Hindawi
Scientific Programming
Volume 2022, Article ID 6800135, 11 pages
https://doi.org/10.1155/2022/6800135

Research Article

Evaluation and Prediction Method of System Security Situational Awareness Index Based on HMM Model

Mengjie Qian

Information Engineering Department, Hebei Vocational University of Technology and Engineering, Xingtai 054000, China

Correspondence should be addressed to Mengjie Qian; polaris119@hdu.edu.cn

Received 7 February 2022; Revised 2 March 2022; Accepted 4 March 2022; Published 12 April 2022

Academic Editor: Hangjun Che

In recent years, with the continuous development and progress of information technology and science and technology, big data has entered all walks of life, integrated into the lives of the public, and has become a necessity for social operation; the gradual development of artificial intelligence has also made life in modern times. People in society are more and more convenient. However, the development of science and technology is also accompanied by corresponding problems, and the war in information has gradually started. This paper simulates the possible information security through the hidden Markov model and then verifies the feasibility and effectiveness of the situation assessment method and the situation prediction method, in order to effectively evaluate the relevant information security level and effectively predict the accuracy of the situation value. The experimental results show that the fluctuation of the situation value corresponds to the different attack behaviors carried out by the attacker, accurately describes the information security status of the system, and verifies the effectiveness and accuracy of the situational awareness method proposed in this paper, while the situation prediction method based on ARIMA predictable short-term changes in situational values can be used for short-term forecasts that require high accuracy.

1. Introduction

In recent years, with the continuous improvement of people's economic income, mobile phones and computers have become the necessities of every family. Cybersecurity cases often occur in our lives, our cyberspace security is facing severe challenges, and many enterprises or organizations will also face cyberattacks, such as the once-famous "Aurora Attack," which attacked Google's mail server attacked. Some foreign hacker organizations hope to steal our country's military secrets and understand our country's political and economic situation by attacking our country's software. In China, some criminals use the Internet to attack some websites to steal the internal data of the websites and defraud some elderly people through the Internet. These situations affect the social atmosphere and reduce the happiness of residents' lives. As these problems become increasingly prominent, it should be put on the agenda to improve system security protection capabilities, pay attention to network security, and monitor and predict possible events. When defending, improve your defense methods and fully understand the network security dynamics; you need to find potential threats as soon as possible, find out possible malicious behaviors, determine the source of the attack, provide important information for eliminating security threats and update network data in a timely manner, and actively fight against security risks.

With the continuous development and progress of information technology and technology, issues related to big data, artificial intelligence, and security posture have entered people's lives, and everyone has gradually become familiar with these once unfamiliar words. Company executives make decisions through big data [1]; the DBES problem [2]; the use of wireless mesh networks [3]; the deployment of intelligent transportation systems [4]; the rise of big data in cloud computing [5]; evaluation research on bank performance [6]; emergence of behavior-oriented artificial intelligence [7]; predicting lake level fluctuations [8]; birth of AIDR [9]; artificial intelligence applications [10]; birth of ADS system [11]; prediction of security situation [11];

security aggregation and configuration related data [12]; analysis of sources and identification methods of information risks [13]; application of artificial intelligence in music education [14]; performance improvement of water pollution monitoring and rapid decision-making systems [15]; network security situation prediction [16]; the combination of artificial intelligence algorithms and physical modeling [17]; and so on.

2. Theoretical Basis

2.1. Situational Awareness

2.1.1. Related Situational Awareness Profiles. Situational awareness was first proposed by Endsley et al. in 1988. It is defined as "the extraction and understanding of the surrounding environmental factors within a certain time and space, and the prediction of the future development trend." Situational awareness originated from the military field and was soon applied to other fields. The application of situational awareness in network security is even more extensive and far-reaching; in enterprise information security management, the defense-discovery-repair approach is adopted [18].

2.1.2. Situational Awareness Model. The three-layer situational awareness model visually represented in Figure 1 is a widely accepted general theoretical model given by Endsley. The model is composed of situational element extraction, situational understanding, and situational prediction and plays an important role in the subsequent research on network security situational awareness systems.

2.1.3. Security Situation Forecast. Security situation prediction is the highest level technology in the whole situational awareness model [19]. The prediction of network security situation plays an important role in the defense of network security. The definition of situation prediction is to make a preestimation of the events or scenarios that will occur in the future to determine the probability of its occurrence, which usually requires rigorous investigation and observation, and artificial intelligence algorithms such as machine learning and deep learning can discover and identify potential patterns in input data and output of the required prediction information. They have achieved great success in computer vision, natural language processing, and other fields and are widely used in artificial intelligence algorithms. They have also been used in network security situation prediction and achieved initial results. Therefore, according to certain scientific basis, through the analysis and study of relevant factors, a specific prediction model as shown in Figure 2 is established.

2.1.4. Cybersecurity Situational Awareness. The network situation is a network state and changing trend, which is affected by factors such as different types of network operating conditions, network behaviors, and user behaviors. These behaviors are combined together to form a network situation. In a large-scale network environment, we can select those security elements for analysis, fully understand the changes in the network situation, and use big data technology to process different types of information. The perception platform integrates user terminals, through different types of perception data sources, fully explores technologies, understands intelligent algorithms, and improves the sensitivity of the network security situational awareness platform.

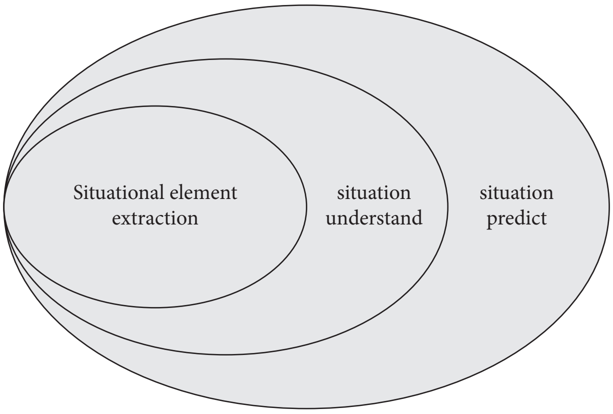

Figure 1: 3-layer situational awareness model.

Situational awareness describes the network system and requires a full understanding of its microstate, which is reflected by various connection parameters. After in-depth mining of parameters, determine the correlation and development trend of information, and use related tools (algorithms or measures) to detect and perceive, and associate the data and information from detection and perception in some way to form knowledge. This completes a basic process of situational awareness as shown in Figure 3.

We know that the biggest feature of network situational awareness [20] is the need to measure the physical network. The following abilities serve as support:

(1) Big data processing capability: the Internet is huge, and a large amount of concurrent data traffic is transmitted through each node. In this kind of data reanalysis, Network maintenance personnel cannot meet the requirements of the end-of-network activity and the characteristics of the network situation rate. Therefore, the sampling method to detect abnormal network activities cannot meet the requirements. Network situational awareness must have the ability of very good processing power for massive data.

(2) Fine analysis capability: the detection of data packets must meet fine-grained requirements, and various data streams and parameters must be efficiently retrieved and matched. To achieve a good perception of the network microstate, in addition to the traditional key physical parameters (source and destination addresses, ports, and protocol types), situational awareness technology must also have the ability to identify multiple logical parameters.

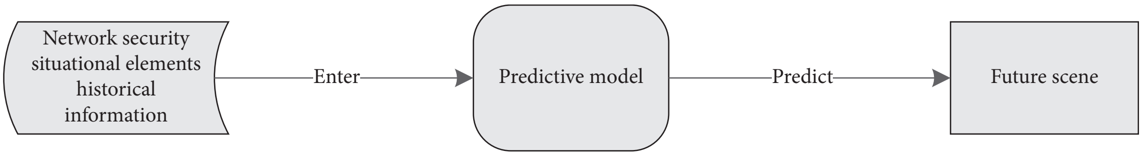
Network security situational elements historical information
Enter
Predictive model
Predict
Future scene

Figure 2: Security situation prediction model.

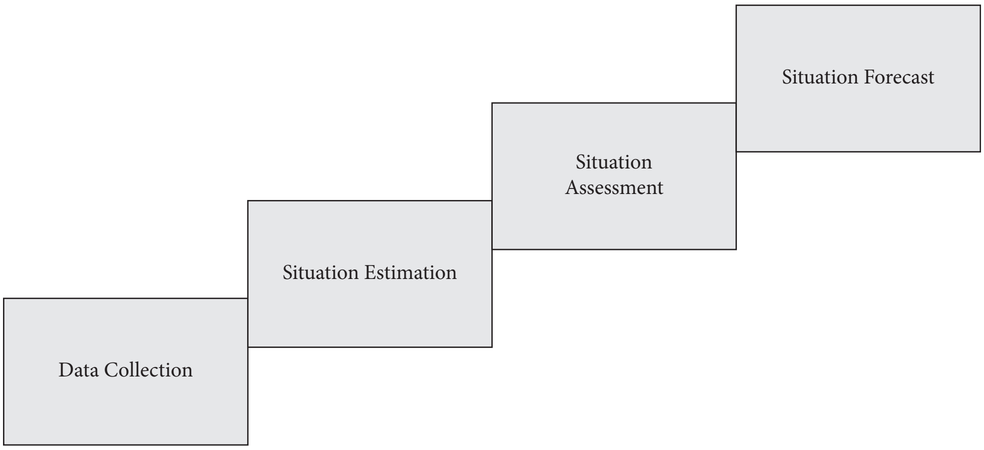

Figure 3: Conceptual model of network security situational awareness.

(3) Protocol identification capability: as an important parameter, the protocol type is very important to accurately identify it. In the process of implementing situational awareness, in addition to identifying widely used Internet standard protocols (such as TCP/IP protocol suite, etc.), other nonstandard and private protocols should also be captured as much as possible. Security poses a threat. For the identified protocol features (fingerprints), a fingerprint database shall be formed, which shall be continuously updated and maintained to support the effective matching retrieval function.

(4) Reliable operation capability: situational awareness technology equipment needs to have reliable operation capability to ensure long-term normal operation. Any interruption of the operation process may cause inaccurate perception information.

(5) Business diversion capability: after all kinds of information are accurately identified, they should be redirected in a certain way; that is, after classifying the information, the next step is processed in a targeted manner. This can not only make the data information more accurate, but also greatly reduce the background processing load, improve operating efficiency, and save memory capacity.

3. Hidden Markov Model

Hidden Markov model is a commonly used probability model in statistics. In this section, the concept of hidden Markov model is introduced from the classic stock market problem, and then three types of problems and solutions of hidden Markov model are introduced. The observation sequence problem extends the one-dimensional hidden Markov model to a multidimensional hidden Markov model.

3.1. Overview of Hidden Markov Models (HMM).

3.1. Overview of Hidden Markov Models (HMM). Hidden Markov model (HMM) is a powerful probabilistic modeling tool for characterizing implicit stochastic processes with observable sequences. HMM have been used in areas such as signal processing, pattern recognition, and machine learning. At the beginning of the twentieth century, Andrei Markov proposed the mathematical theory of Markov Process. Until 1960, Baum and his colleagues proposed and developed the hidden Markov theory model.

Figure 4 depicts a simple example of a Markov process used to describe changes in the stock market. This stochastic process divides the daily changes of the stock market into three states: A city, B city, and a volatile market, which correspond to three observations of a stock's stock price rising, falling, and remaining unchanged. The transition between states is a Markov process with limited time discrete state space, also known as a Markov chain.

Assuming that the probability that the stock is in city A on the first day is 0.7, if it is observed that the stock is rising-falling-falling for three consecutive days, then it can be inferred that the changing state of the stock for three consecutive days is city A-city B-city B, and we can calculate the probability of this happening as

$$p = 0.7 \times 0.2 \times 0.5. \quad (1)$$

If each state is allowed to correspond to multiple observations, for example, when the stock is in market A, not only is there an increase in one observation, but also a

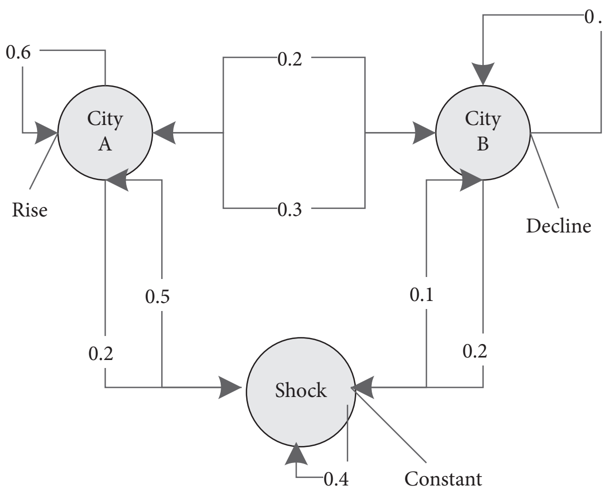

Figure 4: Markov process example.

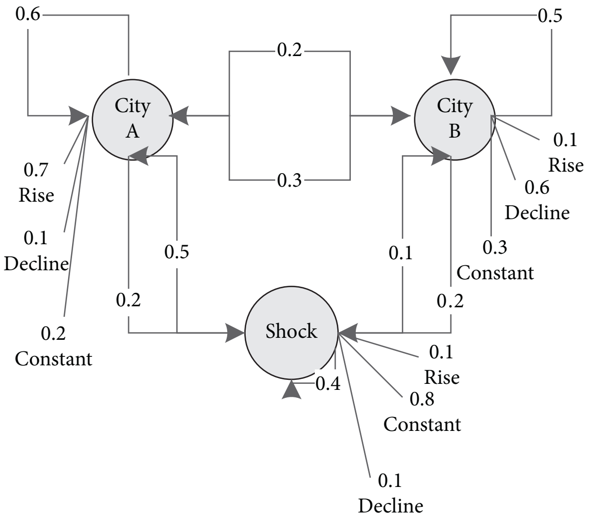

Figure 5: Hidden Markov model example.

decrease or shock may be observed, and the Markov chain can be extended to a hidden Markov model. This change can make the model more expressive. In Figure 5, the stock may also fluctuate or decline slightly in the state of city A. If the stock is observed to rise-fall-fall for three consecutive days, it cannot be said that the stock must be in the B market, so the stock state is "hidden" and can exist in any state sequence with a certain probability.

The mathematical definition of hidden Markov model can be given as follows:

$$\lambda = (A, B, \pi). \tag{2}$$

S is the set of hidden states and V is the set of observable states:

$$\begin{gathered} S = (s_1, s_2, \ldots, s_N), \\ V = (v_1, v_2, \ldots, v_M). \end{gathered} \tag{3}$$

Then define the hidden state sequence Q of length T and the corresponding observation sequence O:

$$\begin{gathered} Q = q_1, q_2, \ldots, q_T, \\ O = o_1, o_2, \ldots, o_T. \end{gathered} \tag{4}$$

A is the implicit state transition matrix. The element a_{ij} of the matrix represents the probability of transitioning from state i to state j. Note that the state transition probability is independent of time:

$$A = \left[a_{ij}\right], a_{ij} = P\left(q_t = S_j | q_{t-1} = S_i\right). \tag{5}$$

B is the observation matrix, and the element $b_i(k)$ of the matrix represents the probability of observing v_k in the hidden state i of the system. This probability is also independent of time:

$$B = \left[b_i(k)\right], b_i(k) = P\left(x_t = v_k | q_t = S_i\right). \tag{6}$$

π is the initial probability distribution matrix, and the elements represent the probability that the system is in each hidden state at the initial moment:

$$\pi = \left[\pi_i\right], \pi_i = P\left(q_1 = S_i\right). \tag{7}$$

The HMM makes two assumptions. The first one is called the Markov assumption, which considers that the current state of the system only depends on the state of the system at the previous moment, which is expressed as

$$P\left(q_t | q_1^{t-1}\right) = P\left(q_t | q_{t-1}\right). \tag{8}$$

The second assumption is called the independence assumption, which considers that the observed state of the system only depends on the implicit state of the system at the current moment, which is expressed as

$$P\left(o_t | o_1^{t-1}, q_1^t\right) = P\left(o_t | q_t\right). \tag{9}$$

3.2. Hidden Markov Three Kinds of Problems. For HMM to be useful in practical applications, three problems related to them must be solved, which are estimation problem, decoding problem, and learning problem.

3.2.1. Estimation Problem. Given an HMM model λ, calculate the probability $P(O|\lambda)$ of occurrence of observation sequence O. This problem can be viewed as evaluating the ability of a known model to predict a given sequence of observations, and by comparing $P(O|\lambda)$ the most appropriate model can be selected. Given a sequence of hidden states Q, the probability of observing sequence O is

$$P(O|Q, \lambda) = \prod_{t=1}^{T} P\left(o_t | q_t, \lambda\right) = b_{q1}\left(o_1\right) b_{q2}\left(o_2\right) \cdots b_{qT}\left(o_T\right). \tag{10}$$

The probability of occurrence of the hidden state sequence Q is

$$P(Q|\lambda) = \pi_{q1} a_{q1q2} a_{q2q3} \cdots a_{qr-1qr}. \tag{11}$$

Given a model, the observation probability can be calculated:

$$\begin{aligned} P(O|\lambda) &= \sum_{Q} P(O|Q,\lambda)P(Q|\lambda) \\ &= \sum_{q1\cdots qr} \pi_{q1} b_{q1}(o_1) a_{q1q2} b_{q2} \\ &\quad \cdot (o_2) a_{q2q3} \cdots a_{qr-1qr} b_{qT}(o_T). \end{aligned} \tag{12}$$

From this, the probability of occurrence of observation sequence O for a given model can be calculated, but its time complexity is in the exponential form with respect to time T (to be precise, it requires $2T.N^T$ calculations). There are a large number of identical operations in the abovementioned calculation process, and these redundant operations can be reduced by means of cache calculation, so as to achieve the purpose of reducing the time complexity. The meshed grid is used to cache the operations that need to be repeated in the calculation process, and the grid can be moved forward until time T to obtain the result. This method is called the forward algorithm. To this end, an intermediate variable α needs to be introduced, which represents the probability that the implicit state is s_i and the observation sequence $o_1, o_2, \cdots, o_t$ is at time t:

$$\alpha_t(i) = P(o_1 o_2 \cdots o_t, q_t = s_i|\lambda). \tag{13}$$

The specific algorithm is as follows:

(1) Initialization:

$$\alpha_1(i) = \pi_i b_i(o_1), 1 \le i \le N. \tag{14}$$

(2) Recursion:

$$\alpha_{t+1}(j) = \left[\sum_{i-1}^{N} \alpha_t(i)\alpha_{ij}\right] b_j(o_{t+1}), 1 \le t \le T-1, 1 \le j \le N. \tag{15}$$

(3) Termination:

$$P(O|\lambda) = \sum_{i-1}^{N} \alpha_T(i). \tag{16}$$

Strictly speaking, the forward algorithm can solve the evaluation problem, but in order to solve the learning problem, a backward algorithm must be introduced, which can also solve the evaluation problem. Similar to the forward algorithm, define an intermediate variable β, given the state s_i at time t, the probability of observing the sequence from o_{t+1} to o_T:

$$\beta_1(i) = P(o_{t+1} o_{t+2} \cdots o_T | q_t = s_i, \lambda). \tag{17}$$

Unlike the forward algorithm, the backward algorithm recurses from the back to the front. The specific algorithm is as follows:

(1) Initialization:

$$\beta_T(i) = 1, 1 \le i \le N. \tag{18}$$

(2) Recursion:

$$\left\{ \beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), t = T-1, T-2, \cdots 1, 1 \le i \le N \right\}. \tag{19}$$

(3) Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_i(i). \tag{20}$$

3.2.2. Decoding Problem. The purpose of decoding is to find the hidden state sequence that is most likely to produce a given observation sequence, that is, the known model λ, and to find the hidden state sequence Q that makes the observation sequence O most likely to appear. The best solution to the decoding problem is to use the Viterbi algorithm, which is another grid algorithm, similar to the forward algorithm, except that the probability at each moment is maximized instead of summing. We can define

$$\delta_t(i) = \max_{q_1, q_2, \cdots, t-1} P(q_1 q_2 \cdots q_t = s_i o_1, o_2, \ldots, o_t|\lambda), \tag{21}$$

which is the probability of the most likely hidden state path that makes the observation sequence appear up to time t. The Viterbi algorithm is as follows:

(1) Initialization:

$$\delta_t(i) = \pi_i b_i(o_1)\ , \ 1 \le i \le N, \Psi_1(i) = 0. \tag{22}$$

(2) Recursion:

$$\begin{aligned} \delta_t(j) &= \max_{1\le i\le N}\left[\delta_t(i) a_{ij}\right] b_j(o_1), 2 \le t \le T, 1 \le j \le N, \\ \Psi_1(j) &= \arg\max_{1\le i\le N}\left[\delta_t(i) a_{ij}\right], 2 \le t \le T, 1 \le j \le N. \end{aligned} \tag{23}$$

(3) Termination:

$$\begin{aligned} P^* &= \max_{1\le i\le N}\left[\delta_r(i)\right], \\ q_r^* &= \arg\max_{1\le i\le N}\left[\delta_r(i)\right]. \end{aligned} \tag{24}$$

(4) Backtracking of the optimal state sequence:

$$q_t^* = \Psi_{t+1}(q_{t+1}^*), t = T-1, T-2, \ldots 1. \tag{25}$$

The main difference between the Viterbi algorithm and the forward algorithm is that the Viterbi algorithm maximizes the probability in the recursive process, rather than summing it up, and stores the state when the probability is the largest, so as to end the used backtracking.

Backtracking allows finding the optimal sequence of states from the states stored in the recursive steps, and there is no easy way to find a suboptimal sequence of states.

3.2.3. Learning Problems. Learning problems are divided into two categories: supervised and unsupervised, corresponding to two standard solutions. Learning problems are divided into two categories: supervised and unsupervised, corresponding to two standard solutions. If the training data set used to solve the learning problem is supervised, that is, when the observation sequence is given, the corresponding hidden state sequence is also specified, and a supervised learning algorithm is used. If the training data set is unsupervised, that is, only the observation sequence is given, the unsupervised learning algorithm, also known as the B-W algorithm, can be used.

The B-W algorithm, jointly proposed by Baum and his colleagues, is a very classic algorithm for solving model parameter selection problems.

In order to describe the estimation process of HMM parameters, the B-W algorithm first defines an intermediate variable to represent the probability of state s_i at time t and state s_{i+1} at time t+1 under the premise of given model parameters and observation sequence:

$$\xi_t(i, j) = P\left(q_t = s_i, q_{t+1} = s_j | O, \lambda\right). \tag{26}$$

The B-W algorithm uses the forward algorithm at time t and the backward algorithm at time t+1, which cleverly combines the forward algorithm and the backward algorithm, also known as the forward-backward algorithm, and $\xi_t(i, j)$ is also called the forward algorithm Backward variable.

According to the Bayesian formula, the forward-backward variable announcement can be written as:

$$\begin{aligned} \xi_t(i, j) &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}, \\ \xi_t(i, j) &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}, \end{aligned} \tag{27}$$

In the formula, the numerator term is $P(q_t = s_i, q_{t+1} = s_j, O|\lambda)$, and the denominator term is $\mathrm{P}(\mathrm{O}|\lambda)$, which is obtained by the full probability formula.

Define $\gamma_t(i)$ as the probability that the model is in state s_i at time t, and establish the relationship between $\gamma_t(i)$ and $\xi_t(i, j)$ as follows:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j). \tag{28}$$

The value obtained by the summation of $\gamma_t(i)$ at time t can be used to represent the expectation of the number of visits to state s_i, that is, the expectation of the number of transitions from state s_i. Likewise, the result of summing $\xi_t(i, j)$ over time can be used to express the expectation of the number of transfers from s_i to s_j.

The idea of the B-W algorithm is to iteratively obtain the new model parameter $\overline{\lambda}$. Once $\mathrm{P}(\mathrm{O}|\overline{\lambda}) > \mathrm{P}(\mathrm{O}|\lambda)$ is found, $\overline{\lambda}$ is assigned to λ and iteratively calculates until $\mathrm{P}(\mathrm{O}|\lambda)$ no longer changes significantly. Therefore, unfortunately, this algorithm can only obtain the local optimum. In order to calculate $\overline{\lambda}$ efficiently, a helper function for $\overline{\lambda}$ is introduced:

$$L(\lambda, \overline{\lambda}) = \sum_{Q} \log[P(O, Q|\lambda)] P(O, Q|\overline{\lambda}). \tag{29}$$

The expectation of the joint distribution $\mathrm{P}(\mathrm{O}, \mathrm{Q}|\lambda)$ is based on the conditional probability $\mathrm{P}(\mathrm{O}, \mathrm{Q}|\overline{\lambda})$.

The log-likelihood function $\mathrm{f}(\lambda) = \log[\mathrm{P}(\mathrm{O}, \mathrm{Q}|\lambda)]$ is a concave function, according to Jensen's inequality:

$$f[E(\lambda)] = f[E(\overline{\lambda})] \geq E[f(\lambda)]. \tag{30}$$

Both sides of the above equation can be written as

$$\begin{aligned} E[f(\lambda)] &= \sum_{Q} \log[P(O, Q|\lambda)] P(O, Q|\overline{\lambda}) = L(\lambda, \overline{\lambda}), \\ f[E(\overline{\lambda})] &= \sum_{Q} \log[P(O, Q|\lambda)] P(O, Q|\overline{\lambda}) = \log\, P(O|\overline{\lambda}), \\ \log\, P(O|\overline{\lambda}) &\geq L(\lambda, \overline{\lambda}). \end{aligned} \tag{31}$$

No matter how $\overline{\lambda}$ changes, as long as $L(\lambda, \overline{\lambda})$ is increased, the infimum of $\mathrm{P}(\mathrm{O}|\lambda)$ can be increased, and then maximizing $L(\lambda, \overline{\lambda})$ can increase the probability of $\mathrm{P}(\mathrm{O}|\lambda)$, namely,

$$\max_{\overline{\lambda}} [L(\lambda, \overline{\lambda})] \Rightarrow P(O|\overline{\lambda}) \geq P(O|\lambda). \tag{32}$$

Therefore, iterative calculation can make $\mathrm{P}(\mathrm{O}|\lambda)$ converge to the maximum point.

$$P(O, Q|\lambda) = \pi_{q1} b_{q1}(o_1) a_{q1q2} b_{q2}(o_2) a_{q2q3} \cdots b_{qr}(o_r) a_{qr-1qr}. \tag{33}$$

$L(\lambda, \overline{\lambda})$ can be split into three items:

$$\begin{aligned} L(\lambda, \overline{\lambda}) &= \sum_{Q} \log[P(O, Q|\lambda)] P(O, Q|\overline{\lambda}), \\ L(\lambda, \overline{\lambda}) &= \sum_{Q} \log\, \pi_{q1} P(O, Q|\overline{\lambda}) \\ &\quad + \sum_{Q} \sum_{t=1}^{T-1} \log\, a_{qrqr+1} P(O, Q|\overline{\lambda}) \\ &\quad + \sum_{Q} \sum_{t=1}^{T} \log\, b_{qr}(o_t) P(O, Q|\overline{\lambda}). \end{aligned} \tag{34}$$

For solving the above problems, there are three natural constraints:

$$\begin{cases} \sum_{i=1}^{N} \overline{\pi_i} = 1, \\ \sum_{j=1}^{N} \overline{a_{ij}} = 1, 1 \leq i \leq N, \\ \sum_{k \Leftarrow 1}^{M} \overline{bj}(k) = 1, 1 \leq j \leq N. \end{cases} \tag{35}$$

According to the Lagrange multiplier method, combined with the above constraints, the extreme value of each component of $L(\lambda,\overline{\lambda})$ can be obtained, and the one-step optimal estimation of the HMM parameters can be obtained:

$$\overline{\pi}_i = \gamma_t(i),$$
$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}, \tag{36}$$
$$\overline{bj}(k) = \frac{\sum_{t=1,o_t=v_k}^{T-1}\gamma_t(i)}{\sum_{t=1}^{T-1}\gamma_t(i)}.$$

The new model parameter λ is obtained iteratively from the above. Once $\mathrm{P}(\mathrm{O}|\overline{\lambda}) \geq \mathrm{P}(\mathrm{O}|\lambda)$ is found, $\overline{\lambda}$ is assigned to λ, and the iterative calculation is performed until $\mathrm{P}(\mathrm{O}|\lambda)$ no longer changes significantly, and the final stone is used as the parameter estimation result of HMM.

The B-W algorithm cleverly uses the idea of maximum likelihood estimation to obtain the model parameter λ that maximizes $\mathrm{P}(\mathrm{O}|\lambda)$.

3.3. A Brief Introduction to Multidimensional Hidden Markov Models. For the n-dimensional mutually coupled observation sequence $\{O^{(1)},O^{(2)},\ldots,O^{(n)}\}$, it can be modeled as $\{\lambda^{(1)},\lambda^{(2)},\ldots,\lambda^{(n)}\}$, respectively:

$$P^{(1)}\left(O^{(1)},O^{(2)},\ldots,O^{(n)}\right) = P\left(O^{(l)}\right)\prod_{m=1,m\neq l}^{n} P\left(O^{(m)}|Q^{(l)}\right), \tag{37}$$

where $l,m = 1,2,\ldots,n$.

The n-dimensional HMM learning algorithm is

$$Q^{(l)} = ary_{Q^{(l)}}^{\max} P\left(O^{(l)},Q^{(l)}\right)\prod_{m=1,m\neq l}^{n} P\left(O^{(m)}|Q^{(l)}\right). \tag{38}$$

4. Experimental Results and Analysis

By selecting the gateway computer that is easy to be attacked, the system is connected to the signal and the ATS intranet at the same time, as the attack object of the experiment. Select DDoS as the main attack method, conduct vulnerability scanning, MS17-010 vulnerability attack, and DDoS attack. Based on the above data, relevant experimental analysis was carried out.

4.1. Situation Assessment Experiment Results and Analysis. By collecting the situational factor index data of gateway computer, CI, and ZC when the system is running normally, the maximum likelihood estimation method is used to fit the probability distribution of the data, and the K-S test is used to determine the distribution and parameters of the data. Then, according to the distribution of the data, the Lloyd–Max method is used to divide the data into states, and the state division interval with the minimum quantization error is obtained.

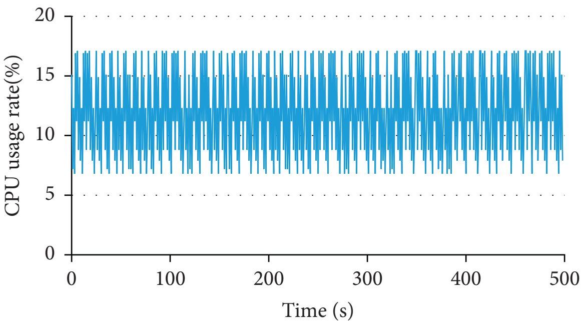

Figure 6: Change of CPU usage in the gateway computer.

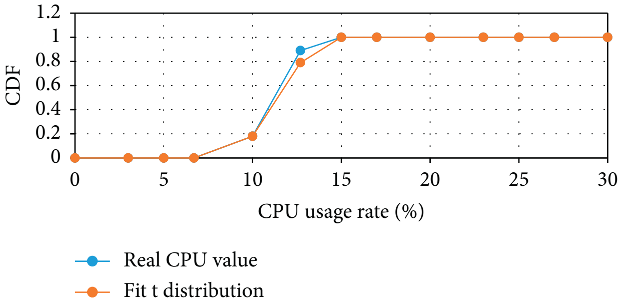

Figure 7: Comparison of data of gateway computer and CDF of fitting t distribution.

Taking the gateway computer as an example, the change and distribution of CPU usage within 1 hour are shown in Figures 6 and 7. The real data and the CDF curve of the fitted t distribution basically coincide, and the K-S test result is passed; it can be considered that the gateway computer CPU usage follows a t distribution with parameters $u = 10.6727$, $\sigma = 0.5594$, and $\nu = 4.3870$.

The change and distribution of RAM usage within 1 hour are shown in Figures 8 and 9. The real data and the CDF curve fitting the Gaussian distribution basically coincide, and the K-S test result is passed; it can be considered that the RAM usage of the gateway computer obeys the Gaussian distribution. The parameters are $u = 29.4752$ and $\sigma = 0.3447$.

Figures 10 and 11 show the change and distribution of the network sending rate within 1 hour. The real data basically coincides with the CDF curve fitting the t distribution, and has passed the K-S test. It can be considered that the gateway computer network sending rate obeys the t distribution. The parameters are $u = 16.1603$, $\sigma = 0.3524$, and $\nu = 3.7766$.

Figures 12 and 13 show the change and distribution of the network reception rate within 1 hour. The real data basically coincides with the CDF curve fitting the t distribution and has passed the K-S test. It can be considered that the gateway computer network reception rate obeys the t distribution. The parameters are $u = 21.5830$, $\sigma = 1.0182$, and $\nu = 4.3939$.

The distribution and distribution parameters of each index data of the gateway computer are summarized as shown in Table 1. The table lists the distribution and parameters of the gateway computer's CPU usage, RAM usage,

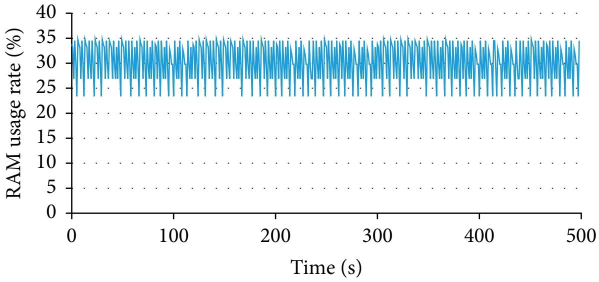

Figure 8: Change of RAM usage in the gateway computer.

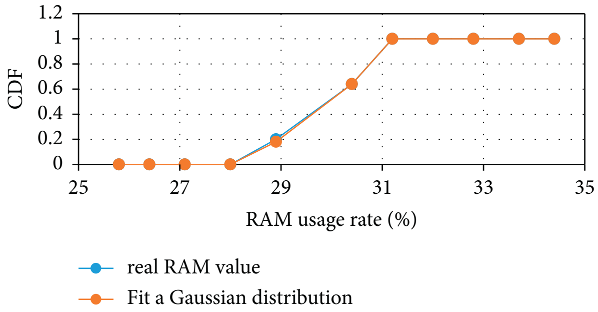

Figure 9: Comparison of RAM data of gateway computer and CDF of fitting Gaussian distribution.

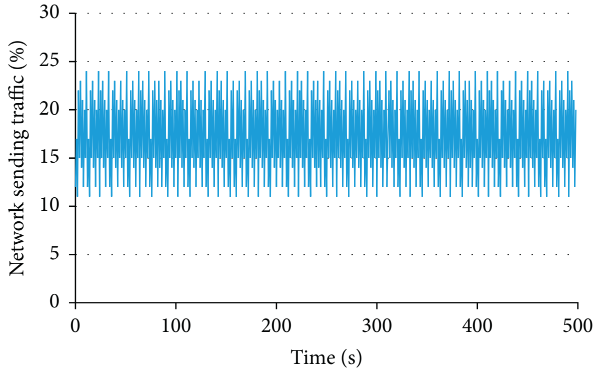

Figure 10: Change of transmission rate of the gateway computer network.

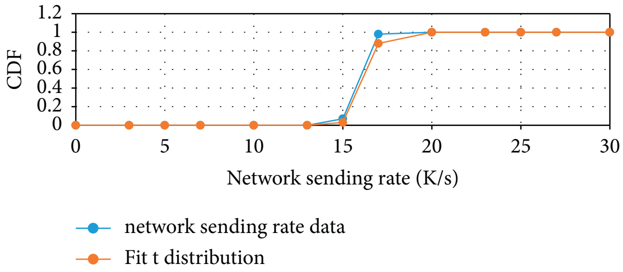

Figure 11: Comparison of network transmission rate of gateway computer and CDF of fitting Gaussian distribution.

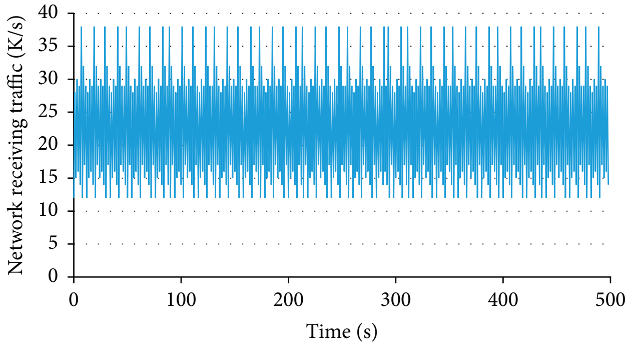

Figure 12: Change of receiving rate of the gateway computer network.

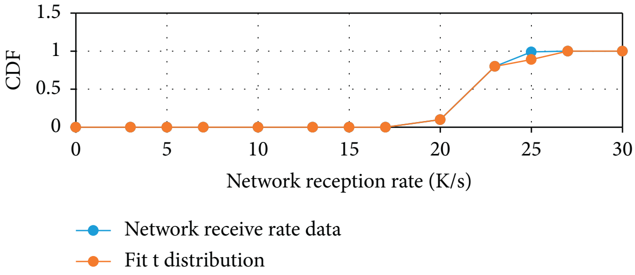

Figure 13: Comparison of receiving rate of gateway computer network and fitting t distribution.

network sending rate, and network receiving rate. Since the gateway computer is running, there is basically no data interaction with the disk, and the read rate and write rate of the disk are usually 0, so we will not do much research on it here.

After obtaining the distribution of each indicator data, the Lloyd–Max algorithm can be used to quantify the data. The quantification interval of each indicator is shown in Table 2.

In the same steps, the distribution and distribution parameters of each index of CI are obtained, as shown in Table 3. Since CI performs a large number of logical operations during the running process, it consumes a lot of CPU, and the average CPU usage rate reaches about 61%. However, the memory space occupied by the CI application software when running is smaller than that of the gateway application software, so the average RAM usage of the CI is slightly lower than that of the gateway computer.

The Lloyd–Max algorithm is used to quantify the data of each index of the CI, and the combined index is shown in Table 4.

Table 5 shows the distribution and distribution parameters of ZC index data. ZC simulation software does not have great hardware requirements, so the average of its CPU usage and RAM usage is at a low level.

The Lloyd–Max algorithm is used to quantify the data of various indicators of ZC, and the quantification interval of each indicator is shown in Table 6.

Table 1: Distribution and parameters of indicators of gateway computer.

Index	Obey the distribution	Parameter
CPU usage	t distribution	$u = 10.6727$, $\sigma = 0.5594$, $\nu = 4.3870$
RAM usage	Gaussian distribution	$u = 29.4752$, $\sigma = 0.3447$
Network sending rate	t distribution	$u = 16.1603$, $\sigma = 0.3524$, $\nu = 3.7766$
Network reception rate	t distribution	$u = 21.5830$, $\sigma = 1.0182$, $\nu = 4.3939$

Table 2: Quantitative interval of indicators of gateway computer.

Index		Quantization interval
CPU usage	0	10.2095 11.1813 12.8721 100.0000
RAM usage	0	27.1261 29.1398 32.0861 100.0000
Network sending rate	0	14.6692 15.9972 16.8667 30.0000
Network reception rate	0	19.4237 21.3072 22.9721 40.0000

Table 3: Distribution and parameters of indicators of CI.

Index	Obey the distribution	Parameter
CPU usage	t distribution	$u = 61.2727$, $\sigma = 0.7584$, $\nu = 5.0870$
RAM usage	Gaussian distribution	$u = 19.5752$, $\sigma = 0.5047$
Network sending rate	t distribution	$u = 8.3603$, $\sigma = 0.3124$, $\nu = 0.7766$
Network reception rate	t distribution	$u = 21.0120$, $\sigma = 0.7182$, $\nu = 0.8939$

Table 4: Quantitative interval of indicators of CI.

Index		Quantization interval
CPU usage	0	59.2095 59.1813 61.8721 100.0000
RAM usage	0	16.9261 19.1398 22.0861 100.0000
Network sending rate	0	6.66920 10.9972 17.8667 30.0000
Network reception rate	0	13.4237 19.3072 25.9721 40.0000

Table 5: Distribution and parameters of indicators of ZC.

Index	Obey the distribution	Parameter
CPU usage	t distribution	$u = 15.9727$, $\sigma = 0.2584$
RAM usage	Gaussian distribution	$u = 21.5752$, $\sigma = 0.1047$
Network sending rate	t distribution	$u = 9.3603$, $\sigma = 0.3124$, $\nu = 0.7726$
Network reception rate	t distribution	$u = 19.0120$, $\sigma = 1.3182$, $\nu = 0.9339$

Table 6: Quantitative interval of indicators of ZC.

Index		Quantization interval
CPU usage	0	14.2095 16.1813 19.8721 100.0000
RAM usage	0	18.9261 21.1398 24.8861 100.0000
Network sending rate	0	7.66920 12.9872 18.8667 30.0000
Network reception rate	0	13.4937 19.9072 25.9891 40.0000

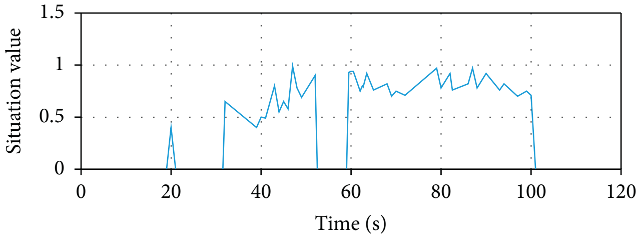

Figure 14: Change trend of situation value.

Figure 14 shows the change trend of the basic operation situation value of the system obtained by simulation. It can be seen from the figure that the change of situation value can be divided into three stages according to time. Around 20s, the situational value suddenly increased from 0 to 0.4, which is consistent with the vulnerability scanning event conducted by the attacker at this time. The posture value then returned to 0, indicating that the attacker had been scanning for vulnerabilities for a short period of time and had no other activity for a period of time. During the period of 34s to 54s, the situation value fluctuated between 0.4 and 0.9, and there was an obvious upward trend. During this period, the attacker used the MS-17010 vulnerability to attack the system and obtained the administrators of the devices in the system

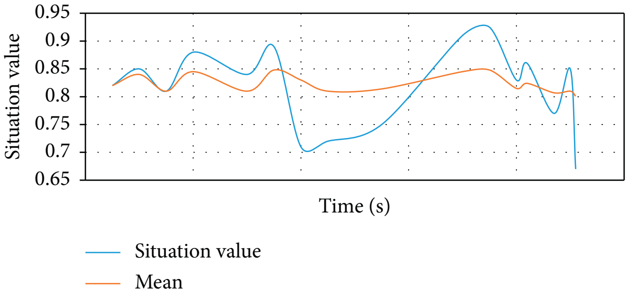

Figure 15: The training sample set.

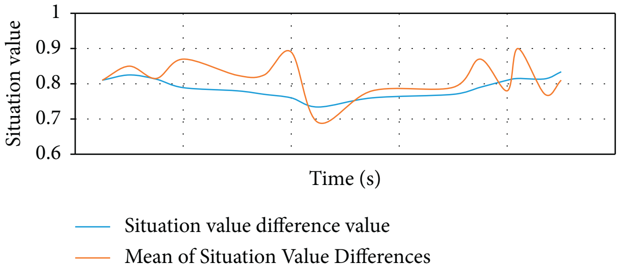

Figure 16: The result of 1 difference of situation values.

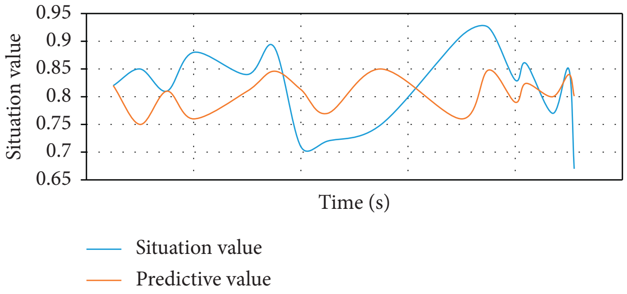

Figure 17: The result of the situation value prediction.

(permission), implanted the DDoS virus on the device, and then stopped activities for a period of time, and the situation value also returned to 0. During the period of 60s to 100s, the situation value fluctuates around 0.8, which is consistent with the event that the attackers control the devices of the system to launch DDoS attacks on the gateway computer. The attacker then ceased activity, and the situational value returned to 0.

The experimental results show that the fluctuation of the situation value corresponds to the different attack behaviors carried out by the attacker, which accurately describes the information security status of the system and verifies the effectiveness and accuracy of the situation awareness method proposed in this paper. And the time node of the change of the situation value is synchronized with the time node of the attacker's attack, which also verifies the real-time performance of the situation assessment method.

4.2. Situation Forecasting Experiment Results and Analysis. This section selects the trend of situation change from 60s to 70s and uses the first 20 data points as training samples and the last 10 data points as comparison samples to verify the situation prediction method. The selected data is shown in Figure 15. Using ADF to check that the situation sequence is not a stationary sequence, the sequence needs to be stationary.

The selected situation sequence is differentiated once, and the obtained difference sequence passes the ADF stationarity test. Therefore, the sequence difference is stopped, and the subsequent steps of ARIMA can be used to predict the situation sequence. The difference result is shown in Figure 16.

The training samples are fitted by the ARIMA method [22]. Determine the model parameters of ARIMA as $p = 3$, $d = 1$, $q = 4$. The parameters in the formula are obtained by using the maximum likelihood estimation method: the constant term $\mu = -0.026513$; the three autoregressive coefficients are $\phi_1 = -0.20651$, $\phi_2 = -0.33681$, and $\phi_3 = -0.4685$; the four moving average coefficients are $\theta_1 = -0.54385$, $\theta_2 = 0.24325$, $\theta_3 = -0.78119$, and $\theta_4 = 0.081789$. Figure 17 shows the situation prediction result using this estimation model. It can be seen from the figure that the predicted value is basically consistent with the trend of the actual value. This shows that the ARIMA model has a high prediction accuracy for the information security situation value of the train control system.

The experimental results show that the situation prediction method based on ARIMA can effectively predict the short-term changes of the situation value, but because the ARIMA method produces the characteristic that the sequence tends to be stable, the predicted value will converge near the mean value of the real value, which cannot reflect the fluctuation of the situation value. It is suitable for short-term forecasts with high accuracy requirements.

5. Conclusion

Information security is an issue that every country should pay attention to. The article is short and mainly explains the use of the hidden Markov model. After simulating the information security experiment, the feasibility and effectiveness of the situation assessment method and the situation prediction method are carried out. After a brief verification, the results also show that the situation assessment method based on the hidden Markov model can effectively assess the level of information security. However, with the passage of time, this method will gradually converge to the mean value of the situation value sequence, so it cannot reflect the fluctuation of the situation. Therefore, the long-term prediction of the situation value can become a follow-up research direction. This article only provides a superficial understanding of security situational awareness and ignores some details in the analysis process. It is hoped that the article can provide ideas for future situational awareness research in the future.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding this work.

References

[1] A. Mcafee and E. Brynjolfsson, “Big data: the management revolution,” *Harvard Business Review*, vol. 90, no. 10, pp. 60–128, 2012.

[2] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222–233, 2014.

[3] A. R. Palin and I. J. Jacob, “Review on fog based spectrum sensing for artificial intelligence,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 3, pp. 66–70, 2018.

[4] Y. Lv, Y. Duan, W. Kang, Z. X. Li, and F. Y. Wang, “Traffic flow prediction with big data: a deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[5] A. Ibrahim, T. Hashem, I. Yaqoob, Nor, and B. Anuar, “The rise of “big data” on cloud computing: review and open research issues,” *Information Systems*, vol. 47, pp. 98–115, 2015.

[6] M. D. Fethi and F. Pasiouras, “Assessing bank efficiency and performance with operational research and artificial intelligence techniques: a survey,” *European Journal of Operational Research*, vol. 204, no. 2, pp. 189–198, 2010.

[7] L. Steels, “The artificial life roots of artificial intelligence,” *Artificial Life*, vol. 1, no. 1_2, pp. 75–110, 2010.

[8] O. Kisi, J. Shiri, and B. Nikoofar, “Forecasting daily lake levels using artificial intelligence approaches,” *Computers & Geosciences*, vol. 41, pp. 169–180, 2012.

[9] F. Ofli, P. Meier, M. Imran et al., “Combining human computing and machine learning to make sense of big (aerial) data for disaster response,” *Big Data*, vol. 4, no. 1, pp. 47–59, 2016.

[10] A. Vega-Muoz, “Toward an integrated disaster management approach: how artificial intelligence can boost disaster management,” *Sustainability*, vol. 13, 2021.

[11] Z. Gu, R. Wang, and S. Wang, “Research ON real-time forecast OF security posture OF information system based ON improved grey-Markov chain,” *Computer Applications and Software*, vol. 34, no. 8, pp. 1–8, 2017.

[12] D. Waltermire and D. Harrington, “Endpoint security posture assessment: enterprise use cases,” *University of Twente Nikos*, vol. 51, no. 02, pp. 83–102, 2015.

[13] Y. J. Liu and H. U. Rong, “Research on the information security situational awareness system based on big data and artificial intelligence technology,” *Management & Technology of SME*, vol. 10, no. 8, pp. 1–8, 2019.

[14] X. Zhang and D. Yang, “Research on music assisted teaching system based on artificial intelligence technology,” *Journal of Physics: Conference Series*, vol. 1852, no. 2, Article ID 022032, 2021.

[15] Z. Zhang and L. I. Fang, “Research ON the water pollution monitoring and rapid decision-making system based ON artificial intelligence agent,” *Journal of Environmental Protection and Ecology*, vol. 20, no. 3, pp. 1565–1573, 2019.

[16] Y. Wang, W. Li, and Y. Liu, “A forecast method for network security situation based on fuzzy Markov chain,” *Lecture Notes in Electrical Engineering*, vol. 260, pp. 953–962, 2014.

[17] G. Chen and S. Li, “Network on chip for enterprise information management and integration in intelligent physical systems,” *Enterprise Information Systems*, vol. 15, no. 7, pp. 935–950, 2021.

[18] H. Xiong, “Application of artificial intelligence in computer network technology in the era of big data,” *Information and Computer (Theoretical Edition)*, vol. 31, no. 21, pp. 128-129+132, 2019.

[19] X. Xiao, L. Chun, K. Peng et al., “Research review of security situation prediction technology based on artificial intelligence,” *Information Security Research*, vol. 6, no. 06, pp. 506–513, 2020.

[20] Y. Duan, R. Lin, X. Liu, Z. Xue, and Y. Shi, “Research on the modeling of financial information security indicators based on threat intelligence,” *Information & Technology*, vol. 25, no. 06, pp. 1–6, 2018.

[21] J. Xiong, S. Wu, C. Peng, and Y. Tian, Eds., *Mobile Multimedia Communications. MobiMedia 2021*, Springer, vol. 394, Cham, 2021.

[22] M. Bhanu, J. Mendes-Moreira, and J. Chandra, “Embedding traffic network characteristics using tensor for improved traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 99, pp. 1–13, 2020.