{dI}{dt}=0 \\ \dfrac{dA}{dt}=0 \end{cases} \tag{3}$$

Calculating formula (3) yields the system's unique disease-free equilibrium point:

$$Q_0=(S_0,E_0,I_0,A_0)=\left(\frac{dN}{d+u},0,0,0\right)$$

The unique virus equilibrium:

$$Q^*=(S^*,E^*,I^*,A^*)=\left(\frac{(d+\mu)(d+\gamma)}{\beta\gamma},\frac{I^*(d+\mu)}{\gamma},I^*,\frac{I^*\mu}{d+\omega}\right)$$

Among them:

$$I^*=\frac{(\mathrm{d}+\omega)\left(d^3+2d^2\mu+d^2\gamma+d\mu^2+2d\mu\gamma+\mu^2\gamma-\beta dN\gamma\right)}{(d+\gamma)(d+\mu)(\beta d+\beta\omega+\xi\mu)}$$

System (2)'s basic reproduction number is derived through the next-generation matrix methods [18], [19]. Let x = (S, E, I, A)$^{\mathrm{T}}$, System (2) can be expressed as $\frac{\mathrm{dx}}{dt}=F(x)-V(x)$. Among them:

$$F(\mathrm{x})=\begin{pmatrix}0\\ \beta SI\\ 0\\ 0\end{pmatrix}$$

$$V(\mathrm{x})=\begin{pmatrix}-dN+dS+\beta SI+\mu S+\xi SA\\ dE+\gamma E\\ \mu I+dI-\gamma E\\ -\mu I+dA+\omega A\end{pmatrix}$$

Find the Jacobian matrix for F(x) and V(x), and we get:

$$DF(Q_0)=\begin{pmatrix}0&0&0&0\\0&0&\beta S&0\\0&0&0&0\\0&0&0&0\end{pmatrix}$$

$$DV(Q_0)=\begin{pmatrix}\mathrm{d}+\mu&0&\beta S&\xi S\\0&d+\gamma&0&0\\0&-\gamma&d+\mu&0\\0&0&-\mu&d+\omega\end{pmatrix}$$

Then, we have

$$R_0=\frac{\beta S\gamma}{(d+\mu)(\mathrm{d}+\gamma)}=\frac{\beta dN\gamma}{(d+\mu)^2(\mathrm{d}+\gamma)} \tag{4}$$

A. Stability Analysis of Disease-free Equilibrium Point

Theorem 1 *Suppose that $R_0<1$, the disease-free equilibrium point Q_0 of the system (2) is locally asymptotically stable in Ω; $R_0>1$ is unstable.*

Proof The Jacobian matrix of the system (2) at Q_0 is:

$$J(Q_0)=\begin{pmatrix}-d-\mu&0&-\beta S&-\xi S\\0&-d-\gamma&\beta S&0\\0&\gamma&-d-\mu&0\\0&0&\mu&-d-\omega\end{pmatrix} \tag{5}$$

Characteristic root of $J(Q_0)$:

$$\begin{cases}\lambda_1=-d-\mu\\ \lambda_2=-d-\omega\\ \lambda_3=-\frac{1}{2}(2d+\mu+\gamma)-\frac{1}{2}(-2\gamma\mu+\mu^2+\gamma^2+4S\beta\gamma)^{\frac{1}{2}}\\ \lambda_4=-\frac{1}{2}(2d+\mu+\gamma)+\frac{1}{2}(-2\gamma\mu+\mu^2+\gamma^2+4S\beta\gamma)^{\frac{1}{2}}\end{cases} \tag{6}$$

If all the parameters in the model are positive, then λ_1, λ_2, and λ_3 are all less than 0. When $R_0<1$, there is $S\beta\gamma<(d+\mu)(d+\gamma)$, therefore, $\mu^2-2\mu\gamma+\gamma^2+4S\beta\gamma<\mu^2-2\mu\gamma+\gamma^2+4(d+\mu)(d+\gamma)$. also because $\mu^2-2\mu\gamma+\gamma^2+4(d+\mu)(d+\gamma)=(2d+\mu+\gamma)^2$, so $\lambda_4<0$.

According to the stability theory presented in the literature [20], when all four eigenvalues of matrix $J(Q_0)$ are negative, system (2) is stable at Q_0. Specifically, if $R_0<1$, it follows that all four eigenvalues of matrix $J(Q_0)$ are negative, indicating that within the feasible region, system (2) exhibits local asymptotically stable at Q_0. If $R_0>1$, then $\lambda_4>0$, there is at least one positive eigenvalue in the matrix $J(Q_0)$, which indicates that on the feasible region, the system (2) is unstable at Q_0.

Theorem 2 *Suppose that $R_0<1$, the disease-free equilibrium point Q_0 of the system (2) is globally asymptotically stable in Ω; $R_0>1$ is unstable.*

Proof Construct the Lyapunov function as follows:

$$V_1=E+\frac{d+\gamma}{\gamma}I \tag{7}$$

Derivative:

$$V_1^{'} = \beta SI - (\gamma + d)E + \frac{d+\gamma}{\gamma}(\gamma E - (d+\gamma)I)$$
$$= \beta SI \frac{(d+\gamma)(d+\mu)\gamma}{(d+\gamma)(d+\mu)\gamma} - \frac{(d+\gamma)(d+\mu)}{\gamma} I$$
$$= \frac{(d+\gamma)(d+\mu)I}{\gamma}(R_0 - 1)$$

Consequently, when $R_0 < 1$, we have $V_1' < 0$. By applying the Lasalle's invariance principle [21], Theorem 2 can be established. Theorem 1 and 2 show that when $R_0 < 1$, the system (2) eventually stabilizes at Q_0.

B. Stability Analysis of the Virus Equilibrium Point

Theorem 3 *Suppose that $R_0 > 1$, the viral equilibrium point Q^* of the system* (2) *is locally asymptotically stable in Ω.*

Proof The Jacobian matrix of system (2) at Q^* is:

$$J(Q^*) = \begin{pmatrix} -d-\mu-I\beta-A\xi & 0 & -S\beta & -S\xi \\ I\beta & -d-\gamma & S\beta & 0 \\ 0 & \gamma & -d-\mu & 0 \\ 0 & 0 & \mu & -d-\omega \end{pmatrix} \quad (8)$$

The characteristic polynomial of formula (8) is as follows:

$$\lambda^4 + C_3\lambda^3 + C_2\lambda^2 + C_1\lambda + C_0 = 0 \quad (9)$$

Among them,

$$C_3 = 4d + \gamma + 2\mu + \omega + \beta I + \xi A > 0$$
$$C_2 = \mathrm{d}(5\mathrm{d} + 2\gamma) + \mu(5\mathrm{d} + \mu + \gamma) + \omega(3d + 2\mu + \gamma)$$
$$+ (\beta I + A\xi)(3\mathrm{d} + \gamma + \mu + \omega) > 0$$
$$C_1 = (d+\mu)(d+\omega)(2d+\gamma+\mu)$$
$$+ \beta I(3d^2 + 2d\gamma + 2d\mu + 2d\omega + \gamma\mu + \gamma\omega + \mu\omega)$$
$$+ A\xi(d+\omega)(2d+\gamma+\mu) > 0$$
$$C_0 = I(\mathrm{d}+\gamma)(d+\mu)(\beta d + \beta\omega + \mu\xi) > 0$$

By calculation:

$$H_1 = C_3 > 0$$
$$H_2 = \begin{vmatrix} C_3 & C_1 \\ 1 & C_2 \end{vmatrix} = C_3C_2 - C_1 > 0$$
$$H_3 = \begin{vmatrix} C_3 & C_1 & 0 \\ C & C_2 & C_0 \\ 0 & C_3 & C_1 \end{vmatrix} = C_1H_2 - \mathrm{C}_3^2C_0 > 0$$
$$H_4 = C_0H_3 > 0$$

According to the Routh-Hurwitz stability criterion [22], when $R_0 > 1$, Q^* is locally asymptotically stable in Ω.

Theorem 4 *Suppose that $R_0 > 1$, the virus equilibrium point Q^* of the system* (2) *is globally asymptotically stable in Ω.*

Proof Reference [23] constructs the Lyapunov function. Let,

$$g(x) = x - 1 - \ln x \quad (10)$$

$g(x) \geq 0$ is always true. Then construct the Lyapunov function as follows:

$$V_2 = S^*g(\frac{S}{S^*}) + E^*g(\frac{E}{E^*}) + BI^*g(\frac{I}{I^*}) \quad (11)$$

Where $B = \frac{\beta S^* I^*}{\gamma E^*} > 0$, therefore V_2 is non-negative and is strictly minimized at the unique equilibrium.

Derivative:

$$V_2^{'} = \left(1 - \frac{S^*}{S}\right)(dN - dS - \mu S - \xi SA - \beta SI)$$
$$+ \left(1 - \frac{E^*}{E}\right)(\beta SI - \gamma E - dE)$$
$$+ B\left(1 - \frac{I^*}{I}\right)(\gamma E - \mu I - dI)$$
$$\leq \left(1 - \frac{S^*}{S}\right)((d+\mu)S^* + \beta S^*I^* - (d+\mu)S - \beta SI)$$
$$+ \left(1 - \frac{E^*}{E}\right)(\beta SI - (d+\gamma)E)$$
$$+ B\left(1 - \frac{I^*}{I}\right)(\gamma E - (d+\mu)I)$$
$$= -(d+\mu)\frac{(S-S^*)^2}{S}$$
$$+ \beta S^*I^*\left(1 - \frac{SI}{S^*I^*} - \frac{S^*}{S} + \frac{I}{I^*}\right)$$
$$+ \beta S^*I^*\left(1 - \frac{SIE^*}{S^*I^*E} - \frac{E}{E^*} + \frac{SI}{S^*I^*}\right)$$
$$+ B\gamma E^*\left(1 - \frac{EI^*}{E^*I} - \frac{I^*}{I} + \frac{E}{E^*}\right)$$

Substituting the value of B, we find:

$$V_2^{'} \leq -(\mathrm{d}+\mu)\frac{(S-S^*)^2}{S} + \beta S^*I^*C$$

Where:

$$C = 3 - \frac{S^*}{S} - \frac{SE^*I}{S^*EI^*} - \frac{EI^*}{E^*I}$$

Let $C = -\mathrm{g}\left(\frac{S^*}{S}\right) - \mathrm{g}\left(\frac{SE^*I}{S^*EI^*}\right) - \mathrm{g}\left(\frac{EI^*}{E^*I}\right)$, According to the properties and meaning of logarithm, we can get $C \leq -\mathrm{g}\left(\frac{S^*}{S}\right) - \mathrm{g}\left(\frac{SE^*I}{S^*EI^*}\right) - \mathrm{g}\left(\frac{EI^*}{E^*I}\right) \leq 0$, so $V_2^{'} \leq 0$.

Therefore, when $R_0 > 1$, according to the stability theory [24], Q^* is globally asymptotically stable in Ω. From Theorem 3 and Theorem 4, when $R_0 > 1$, system (2) will eventually stabilize at Q^*, and the malicious code in the network will always exist.

C. Sensitive Evaluation

Research and comprehension of the change in the basic reproduction number R_0 are critical in cleaning up malicious Code in the network. Parameter sensitivity analysis can help determine the impact of changing parameter values on the intensity of viral transmission, allowing for the development of the best preventive and control approach. The basic reproduction number of system (2) is determined by parameters such as β, d, γ, and μ. The sensitivity of R_0 to these parameters is calculated by the following formula:

$$S(R_0,\ param) = \frac{param}{R_0} \bullet \frac{\partial R_0}{\partial [param]} \quad (12)$$

The sensitivity of R_0 to the parameters β, d, γ, and μ is

described as follows:

$$S(R_0,\beta)=1$$

$$S(R_0,d)=1-\frac{d(3d+2\gamma+\mu)}{(d+\mu)(d+\gamma)}$$

$$S(R_0,\gamma)=\frac{d}{d+\gamma}$$

$$S(R_0,\mu)=-\frac{2\mu}{d+\mu}$$

To further investigate the changes, we present the parameter values and calculate the sensitivity coefficients under specific conditions. The results of these calculations are displayed in Table III. It is evident from Table IV that β, d, γ, and R_0 exhibit a positive correlation. Specifically, as β, d, and γ increase, R_0 also increases correspondingly. In detail, when the value of β rises by 1%, R_0 will similarly increase by 1%; when d increases by 1%, R_0 will rise by approximately 0.99%; and when γ increases by 1%, R_0 will experience an increase of about 0.00002%. Furthermore, there exists a strong negative correlation between μ and R_0. When μ increases by 1%, R_0 is expected to decrease by approximately 1.9997%.

Through calculation and analysis, we can know that β, d, and μ have a very strong impact on R_0 and are the key factors affecting R_0. Therefore, when formulating a defense strategy, giving priority to reducing the infection rate β of the node, reducing the dynamic change rate d of the node, and increasing the recovery rate μ of the infected node can effectively reduce the value of R_0 and control the spread of malicious code. Secondly, reducing the activation rate γ of the latent virus can also control the spread of the virus to a certain extent.

TABLE III
ELEMENT VALUES AND SENSITIVITIES

Parameter	Value	Sensitive
β	0.0000002	1
d	0.0000006	0.99955
γ	0.004	0.00002
μ	0.004	-1.9997

V. NUMERICAL SIMULATION EXPERIMENTS

This section presents numerical simulation experiments conducted using the MATLAB R2023b platform under the Intel Core i5-8300H CPU, 2.30GHz main frequency, 8GB memory, and Windows 10 operating system environment. The experiment aims to verify the theorem's correctness and observe the impact of state transition parameters on the number of infected nodes. Reference [26] to set the key parameter values of the experiment.

A. Change of each state Over Time

In the first experiment, the initial number of nodes is S(0) = 90000, E(0) = 5000, I(0) = 5000, A(0) = 0, and R(0) = 0. The initial parameter value is d = 0.0000006, β = 0.0000002, ξ = 0.0000003, γ = 0.004, μ = 0.004, ω = 0.004, thus the threshold value R_0 = 0.000375 < 1. According to Theorem 1 and Theorem 2, the malicious code in the network will eventually disappear. The experimental results are shown in Figure 3, which shows the change in the number of nodes in different states over time. The figure reveals that during the initial stage of the system, the invasion of malicious code transforms susceptible nodes into latent nodes, leading to a slight increase in the number of latent nodes in a short period. Subsequently, the malicious code is activated, and the latent nodes begin to transform into infected nodes. This transformation process leads to a decrease in the number of latent nodes and an increase in the number of infected nodes. Ultimately, the number of both sides gradually decreases and eventually tends to 0. The number of immune nodes tends to stabilize, which indicates that the entire system finally reaches stability, the virus in the network will eventually disappear, and the whole network is finally in an immune state. This process fully verifies the correctness of Theorem 1 and Theorem 2, that is, under the appropriate defense mechanism, the virus in the network can be effectively controlled and eventually eliminated.

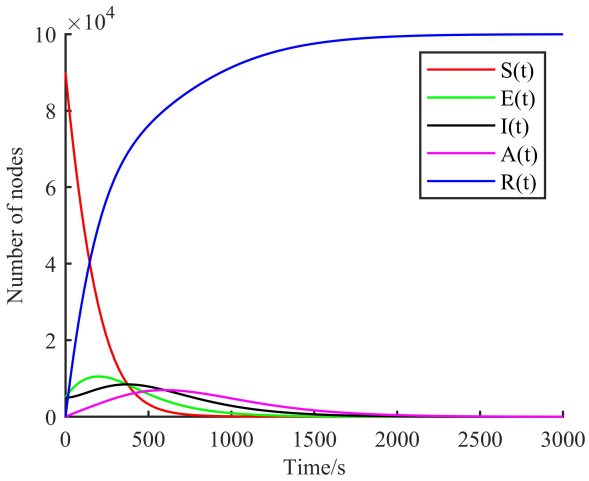

Fig. 3. When R_0 < 1, Changes in the number of nodes in each state over time

In the second experiment, The initial number of nodes is S(0) = 80000, E(0) = 10000, I(0) = 10000, A(0) = 0, and R(0) = 0. The initial parameter value is d = 0.0000006, β = 0.0000002, γ = 0.004, μ = 0.00004, ω = 0.004, thus the threshold value R_0=7.21 > 1.

According to Theorems 3 and 4, infected nodes in the network will not disappear. Figure 4 displays the detailed evolution of nodes over time, based on the experimental data. In the early stages of the system, the number of infected nodes is modest. However, as time passed, malicious code moved quickly throughout the network, and nodes that were previously susceptible were gradually turned into infected nodes, resulting in a considerable increase in the number of infected nodes in the system. The number of infected nodes in the system eventually stabilizes, indicating that the system as a whole is stable.

The experimental results support Theorems 3 and 4, namely that harmful code will always exist and cannot be eradicated, and nodes in the network will continue to be affected by malicious code. This is extremely important for understanding and blocking the transmission of harmful code over the network. To reduce the impact of malicious code attacks on network systems, continual monitoring and defensive measures should be implemented.

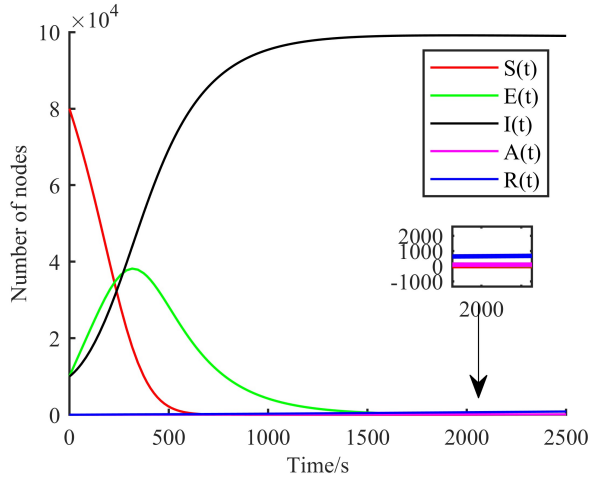

Fig. 4 When $R_0 > 1$, Changes in the number of nodes in each state over time

B. *Impact of the Alert Mechanism on the System*

Experiments 3 and 4 are designed to investigate the effect of the alert system on the progression of the number of infected nodes. In the SEIAR model, if the alert rate ξ is 0, the model's alert mechanism becomes useless, resulting in the SEIR model. Changing the alarm rate ξ has a direct impact on controlling the spread of harmful code. In these two studies, we used alert rates of 0, 0.003, 0.00003, and 0.0000003 to see how the number of infected nodes in the system changed as the alert rate varied.

In Experiment 3, except for ξ, all nodes have the same beginning number and parameter values as in Experiment 1, with R_0 values smaller than 1. The experimental results are depicted in Figure 5. In the early stages of the system, the number of infected nodes rapidly increases to a peak, then gradually falls and eventually disappears. Increasing the value of ξ results in fewer infected nodes. Table IV shows that setting ξ to 0.0000003, 0.00003, or 0.003 reduces the peak number of infected nodes by 4.69%, 36.94%, and 43.57%, respectively, compared to the SEIR model without an alert mechanism. The higher the value of ξ, the faster the system stabilizes and removes dangerous code from the network.

In Experiment 4, except for ξ, all nodes have the same initial numbers and parameter values as in Experiment 2, with R_0 values greater than 1. The experimental results are depicted in Figure 6. The number of infected nodes in the system continues to climb until it reaches stability. Smaller ξ values result in a higher peak number of infected nodes. Table IV shows that when ξ is 0.0000003, 0.00003, or 0.003, the peak number of infected nodes drops by 1.07%, 30.74%, and 84.70%, respectively, compared to the SEIR model without an alert mechanism. The higher the value of ξ, the sooner the system stabilizes and successfully controls dangerous code in the network. Although the infected nodes in the system will not disappear at this time, the presence of the alert mechanism can significantly limit the number of infected nodes in the system and the impact of harmful code on the network.

Combining the results of Experiments 3 and 4, the SEIR model has a substantially greater peak number of infected nodes than the SEIAR model, and the system achieves stability sooner. The alert mechanism's usefulness in limiting the spread of malicious code in the control system has been proven over time. As a result, increasing the system's alert rate can delete malicious code more quickly, allowing the system to achieve stability faster. We can use methods like optimizing information transmission strategies and strengthening the collaboration capabilities of network nodes to build an efficient information dissemination network, enhance the network's ability to respond quickly to malicious code, improve the overall anti-virus performance of the network, and more effectively curb the spread of malicious code in the network to ensure the system's security.

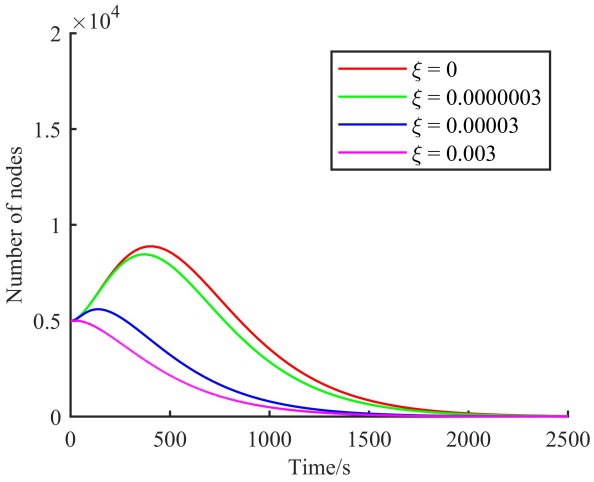

Fig. 5 When $R_0 < 1$, Changes of infected nodes in the system under different ξ values

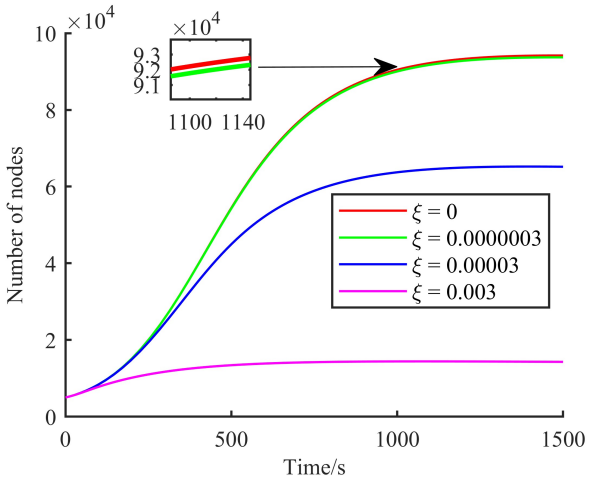

Fig. 6 When $R_0 > 1$, Changes of infected nodes in the system under different ξ values

TABLE IV
COMPARISON OF SEIR INFECTED NODE PEAK RATIO UNDER DIFFERENT ξ VALUES

ξ	Experiment 3($R_0 < 1$)	Experiment 4($R_0 > 1$)
0.0000003	4.69%	1.07%
0.00003	36.94%	30.74%
0.003	43.57%	84.70%

C. *Comparison of the New System with Other Systems*

Experiments 5 and 6 observed the changes in infected nodes over time under different inhibition strategies. Select the new SEIAR system as system 1. Select the traditional SEIR system as System 2. In the literature [26], the malicious

code propagation model of isolation strategy is adopted as System 3.

Experiment 5 selects S(0) = 90000, E(0) = 5000, I(0) = 5000, A(0) = 0, R(0) = 0, and the values of each parameter are selected as d = 0.0000006, β = 0.0000002, γ = 0.004, μ = 0.004, ω = 0.004, at this time, All three systems have $R_0 < 1$. Among them, ξ is a parameter unique to System 1, ξ = 0.00003. When ξ = 0, system 1 degenerates into system 2. And θ is a parameter unique to System 3, θ = 0.0004. The experimental results are shown in Figure 7. The number of infected nodes quickly rises to a peak and then slowly decreases until it disappears. As shown in Table V, the peak value of infected nodes in System 1 is higher than that in System 2 and System 3, and the peak value is 36.94% lower than that in System 2 and 36.55% lower than that in System 3.

Experiment 6 selects, S(0) = 90000, E(0) = 5000, I(0) =5000, A(0)= 0, and R(0) = 0. The values of each parameter are selected as d = 0.0000006, β = 0.0000002, γ = 0.004, μ = 0.00004, ω = 0.004, θ = 0.0004, and $R_0 > 1$ for both systems. The experimental results are shown in Figure 8. In the early stage of the system, the infected nodes quickly rise to the peak value and then slowly decline until the system is balanced and stable. At this time, the infected nodes in the system will not disappear. As shown in Table V, the peak value of infected nodes in system 1 is 30.74% lower than that in system 2 and 30.13% lower than that in system 3. System 1 reaches equilibrium faster than System 2 and System 3. In a balanced network, the impact of malicious code in System 1 is smaller than in System 2 and System 3.

Based on the results of Experiments 5 and 6, the SEIAR model described in this research is effective, and the developed alert mechanism outperforms the classic SEIR model of System 2 and the suppression approach of System 3. In the following tests, we will continue to extensively study the influence of each parameter in the model on the propagation of malicious code, with the goal of finding a more optimized parameter setting to increase the accuracy and usefulness of the model.

D. The Impact of Other Parameters on the Number of Infected Nodes

Experiments 7 and 8 measure how the infection rate β affects the number of infected nodes. The β values for the two experiments are 0.00000009, 0.0000002, 0.0000004, and 0.0000006, respectively.

In Experiment 7, nodes in all states except β have the same initial number and parameter values as in Experiment 1, with R_0 values smaller than 1. The experimental results are depicted in Figure 9. In certain circumstances, the number of infected nodes grows rapidly in the early stages, reaches a peak, and then gradually lowers until it disappears. Increasing the value of β leads to more infected nodes and a higher peak value. Smaller β values lead to faster system stability.

In Experiment 8, nodes in all states except β have the same beginning number and parameter values as in Experiment 2, with R_0 values greater than 1. Figure 10 shows the experimental outcomes. The higher the β value, the faster the number of infected nodes grows, the shorter the time it takes to reach the peak, and the more infected nodes remain when the system stabilizes. Smaller β values result a slower system stabilization time. At the same time, there are fewer infected nodes after the system stabilizes.

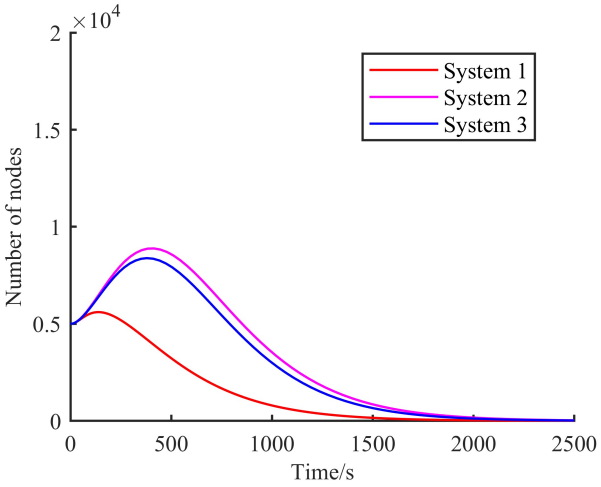

Fig. 7 When $R_0 < 1$, The number of infected nodes in different systems changes over time

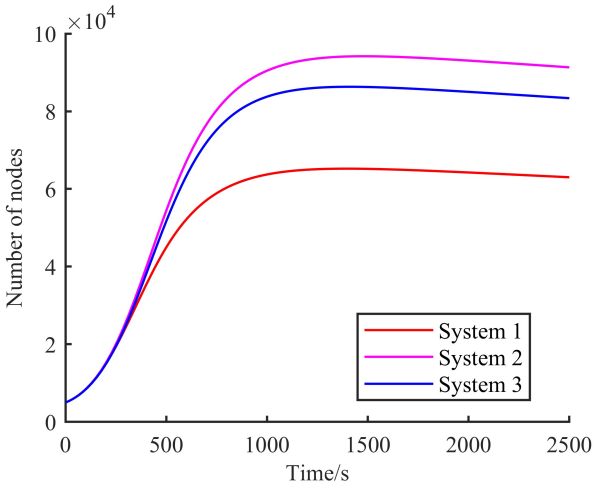

Fig. 8 When $R_0 > 1$, The number of infected nodes in different systems changes over time

Table V
THE PROPORTION OF INFECTED NODES REDUCED IN THE NEW SYSTEM COMPARED TO OTHER SYSTEMS

System	The percentage reduction in Experiment 5($R_0 < 1$)	The percentage reduction in Experiment 6($R_0 > 1$)
System2	36.94%	30.74%
System3	36.55%	30.13%

The results of Experiments 7 and 8 show that the infection rate β is a key parameter that determines the speed and scale of malicious code propagation. To better ensure the stability of the system, it is necessary to control the value of the infection rate to the maximum extent. Education and training can be used to improve public awareness of network security threats, strengthen the audit of smart contract codes, establish emergency response mechanisms to quickly take countermeasures in the early stages of virus transmission, and strengthen the detection capabilities of malicious codes. This can minimize the ability of malicious code to spread and infect, slow down the spread of malicious code, and limit its scope of influence.

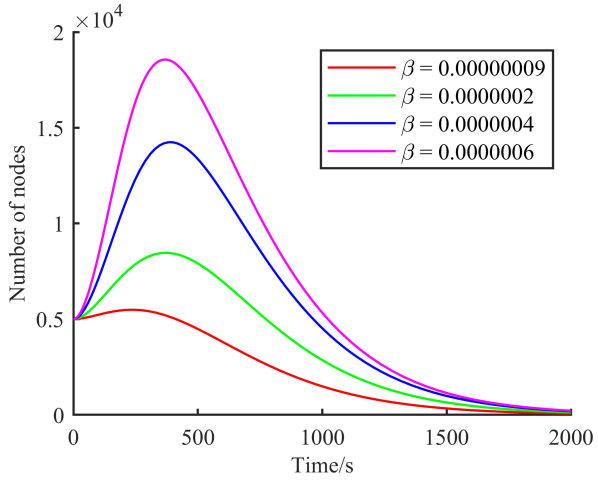

Fig. 9 When $R_0 < 1$, Changes of infected nodes in the system under different β values

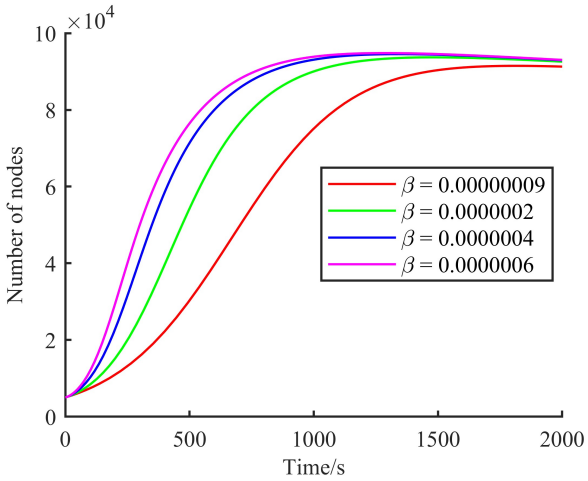

Fig. 10 When $R_0 > 1$, Changes of infected nodes in the system under different β values

Experiments 9 and 10 measure how activation rate γ affects the number of infected nodes. The γ values for the two experiments are 0.002, 0.004, 0.006, and 0.008, respectively.

In Experiment 9, nodes in all states except γ have the same beginning number and parameter values as in Experiment 1, with R_0 values smaller than 1. The experimental results are depicted in Figure 11. In the early stages of the system, the number of infected nodes rapidly increases, indicating that the malicious code in the nodes is easily triggered. As time passes, the number of infected nodes reaches a high before gradually decreasing until it disappears entirely. Increasing the activation rate γ leads to faster proliferation of infected nodes and a larger peak value. This shows that under the same control measures, a higher activation rate will lead to more serious infections, but the malicious code in the network will eventually be cleared.

In Experiment 10, nodes in all states except γ have the same beginning number and parameter values as in Experiment 2, with R_0 values greater than 1. The experimental results are depicted in Figure 12. The number of infected nodes increases fast in the early stages before stabilizing. This means that unless appropriate control mechanisms are implemented, the disease or malicious code will continue to spread throughout the network. As γ increases, the number of infected nodes grows faster, the time to achieve the peak decreases, and the number of infected nodes increases after stability. This indicates that a higher activation rate will accelerate the spread of malicious code, resulting in more nodes being infected in the end.

Experiments 9 and 10 showed that lowering the activation rate can dramatically reduce losses associated with network virus defense. Efforts to improve detection efficiency and network participant vigilance against malicious code might increase the activation rate γ of latent nodes.

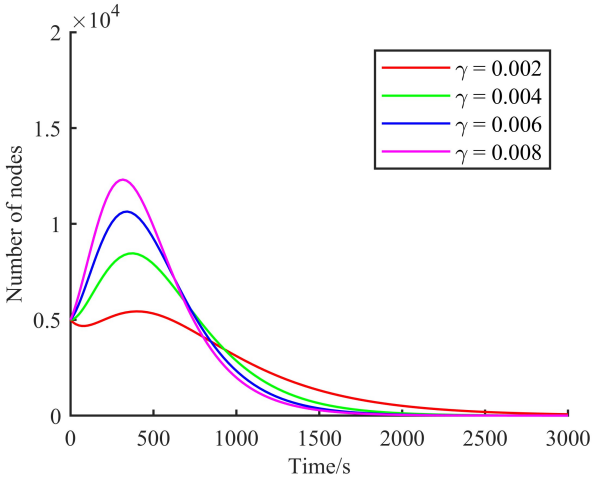

Fig. 11 When $R_0 < 1$, Changes of infected nodes in the system under different γ values

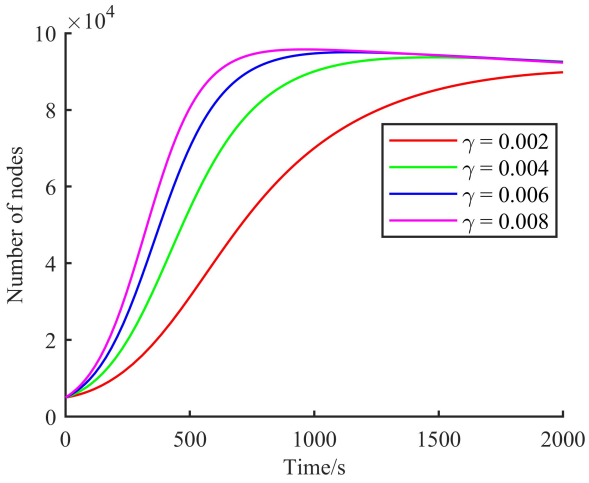

Fig. 12 When $R_0 > 1$, Changes of infected nodes in the system under different γ values

Experiments 11 and 12 test how changing the immunity rate μ affects the number of infected nodes over time. The μ values for the two experiments are 0.002, 0.004, 0.006, and 0.008, respectively.

In Experiment 11, except for the immunity rate μ, the initial number and parameter values of the nodes in other states are the same as in Experiment 1, with all associated R_0 values smaller than 1. The experimental results are depicted in Figure 13. The number of infected nodes climbs rapidly in the early stages, then begins to fall after reaching a peak, and eventually approaches zero, suggesting that the malicious code has been successfully managed. As the immunity rate μ increases, the peak number of infected nodes drops and the duration to reach the peak increases. This demonstrates that a higher immunity rate helps to eliminate infected nodes faster, therefore more effectively regulating the spread of the disease

or dangerous code.

In Experiment 12, all state nodes except μ have the same beginning number and parameter values as in Experiment 2, with R_0 values greater than one. The experimental results are depicted in Figure 14. The number of infected nodes grows significantly in the early stages before gradually stabilizing, showing that the malicious code remains in the group. As the immunity rate μ rises, the number of infected nodes grows faster, and the time to reach a stable state increases. The number of infected nodes in the stable state reduces as μ increases. This demonstrates that in this scenario, boosting the immunity rate can accelerate the speed at which the network achieves a stable state and minimize the number of infected nodes in the final network.

Experiments 11 and 12 show that increasing the immunity rate μ in the system increases the possibility of the infected node gaining immunity, allowing for faster infection control. Improving virus detection and node recovery can increase the probability of infected nodes gaining immunity, reducing the need for defense against malicious code and preventing its spread in the network.

In Experiment 13, except for the conversion rate ω, the beginning numbers and parameter values of other state nodes are consistent with those in Experiment 1. The variable ω represents the likelihood of converting an alert node to an immune node per unit of time. The experimental values for ω are 0.0004, 0.004, 0.04, and 0.4.

The experimental results are shown in Figure 15. The number of infected nodes experiences rapid growth during the initial phase of the system, followed by a gradual decline after reaching its peak until it ultimately disappears. Notably, lower values of ω correspond to reduced peak levels of infected nodes, indicating that a diminished conversion rate is beneficial for controlling these peaks. The waiting time required for an alert node to convert into an immune node plays a crucial role in influencing ω. During this process, the alert node disseminates prevention and control information to adjacent nodes while awaiting responses from both these nodes and their administrators. An extended fixed waiting time T results in smaller values for ω and consequently slows down the growth rate of infected nodes within any given timeframe. Furthermore, variations in ω also exert influence over the peak value of infected nodes; higher values facilitate quicker reductions in infection rates but may simultaneously lead to elevated peak counts among infected individuals. Overall, while ω has relatively minor effects on total infection numbers within the system, optimizing alert mechanisms and enhancing the efficacy of alert nodes can significantly mitigate damage caused by malicious code to network participants.

The experimental results presented above indicate that parameters such as β, γ, μ, ω, and ξ are critical factors influencing the variation in the number of infected nodes. Among these parameters, β and γ have a direct impact on the basic reproduction number, which is essential for assessing the trends in malicious code propagation. β represents the probability of susceptible nodes becoming infected. A higher value of β correlates with an increased R_0, signifying a greater likelihood of network participants being compromised. γ denotes the probability that latent malicious code will be activated; thus, a larger γ value enhances the chances that dormant malicious code within a node will convert it into an infected state. These two parameters are pivotal in determining R_0. By reducing their values within the network, one can effectively control the range of R_0 and consequently influence both the extent and duration of malicious code dissemination. μ indicates the probability that infected nodes achieve immunity. The higher the μ value, the more nodes acquire immunity per unit time. ω reflects the likelihood of alert nodes transitioning to immune status. A lower ω value suggests a greater presence of alert nodes within the network, thereby facilitating more effective dissemination of alert information. ξ signifies the probability that susceptible nodes gain immunity directly through alerts issued by other nodes. Optimizing this alert mechanism can significantly enhance ξ, enabling faster attainment of immunity among networked nodes and accelerating system stability. The interplay among these transition probabilities is crucial for managing malicious code spread effectively. By manipulating these parameters, defenses within the blockchain network can be enhanced while mitigating the potential impact of malicious intrusions, thereby limiting losses to manageable levels.

VI. Conclusion

This study not only theoretically verifies the SEIAR model's correctness, but also the effectiveness of the alert mechanism in numerical simulation, providing a new perspective for a more in-depth understanding of the propagation mechanism, as well as prevention and control strategies for malicious code in blockchain networks. In-depth research on the transmission mechanism of malicious code leads to more precise security protection measures for blockchain networks, as well as new ideas for developing a safer blockchain network. In the future, consider including more parameters into the model, such as the time lag of network information propagation and efficient malicious code detection techniques, to improve the malicious code propagation model and bring it closer to the actual network environment. The research will be expanded to numerous types of blockchain networks, and the application of the model in multiple scenarios will be investigated to encourage the healthy development of blockchain technology.

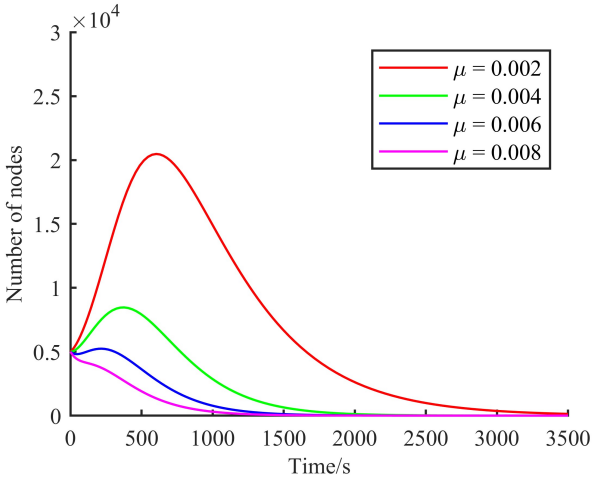

Fig. 13 When $R_0 < 1$, Changes of infected nodes in the system under different μ values

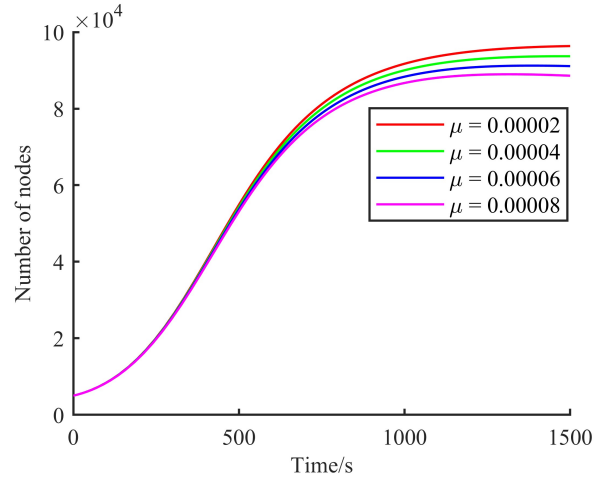

Fig. 14 When $R_0 > 1$, Changes of infected nodes in the system under different μ values

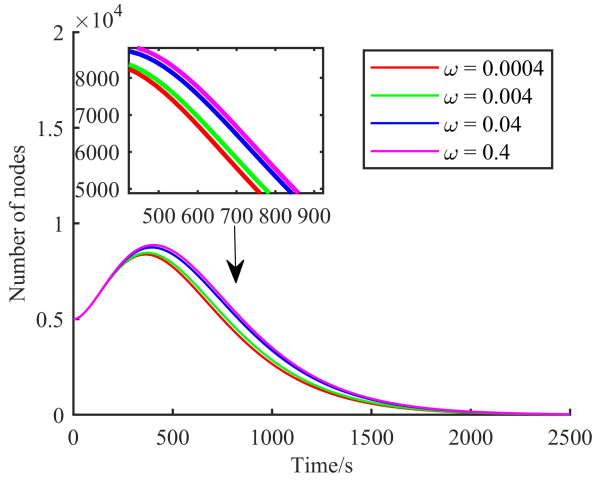

Fig. 15 When $R_0 < 1$, Changes of infected nodes in the system under different ω values

REFERENCES

[1] Jang-Jaccard, Julian, and Surya Nepal. "A survey of emerging threats in cybersecurity." Journal of computer and system science, vol.80, no.5, pp973-993, 2014

[2] Wu Guangyu, Jian Sun, and Jie Chen. "A survey on the security of cyber-physical systems." Control Theory and Technology, vol.14, no.1, pp2-10, 2016

[3] Zeng S Q, Huo R, Huang T, et al. "Survey of blockchain: principle, progress and application." Journal on Communications, vol.41, no.1, pp134-151, 2020

[4] Cai X Q, Deng Y, Zhang L, et al. "The principle and core technology of blockchain." Chinese journal of computers, vol.44, no.1, pp84-131, 2021

[5] Tian GH, Hu YH, Chen XF. "A survey on attack and defense of block-chain system." Journal of Software, vol.32, no.5, pp1459-1525, 2021

[6] Ke Yuan, Yingjie Yan, Lin Shen, Qian Tang, and Chunfu Jia, "Blockchain Security Research Progress and Hotspots." IAENG International Journal of Computer Science, vol.49, no.2, pp433-444, 2022

[7] Liu A D, Du X H, Wang N, et al. "Research progress of blockchain technology and its application in information security." journal of software, vol.29, no.7, pp2092-2115, 2018

[8] Fang Z, Zhao P, Xu M, et al. "Statistical modeling of computer malware propagation dynamics in cyberspace." Journal of Applied Statistics, vol.49, no.4, pp858-883, 2022

[9] Frutos-Bernal E, Rodríguez-Rosa M, Anciones-Polo M, et al. "Analyzing Malware Propagation on Wireless Sensor Networks: A New Approach Using Queueing Theory and HJ-Biplot with a SIRS Model." Mathematics, vol.12, no.1, pp135, 2023

[10] Kephart J O, White S R. "Directed-graph epidemiological models of computer viruses." Computation: the micro and the macro view, pp71-102, 1992

[11] Mishra B K, Prajapati A. "Dynamic model on the transmission of malicious codes in network." International Journal of Computer Network and Information Security, vol.5, no.10, pp17, 2013

[12] Toutonji O A, Yoo S M, Park M. "Stability analysis of VEISV propagation modeling for network worm attack." Applied mathematical modeling, vol.36, no.6, pp2751-2761, 2012

[13] Li F, Ren J. "Suppression of Malicious Code Propagation in Software-Defined Networking." Wireless Personal Communications, vol.135, no.1, pp493-516, 2024

[14] Ding Jian, Zhao Tao, Liu Zhigang, and Guo Qiong. "Stability and bifurcation analysis of a delayed worm propagation model in mobile internet." IAENG International Journal of Computer Science, vol.47, no.3, pp533-539, 2020

[15] Tang W, Liu Y J, Chen Y L, et al. "SLBRS: network virus propagation model based on safety entropy." Applied Soft Computing, 2020

[16] Zhang Z, Kundu S, Wei R. "A delayed epidemic model for propagation of malicious codes in wireless sensor network." Mathematics, vol.7, no.5, pp396, 2019

[17] Liu G, Peng B, Zhong X, et al. "Differential games of rechargeable wireless sensor networks against malicious programs based on SILRD propagation model." Complexity, 2020.

[18] Huang S Z. "A new SEIR epidemic model with applications to the theory of eradication and control of diseases, and to the calculation of R0." Mathematical Biosciences, vol.215, no.1, pp84-104, 2008

[19] Madinei H, Rezazadeh G, Azizi S. "Stability and bifurcation analysis of an asymmetrically electrostatically actuated microbeam." Journal of Computational and Nonlinear Dynamics, vol.10, no.2, pp, 2015

[20] Jackson M, Chen-Charpentier B M. "Modeling plant virus propagation with delays." Journal of Computational and Applied Mathematics, vol.30, no.9, pp611-621, 2017

[21] La Salle J P. "The stability of dynamical systems." Society for Industrial and Applied Mathematics, vol., no., pp, 1976

[22] Clark R N. "The Routh-Hurwitz stability criterion, revisited." IEEE Control Systems Magazine, vol.12, no.3, pp119-120, 1992

[23] Sigdel R P, McCluskey C C. "Global stability for an SEI model of infectious disease with immigration." Applied Mathematics and Computation, pp684-689, 2014

[24] Lavretsky E, Wise K A. "Lyapunov stability of motion." Robust and Adaptive Control: With Aerospace Applications, pp373-411, 2024

[25] Li C, Ren J, Li F. "A Novel Malicious Code Propagation Model Based on Dual Defense and Honeypot Feedback." International Journal of Network Security, vol.26, no.4, pp622-634, 2024

[26] WANG G, LU S, HU X. "Network virus spreading SEIQRS model and its stability under escape mechanism." Journal of Harbin Institute of Technology, vol.51, no.5, pp131-137, 2019