

A parallel game model-based intrusion response system for cross-layer security in industrial internet of things

Siyang Yu^{1,2}  | Fan Wu²  | Baoding Chen² | Ronghui Cao³ | Zhibang Yang² | Keqin Li^{2,4}

¹College of Information Technology and Management, Hunan University of Finance and Economics, Changsha, China

²College of Computer Science and Engineering, Hunan University, Changsha, Hunan, China

³College of Computer and Communication Engineering, Changsha University of Science Technology, Changsha, Hunan, China

⁴Department of Computer Science, State University of New York, New York, USA

Correspondence

Fan Wu, College of Computer Science and Engineering, Hunan University, Changsha, Hunan 410008, China.
Email: wufan@hnu.edu.cn

Funding information

National Key R&D Program of China, Grant/Award Number: 2020YFB2104000; Key Area Research Program of Hunan, Grant/Award Number: 2019GK2091; Program of National Natural Science Foundation of China, Grant/Award Number: 62002114; Program of Natural Science Foundation of Hunan, Grant/Award Numbers: 2021JJ40108, 2023JJ30102; Program of Natural Science Foundation of Changsha, Grant/Award Number: kq2202318; NSFC, Grant/Award Number: 61802032

Summary

With the rise of industrialization, the importance of the industrial Internet of Things (IIoT) has increased significantly, and with it comes a variety of security threats. Therefore, the security of these networks is critical. Industrial Response Systems (IRSs), as the last line of security, plays an important role in the security system of the Industrial Internet of Things. In this paper, a new IRS model based on the non-cooperative game is proposed. First, by combining the Partially Observable Markov Decision Process (POMDP) model with the stochastic game model based on the expanded attack tree, our model could effectively perceive the changes at each node. Second, our model incorporates the alarms of intrusion detection system (IDS) and the physical quantities of sensors in Industrial Cyber-Physical System (ICPS) into the quantization system so that the model can respond to intruders more accurately and comprehensively. Finally, we develop this model based on multiprocessors to speed up the solution process, and adopt an approximation algorithm to reduce the number of iterations of the POMDP

KEYWORDS

approximation algorithm, parallel, partially observable Markov decision, stochastic games

1 | INTRODUCTION

The IIoT is an extension of the Internet of Things (IoT) that integrates the communication, collection and processing processes of industrial data and implements them in the real-time industrial network. The IIoT combines IoT devices with traditional industrial control system (ICS) to create a system that is more efficient than traditional industrial control systems. In this regard, the supervisory control and data acquisition (SCADA) system as the largest subset of the industrial control system, plays a key role in IIoT.^{1,2} Therefore, the number of attacks they face is increasing day by day. Among these attacks, network intrusion as the most common and effective attack type, is an urgent need for appropriate measures.^{3,4}

In general, intrusion processing methods can be divided into three categories: The first is intrusion prediction and prevention, the second is intrusion detection system (IDS), which attempts to detect abnormal or illegal network activity in daily network communications, such as packet payload format detection technology to identify attacks with error instructions.⁵ Finally, as the last safeguard of the system security, the IRSs plays an important role in the overall defense system, which can respond to the intruders in a timely manner and minimize the loss of the attacked system. So IRSs has been widely studied⁶ and applied to various network environments⁷⁻¹⁰ in recent years.

However, most of the studies cannot be directly applied to the IIoT because the IIoT not only involves traditional network data, but also is based on various types of CPS¹¹ and the CPS contains a large number of physical parameters represented by sensors. And most of these studies took information layer data as the basis of the model.¹²

With this background, in this paper, we design a IRS model called Low-Cost Intrusion Response System (LIRS), which unifies the two types of data in IIoT and uses them as the data source of our model. In order to recreate the invasion response scenario in the real industrial environment as much as possible, we model the interaction between the attacker and the defender and abstract them into a non-cooperative Starkberg stochastic game model, which has been demonstrated to be feasible in the paper written by Zonouz et al.¹³ In our model, we merge the expanded attack tree with the actual scene node graph to form an expanded attack tree that can provide a formal expression for subsequent model construction and integrates the alarms in IDS with the security score of each node in the structure. To reduce the impact of false positive rate and false negative rate of information layer data caused by the use of IDS alarm, we apply POMDP to the calculation of the payoff matrix, which is the core process of the game-theoretic model. In this model, the system analyzes its best behavior by solving the decision model every time. In this way, we can minimize the maximum damage to the system caused by the next intrusion behavior of the attacker. Also, the system changes its value model over time, which reduces the cost of response behavior at a later time.

In the following, this paper improves the existing intrusion response technology from three aspects:

- We evaluate the risk index of the system state according to the physical quantity of CPS components, the CVSS score, and the IDS alarm during the process of decision-making, which improves the adaptability and prediction accuracy of the evaluation system.
- We conduct modeling using the combination of the POMDP and the game-theoretic model using an expanded attack tree so that we can assess the system's attack in real time and make accurate decisions for it. In addition, this model is more compatible to quantification of other types of indicators.
- The parallel method is used to speed up the operation of the game, and the approximate method is added to the operations solution process of the Markov decision to reduce the number of iterations.

The remainder of the paper is organized as follows. Section 2 reviews the related work. The response engine and game model are presented in Section 3. Section 4 analyzes the experimental results and shows the performance of the algorithms. The conclusion is summarized in Section 5.

2 | RELATED WORK

With the increasing importance of IIoT, the complexity and vulnerability of industrial networks become evident, so that the system's response to intrusions becomes particularly important,⁶ which has led to the development of various IRSs in recent years.^{14,15} Li et al.¹⁶ gave an approach to dynamic decision making for intrusion response that uses the Pareto-optimal set to deal with it. In Reference 17, Sharif Ullah et al. formulated the interactions between the attacker and the defender as an attack graph and an object instance graph, and they discouraged further intrusions by increasing the uncertainty of subsequent attacks and the cost of time and space by the attacker. However, when modeling the defense strategy, these studies only considered the maximum gain under the current state faced by the defender and ignored that the process of intrusion response is a process of constant interaction between the attacker and the defender, which causes the model to fall into the locally optimal solution.

As a common modeling method of the non-cooperative behavioral process of multiple individuals, the stochastic game-theoretic model has been widely used in IDS and ICS^{18,19} in recent years. In the field of decision-making based on revenue, Liu Gang et al. provided a new iterative proximal algorithm to solve asset pricing in cloud computing.²⁰ Xingshuo et al.²¹ constructed a differential game to reduce the frequency of intrusion. Zonouz et al.¹³ proposed a game model using an observable competitive Markov decision process (POCMDP) to reduce system vulnerability. However, in other research directions, researchers have conducted various in-depth studies on the influence of physical quantities. However, the data sources used in these articles are all from information layer data, such as IDS alarm and traffic packet information. When applied to IIoT, the result may fail to obtain the optimal solution due to incomplete input data coverage, and the decisions they took may become patchiness due to the neglect of the influence of physical quantities in the system.

Aiming at the improvement of IRS in IIoT, some researchers have implemented new models by combining CPS features with traditional network features. Orojloo et al.²² built a model to evaluate the security of CPS consisting of the probability of being attacked and the time-to-shutdown(TTSD) of CPS.²³ Chong Wang et al.²⁴ found approximate dynamic programming to appease the MDP model they built, which is driven by post-decision states and the forward dynamics algorithm. Xuan Zhao et al.²⁵ combined network traffic packets with physical layer parameters, such as CPU utilization rate and memory utilization rate of mobile phone, to analyze and predict the behavior of mobile phone users. All these works have established the mechanism of integrating the physical quantity in the industrial system with the information in the information layer, and combined with other decision models on this basis to carry out decision simulation and message prediction. However, when they use physical quantities, they do not carry out error filtering first. If such methods are introduced to IRS under IIoT or ICS environment, the noise and errors in the physical quantities may lead to misjudgments in the subsequent decision-making process.

Compared with other existing models, the new IRS's decision-making model based on game-theoretic and POMDP in this paper can effectively deal with various types of intrusion behaviors in IIoT. It unifies the information layer data and physical layer data of sensors, and then abstracts the

attack degree of each independent node and functional department to form an expanded attack tree. After the decision simulation of the parallel game model based on POMDP, it finally forms the intrusion response decision of the defender under a certain state.

3 | RELATED CONCEPTS

Definition 1 (Stochastic game). A repeated game model with probabilistic transitions. In this model, players repeatedly choose actions depending on their current state and the action. This model can be described as a seven-element group:

$$G = \langle GP, S, A, D, P, \pi, R \rangle. \quad (1)$$

In Equation (1), GP represents the players, S for the system state, A represents the attacker's action set, D represents the defender's action set, P represents the transition probability matrix, π represents the policy set of the game participants, and R represents the income matrix of the players in the game.

We consider a network intrusion as an attack and defense process and model this process as a stochastic game model. In this model, the attacker can gain by a specific attack behavior, and the defender resists the attack by taking countermeasures to reduce the losses suffered. In addition, the model enters a certain state based on probability after the attacking work and the defending action are completed. Take the Prisoners Dilemma as an example, as shown in Table 1. From the table, it can be seen that the best outcome for both prisoners should be not to confess, the number of years in prison is the shortest, but here there is a very stable dominant strategy for both sides—confession, because confession can avoid their own worst situation, regardless of the other party to take the attitude of their own confession is the best, but the results are for both sides and its bad. In fact, this is a maximum-minimum strategy. In this case, the conflict is caused by an uncertainty in the actions of both sides. Therefore, we need to calculate the strategy probability of both sides in each state, the corresponding gain and the probability of state transfer in the stochastic game model, because the attacker and the defender do not know what actions the other side will take due to incomplete information.

Definition 2 (POMDP). The POMDP is a generalization of a Markov decision process and provides a method for modeling a decision process in which the decision makers cannot directly obtain the state variable representing the current state of the system. Typically, a POMDP model can also be divided into a six-element group $\langle S, A, Y, B, P, Q, R \rangle$. Here we only explain the members, which are different from those of the stochastic game: Y represents the set of observable state quantities, B represents the belief state space, Q represents the conditional observation probability matrix.

Definition 3 (linear time-invariant, LTI). A system, due the constraints of linearity and time-invariance, can produce output signals from any input signal subject. Generally, this system can be described by the following equation:²⁶

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{v}_k. \end{aligned} \quad (2)$$

Definition 4 (probability calculation of attack tree nodes). Since an attack tree is determined from the bottom up, the attack probability of the entire tree is determined after determining the probability of all leaf nodes. Under the influence of the logic gate, the attack probability h of a non-leaf node can be calculated by the following iterative calculation:

$$\delta(q) = \begin{cases} \prod_{i \in I} \delta(i), & \text{if } q \text{ is an AND gate,} \\ 1 - \prod_{i \in I} (1 - \delta(i)), & \text{otherwise} \end{cases}. \quad (3)$$

In this equation, $i \in I$ represents the child node of node q , and $\delta(i)$ is the probability of being attacked by child node i .

TABLE 1 The Prisoners Dilemma.

Number of years P_A	P_B	
	Frankly	No confession
Frankly	-6, -6	0, -10
No confession	0, -10	-1, -1

4 | LOW-COST INTRUSION RESPONSE SYSTEM

An industrial control system, as a system that simultaneously contains information space and physical space and aims to monitor and manage one or more industrial production tasks, often faces interference and invasion from information space and physical space simultaneously. The nature of the industrial environment determines that the information source of the intrusion response system in the industrial control system is diverse, and the data is mixed and difficult to handle. Secondly, the complexity of the industrial control system and the interaction of the intrusion response determine that the influence of the defense strategy on the industrial environment and the behavior of the attacker should be fully considered when deciding the intrusion response in a more complex industrial control system. To solve these problems, the basic idea of this paper is to integrate the alarm data and the physical quantities of the industrial sensors of the intrusion detection system into a new data system and optimize the non-cooperative stochastic game model based on the improved partially observable Markov decision process.

4.1 | Data node cross-layer risk assessment system

The scene of this paper is established in a system where part of the node alarm and part of the physical quantities associated with the physical equipment can be observed so that the extension tree node is formed by part of the observable node plus another part of the unobservable result node. Among the data sets sampled in a power plant attack and defense drill, the numbers of these two messages (IDS alarms and equipment monitoring alarms) are given in Table 2.

In the above data, the sensor errors usually occur in the floodgate control system, voltage conversion system and other systems that cannot obtain states directly through the information layer.⁷ In this case, the risk coefficient evaluation mechanism for a single data set is unable to evaluate the state of system comprehensively and accurately. Therefore in this case, we will integrate the two evaluation systems into our new evaluation system.

4.1.1 | Network layer node risk assessment model

The data source for the risk assessment of the network layer nodes is the alarm set IDS. The related logs before and after each IDS-alarm time are counted in the audit log historical data, and a binary set is created with the IDS-alarm according to the event information obtained from the association relation of the log events. Finally, the binary set between IDS alarm and events can be obtained. At this time, Bayesian equation is used to calculate the node risk degree for the set.

$$\delta(I|O_l^i) = \frac{P(I) \cdot \prod_{o_j \in O_l^i} P(o_j|I)}{P(I) \cdot \prod_{o_j \in O_l^i} P(o_j|I) + P(\bar{I}) \cdot \prod_{o_j \in O_l^i} P(o_j|\bar{I})}. \quad (4)$$

The $I \in L$ in Equation (4) represents leaf node I . If it receives an alarm, the value is set to 1; and if it is normal, the value is set to 0. $o_j \in O_l^i$ represents all IDS alarms received by node I , and O represents the set of IDS alarms that the whole system can receive. The $\delta(I|O_l^i)$ indicates the probability of the corresponding event I of the node when the IDS alarm is triggered, where O_l^i represents the different types of IDS alarms in the node, and $P(I)$ represents the prior probability of the event I . $P(o_j|I)$ represents the probability of receiving an alert from o_j in the event of I . This parameter can be calculated by the conditional probability equation:

$$P(o_j|I) = \frac{P(o_j, I)}{P(I)}. \quad (5)$$

In Equation (5), the probabilities of $P(I)$ and $P(o_j|I)$ are derived from the statistics of the historical logs. When there is little historical data, $P(I)$ can be replaced by an a priori probability using the CVSS-corrected score described, while $P(o_j|I)$ in Equation (5) specifies a dataset subject to a constant probability distribution when there is little historical data. During the computation, the logged historical data is added to the dataset for operation, and the dataset is removed after the amount of historical data reaches a fixed threshold. Introducing the result of Equation (4) into the network layer risk estimation can take into account the node risk estimation error and reduce the calculation error caused by a certain degree of error.

TABLE 2 The number of value of two kinds.

Value name	Number of values
IDS alarms	3954
Sensor value	834

4.1.2 | ICPS risk assessment model

In industrial control network, the information collected by various sensors about physical quantities and status of industrial equipment is very important, so they often become the target of attackers. There are attacks on sensors in industrial networks that corrupt sensor parameters or transmit false signals for physical quantity information and state information after maliciously controlling industrial equipment. The system can model the industrial field devices, sensors, and detectors that can independently complete an industrial process as an ICPS. Therefore, the problem in this section is transformed into building a risk assessment model for ICPS. In addition to creating a model to calculate coefficients, this model must also create an estimator for sensor information and device status information of physical layer and control layer devices.

First, the attacked discrete state of ICPS is modeled using the random hybrid model and the properties of the linear time-invariant system,^{27,28} and then the continuous process system in a certain discrete state is modeled in the random hybrid model using the iterative equation of the linear time-invariant system.

Discrete state analysis of ICPS. It is assumed that a physical information system has x controllers and Y sensors, and each component has a normal state and an attacked state. At this time, the number of discrete states of the entire attacked system is in the current state:

$$|X_D| = \sum_{i=0}^{x+y} \frac{(x+y)!}{(x+y-i)!}. \quad (6)$$

In any discrete state, the system is a linear time-invariant continuous system. At this point, the properties of the random hybrid model are brought into the linear time-invariant system, and the equation can be obtained by Equation (2).

$$\begin{aligned} x_{k+1} &= Ax_k + B(u_k + ACK_k^{u,s^k}) + w_k \\ y_k &= Cx_k + ACK_k^{y,s^k} + v_k. \end{aligned} \quad (7)$$

To avoid confusion with the system state vector x , x_d is replaced by s . In Equation (7), ACK_k^{u,s^k} represents the attacker's attack signal on the actuator in the discrete state s_k . For the concealment and destructibility of the attack, the attacker will determine the type of attack signal according to the system characteristics. Since this system is a linear time invariant system, most of the attacks against this type of system are additive attack signals. In this model, a discrete state transition probability matrix Tr is created based on the success rate of the attack for each component involved, that is, $Tr_{ij}^j = p(s^{k+1} = s^j | s^k = s^i)$. The sum of each row and column of the matrix are identical to 1.

Second, Assuming that the system is in a certain state $s^j = x_d^k \in X_D$, x_{k+1} in Equation (7) can be expressed as:

$$x_{k+1} = A^{s^j} x_k + B^{s^j} (u_k + \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ u_k^{s^j} \end{bmatrix}) + w_k = A^{s^j} x_k + B^{s^j} u_k + B_a^{s^j} u_k^{s^j} + w_k. \quad (8)$$

In Equation (8), the rank of the unit submatrix I is the number of components of the attacker's attack signal ACK_k^{u,s^k} . To facilitate the following computation, this section copies the additive signal owned by ACK_k^{u,s^k} in the control vector u_k^d , and the other component values are 0 by default, forming $u_k^{s^j}$, and $B_a^{s^j} = B^{s^j} \begin{bmatrix} 0 \\ u_k^{s^j} \end{bmatrix}$ at this time.

For the sensor vector y_k , its reading will be inaccurate after being attacked by the additive semaphore. Since this attack semaphore has nothing to do with the system state variable, the attack signal will be added directly to y_k . Now, assuming that $y_k^{s^j} = y_k - ACK_k^{y,s^k}$, y_k in Equation (7) can be expressed as:

$$y_k^{s^j} = C^{s^j} x_k + v_k. \quad (9)$$

Combined with the recursive state estimator in Reference 29 to form a new state estimator, as shown in Equation (10):

$$\begin{aligned} \hat{x}_{k|k-1} &= A_{k-1} \hat{x}_{k-1|k-1} \\ \hat{u}_{k-1} &= M_k (y_k - C_k \hat{x}_{k|k-1}) \\ \hat{x}_{k|k}^* &= \hat{x}_{k|k-1} + B_{k-1} \hat{u}_{k-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k}^* + K_k (y_k - C_k \hat{x}_{k|k}^*). \end{aligned} \quad (10)$$

In Equation (10), if $\hat{x}_{k-1|k-1}$ is an unbiased estimate of x_{k-1} , then $\hat{x}_{k|k-1}$ is offset because the input of the real system is unknown. Therefore, the unbiased estimate of the unknown input must be estimated by measuring the second equation of the group of Equation (10), and the estimate of the control amount of the second equation can effectively eliminate the possible deviation of the control amount at $k - 1$ and use it in the third equation

to obtain the unbiased estimate $\hat{x}_{k|k}^*$ of x_k . In the last equation, an update similar to the Kalman filter is used to minimize the variance of the unbiased estimator obtained by the set of Equation (10) and obtain the final result $\hat{x}_{k|k}$.

M_k represents the estimator for the unknown input. According to the literature,³⁰ this matrix can be expressed as:

$$M_k = (F_k^T \bar{R}_k^{-1} F_k)^{-1} F_k^T \bar{R}_k^{-1}. \quad (11)$$

In Equation (11), $F_k = C_k B_{k-1}$, $\bar{R} = \mathbb{E}[e_k e_k^T = C_k P_{k|k-1} C_k^T + R_k]$, where e_k is the estimated error between the sensor vector estimate \tilde{y}_k and the actual vector y_k , where R_k satisfies the condition $R_k = E[v_k v_k^T]$. And K_k represents the income matrix that minimizes the variance of the system state estimator.

$$\begin{aligned} P_{k|k-1} &= A_{k-1} P_{k-1|k-1} A_{k-1}^T + Q_{k-1} \\ \bar{R}_k^* &= (I_p - C_k B_{k-1} M_k) \bar{R}_k (I_p - C_k B_{k-1} M_k)^T \\ S_k^* &= (I_p - C_k B_{k-1} M_k) S_k (I_p - C_k B_{k-1} M_k)^T \\ P_{k|k}^* &= (I_n - B_{k-1} M_k C_k) P_{k|k-1} (I_n - B_{k-1} M_k C_k)^T + B_{k-1} M_k R_k M_k^T B_{k-1}^T \\ K_k &= (P_{k|k}^* C_k^T + S_k^*) a_k^T (a_k \bar{R}_k^* a_k^T)^{-1} a_k. \end{aligned} \quad (12)$$

$S_k \in R^{p \times p}$ in Equation (12) represents a random irreversible matrix, a_k represents an arbitrary $rank(\bar{R}_k^*) \times m$, and satisfies $a_k \bar{R}_k^* a_k^T$ is a full rank matrix. In Equation (12) $Q_{k-1} = E[w_{k-1} w_{k-1}^T]$. Put the estimators M_k and K_k into Equation (10), the $\hat{x}_{k|k}$ is the unbiased estimator. At this point, $y_k^{s^i}$ in Equation (9) is the sensor unbiased estimator of the current state.

When the ICPS system is in a discrete state at time k , the participation of the system in the risk coefficient calculation is best represented by the unbiased estimator of the system state variable $\hat{x}_{k|k}$ and the unbiased estimate of the sensor variable in the system. The range of values of this type of data is uncertain. It cannot be directly processed by the normalization function. In this section, the extraneous state variable system MTTR (Mean-Time-To Recovery /Repair, MTTR) is introduced to quantify the risks represented by these estimates. MTTR is the fundamental measure of project reparability. In a fixed cyber-physical system, the maximum value it can reach in a given state of MTTR is fixed. Assuming that the maximum value is M_{max} , and using historical data to perform function fitting processing for the relationship between the physical size group and MTTR, and using the fitted function to solve the MTTR in a certain state s^i , assuming that the computation result is m , then the risk value of the node is $risk_{s^i} = \frac{m}{M_{max}}$.

4.2 | Game model

Based on the POMDP method for policy decisions, this chapter proposes a IRS policy resolution framework, which is accelerated and optimized based on this framework to achieve better performance of the resolution framework.

4.2.1 | Intrusion response model construction

In an industrial control network, there are communication-blocking devices such as gatekeepers/industrial gateways between different layers. Such devices require that specially structured data packets of adjacent layers be allowed to pass. Under this assumption, this paper considers the different layers as one of the systems. Independent subsystems, each subsystem state has an initial state and a final state. Based on the risk assessment system described in Section 4.1, the reward function of the game model is:

$$R(s, a) = (abs(\delta_g(s) - \delta_g(s')))^{C_1} T^{C_2}. \quad (13)$$

In the game process, the execution of each operation involves an effective time cost. At the same time, the time costs of both sides of the game are integrated into the risk evaluation system to obtain Equation (13). In Equation (13), T is the time period in which the strategy takes effect, which can also be called the time cost. C_1 and C_2 are normal numbers. In this equation, the probability difference of the final state before and after the behavior is evaluated and the difference is used as part of the income function. Assuming that the current subsystem has n subnodes, the equation for the attainment probability of the final state of the current subsystem is Equation (3).

Depending on the different adjacent states of the logic gates between the subsystems, there are 1 or $2^n - 1$ possibilities for the final state. In the industrial setting, when a logic level contains multiple nodes, a substructure similar to the attack tree is formed among the nodes distributed in the subsystem. When q is the root node, the attack tree structure of node q described by Equation (3) is shown in Figure 1.

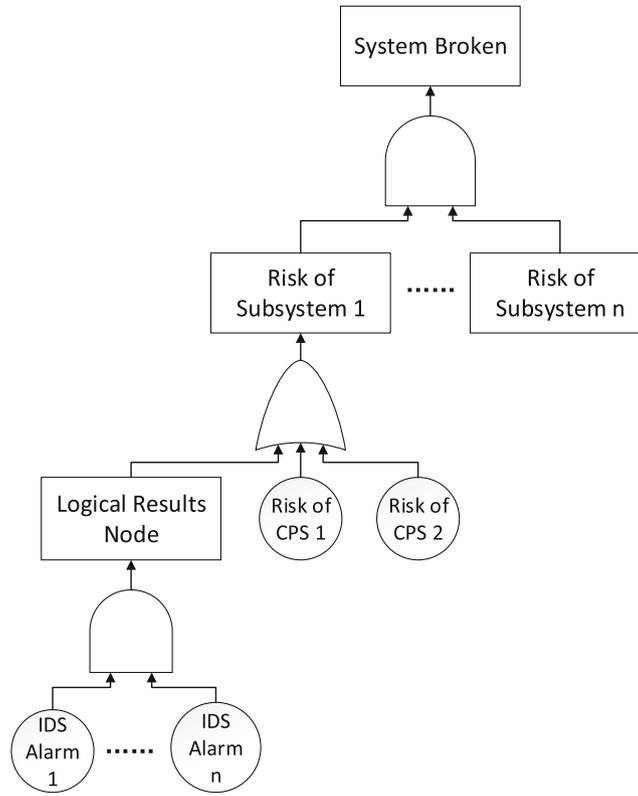


FIGURE 1 The schematic diagram of attack tree structure.

Equation (3) also holds for other nodes belonging to the same root node. In a given state s , the probability that the POMDP game model takes belief state B can be calculated using Equation (14).

$$b_j^+ = \text{Prob}(S^+ = s_j | Y = y_k, S = s_h, A = a_i, b) = \frac{\sum_h Q_{hik} P_{hij} b_h}{\sum_h Q_{hik} b_h}. \quad (14)$$

b in Equation (14) represents the probability of being of each belief state. We describe it using the following equations:

$$b(s) = \prod_{n \in \mathcal{N}} \left(1_{[s_i=1]} \cdot \delta(l) + 1_{[s_i=0]} \cdot (1 - \delta(l)) \right). \quad (15)$$

In Equation (15), the $[s_i = 1]$ represents the signum function. We use it to different nodes with different values. And P in Equation (14) represents the probability of transition between states, which is determined by the initial state transfer matrix, IDS alarm and physical node risk assessment index. We can calculate this using the following formula:

$$P_{hij} = \text{Prob}(S^+ = s_j | S = s_h, A = a_i) = \begin{cases} p_{hj}^{\text{init}} * (1 - p_i^{\text{succ}}) + p_i^{\text{succ}}, & \text{if } a_i \text{ end state is } j \\ p_{hj}^{\text{init}} * (1 - p_i^{\text{succ}}), & \text{otherwise} \end{cases}. \quad (16)$$

In this formula, $a_i \in A_h$ represent a possible action that starts in state h .

Actually, after introducing the belief state space, the POMDP problem can be transformed into a Markov chain problem based on the belief state space to solve it.³¹ Through the introduction of the belief state space, the POMDP problem can be regarded as a Belief MDP problem. Accordingly, the method to solve the MDP can also be introduced into this game model in this paper. Therefore, in this paper, we use the P_b , R_b matrix to transform the POMDP into a MDP solution to simplify the solution process.

Algorithm 1. The generation algorithm of belief state transition matrix

Require:

- D : leaf node data set of IDS alarms;
- $D2$: set of physical signals for some components;
- S : state transition matrix;
- A, B, C : the given matrices of components;

Ensure:

- Pb : A matrix which stores the transition probability between belief states;
- Rb : A matrix that stores behavioral rewards in belief states;
- L : State estimator of $D2$;
- 1: Naive Bayes classifier $T \leftarrow D2$;
- 2: False positive rate FP_{D1} , False negative rate FN_{D1} , positive rate TP_{D1} and negative rate $NP_{D1} \leftarrow D$;
- 3: Calculating L by Equation (10);
- 4: $\mathbf{x}_k = \mathbf{x}_{k-1} L$;
- 5: Positive rate (also risk assessment) TP_{D2} : calculated by $\hat{\mathbf{x}}_k$ and $T, FP_{D2} = FN_{D2} = 0$;
- 6: Calculating reward matrix R of the system by Equation (13);
- 7: get Pb by Equation (14) and Equation (15);
- 8: $Rb: Pb \times R$;
- 9: return Pb, Rb, L

Algorithm 1 shows the simplification process of the game model based on POMDP. In line 1, the polynomial fitting method is used to fit the data set $D2$ to a function representing the relationship between the data component $D2$ and the attack risk, which is used to evaluate the system state based on an unbiased estimator. In line 3, the set of Equation (10) is integrated into an unbiased estimator L of a physical information system, and the physical estimator of the next state is obtained by iterative operations. Using the Bayesian probability formula, the iterative formula in line 4, and the relationship function obtained in line 1, Algorithm 1 calculates and normalizes the FP , FN , TP , and NP of the two kinds of data sources in line 2 and line 5, respectively. In lines 6-8, Algorithm 1 switches the variables dependent on different matrices in the model from the system state s to the belief state b . This kind of matrix can transform the POMDP solution process into a relatively simple MDP solution.

Algorithm 1 completes the preliminary preparation of the decision model, which includes three steps: Determination of valuation indices, quantitative and normalized indices, and simplification of the solution model. After matrix transformation, the POMDP can be considered as a belief MDP. The iterative function of the belief MDP model to find the optimal solution is:

$$V_{k+1}(s) = \sum_{a \in A} \pi(a|s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_k(s') \right). \quad (17)$$

In the belief MDP model, the reward function $\rho(b, a, b')$ of the belief state representation is used to replace the parameter R_s^a in Equation (17), and $\rho(b, a, b') = \sum_{s, s' \in S} b(s)b'(s')r(s, a, s')$. In addition, $\sum_{a \in A} \pi(a|s)$ can be replaced by $\sum_{o \in \mathcal{O}} P(o|b, a)$ by increasing the belief state b and the observable value o . Then, by combining Equation (17) with the game model of this paper, the game solution formula can be obtained by using the min-max algorithm:

$$V_*(b, a) = \max_{a \in A_b} \left\{ \sum_{o \in \mathcal{O}} P(o|b, a) \left\{ \rho(b, a, b') + \gamma \left[\max_{a'_o \in A_{b'}} \sum_{o' \in \mathcal{O}} P(o'|b', a'_o) \left(\rho(b', a'_o, b'') + \gamma V(b'') \right) \right] \right\} \right\}. \quad (18)$$

The $A_{b'}$ in this equation means the optional action set of an attacker for belief state b' , and b' can be calculated by Equation (14). Since the parameters in Equation (13) take the expression of absolute value, Equation (18) calculates that the expected reward is positive for both sides of the game. And substitute min for max in the min-max algorithm.

4.2.2 | Iterative strategy

In this part, we introduce the Multiplicative Weights Update Method (MWUM)^{32,33} for large MDP's strategy iteration, this algorithm can speed up the convergence of algorithms, and this algorithm has the best performance in our simulation. The method uses the multiplicative update rule to iteratively change the weights to obtain a near-optimal strategy, which can be obtained by the Algorithm 2.

The algorithm has proved that the weight eventually converges to a point. Therefore, in this section, the algorithm is used to obtain the convergence value of the maximum likelihood strategy. Since Algorithm 2 takes the reward matrix as input, R_b of Algorithm 1 can be used to replace \mathcal{M} . After several rounds of iteration, the maximum value w_i^T in the weight vector w^T can be obtained and further used in the MPD iteration process.

In our model, we combined this algorithm to speed up the iteration of MDP decision process. Before the MDP iteration of a certain state, we will input the parameters into the MWUM for several iterations firstly, and output the final results after determining its stability, then use this value as the threshold. After each MDP iteration, we calculate the ratio of V given by formula $\frac{V_i}{\sum_{V_j \in V} V_j}$, and the iteration stops when the maximum ratio reaches this threshold and the output result or the result satisfies $|V_t(b) - V_{t-1}(b)| < \epsilon$ where the ϵ is a normal number.

Algorithm 2. Multiplicative weights update algorithm (MWUM)

Require:

- ϵ : An approximation parameter which satisfies $0 \leq \epsilon \leq 1/2$;
- T : Number of iterations;
- \mathcal{M} : Total reward matrix;

Ensure:

- M_s : except reward of state s ;
 - 1: initialize With each policy i , associate the weight $w_i^{(1)} := 1; \rho := 1$;
 - 2: FOR $t = 1; t \leq T; t++$
 - 3: FOR $i = 1; i \leq m; i++$
 - 4: $p_i^t := \frac{w_i^t}{\sum_{i=1}^m w_i^t}$;
 - 5: $w_i^{t+1} := w_i^t (1 - \epsilon)^{M(\pi_i, e_t) / \rho}$;
 - 6: end for
 - 7: choose the maximum $M(\pi_i, e_t)$ which with probability proportional to its weight w_i^t , get the policy i and its event e ;
 - and if $\rho \leq M_i$, let $\rho := M_i$;
 - 8: end for
 - 9: return maximum of w^T
-

4.2.3 | Parallel arithmetic

In practice, we decompose the solution process of Equation (18) into two parts according to the decision-making order, and we solve the MDP value function of the attacker's and defender's belief states respectively. Then, we form the income matrix with the V value of each state after solving, and use the min-max algorithm to solve the matrix.

Algorithm 3. Parallel POMDP-based game decision algorithm

Require:

- D : leaf node data set of IDS alarms;
- D_2 : set of current physical signals for some components;
- S : state transition matrix;
- Q : conditional observation probability matrix;
- A, B, C : the given matrices of components;

Ensure:

- max_i : the serial numbers in the action/policy set;
- 1: Initialize: R : an empty matrix which will be used to store the reward of both players;
- 2: $P_b, R_b, L \leftarrow$ Use the Algorithm 1 with input parameters;
- 3: $V_d, A_d \leftarrow$ Value function results, actions of set that obtained through solving of belief MDP and Algorithm 2;
- 4: $CORE_{tasks} \leftarrow B_d$, which is a collection of different belief states produced by A_d and current state b ;
- 5: $CORE_{tasks}$ **foreach**;
- 6: change the value of S, Q by A_{d_i} ;
- 7: modify the P_b and R_b by Algorithm 1;

- 8: $V_a, A_a \leftarrow$ Value function results, actions of set that obtained through solving of belief MDP;
- 9: $R(i, j) = \{V_{d_i}, V_{a_j}\}$, the j satisfies $V_{a_j} = \max V_a$;
- 10: end.foreach;
- 11: $\{max_i, max_j\} \leftarrow$ saddle point of the R ;
- 12: return max_i ;

In this process, our model calculates the actions and gains that an attacker will take in game theory on the basis of a belief state s' after the action a_j the defender has taken. The relationship between them is one-to-many, and the behavior between different belief states does not interfere with each other, so here we adopt a multi-processor parallel method, which can accelerate the computing process of the POMDP, to complete this subprocess.

The overall algorithm of the LIRS is shown in Algorithm 3 which is combined with Algorithms 2 and 1. On line 1-2, we evaluate the current state of the system statically and initialize the matrices needed for subsequent processes by Algorithm 1. On line 3, we get the vector V_d that represents the value of the possible actions and the vector A_d that represents the possible actions via the combination of the MDP and Algorithm 2. On line 4, we decompose the attacker's decision process into multiple subtasks based on the number of CPU processors, and we also establish multiple message queues here to handle cases where the number of subtasks is more than the number of processors. Because of the convergence of POMDP in finite horizon.³⁴ On line 6-10, we use the same process for the individual state of attacks just like line 2-4. We use the minimax algorithm to find the saddle point on line 11.

In summary, the overall framework of our model can be described in the following Figure 2:

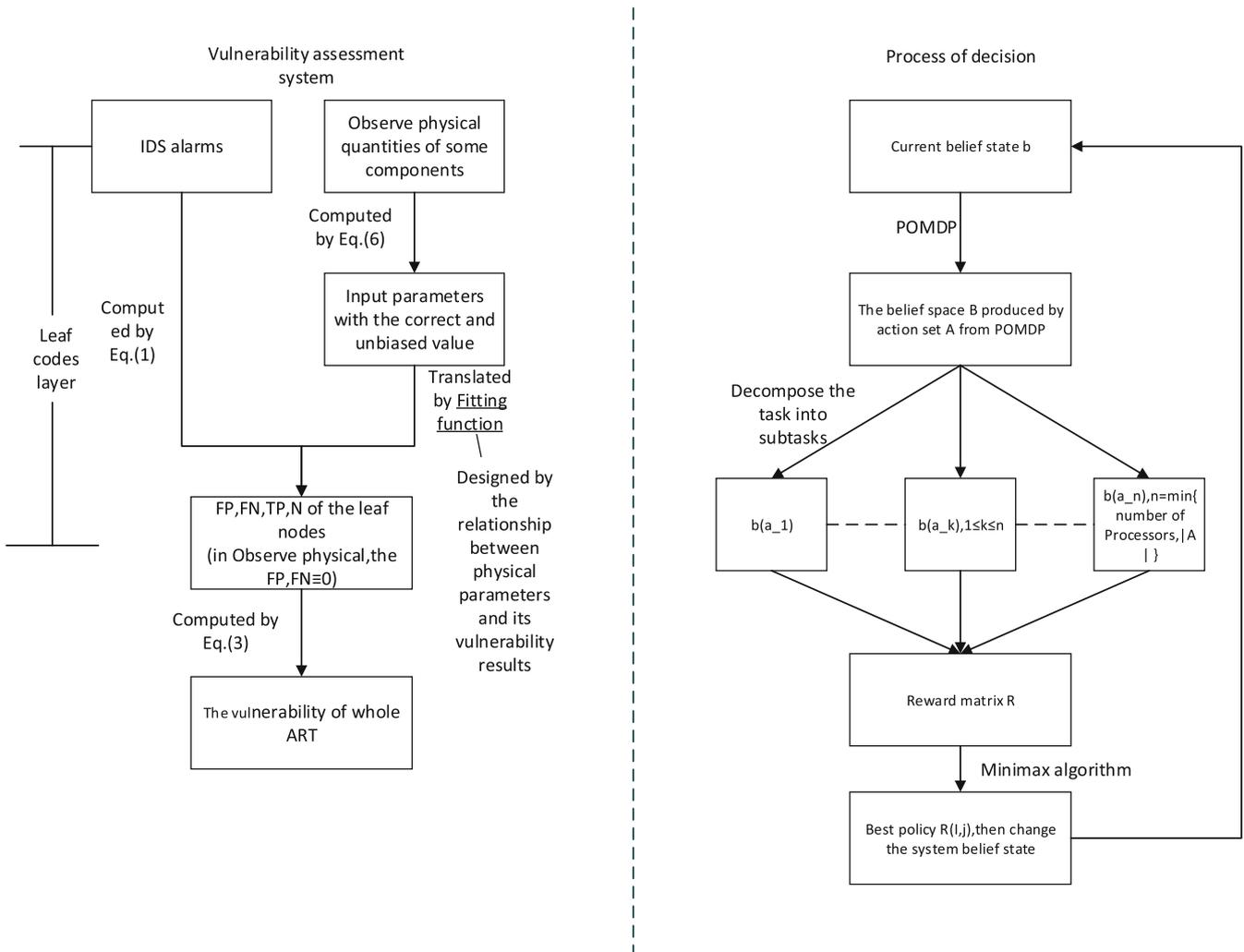


FIGURE 2 The process of analysis.

5 | EXPERIMENTS

In this section, we implement our game model on a simulated small hydropower control system and evaluate the experimental results on it and on other simulation experiments. Our simulation experiments are conducted on a Windows system with an Intel(R) Core(TM) i7-9750 CPU @2.60GHz with 16 GB of RAM.

5.1 | Experiment preparation

This system provides an overview of the SCADA of a small hydro power plant, and Figure 3 shows its structure. As shown in Figure 3, there are several hosts and PLCs, where PLCs represent a simulated pump control device consisting of two actuators(a motor and an intelligent pump)and a sensor. This LTI structure contains a three-component input vector and a four-component output vector.

We can see the structure of the entire experimental environment in Figures 3 and 4. On this basis, we abstract the functional nodes represented by each host and the simulation software executed on them as shown in Figure 5:

In Figure 5, we leave the structures of layers L2 and L3 corresponding to the upper two layers of Figure 4 partially omitted, since the leaf nodes corresponding to the result nodes of the two layers are both IDS alarms, similar to the PMU nodes of the L1 layer.

And some of the exposed vulnerabilities are shown in Table 3:

By exploiting these vulnerabilities, an attacker can invade the system using multiple nodes. Now we assume that the attack and defense only respond to the nodes that their strategy can reach, and then we can remove the leaf nodes that cannot be directly affected by both sides from experimental consideration.

In our experiment, we use the simulation software to simulate the industrial environment implemented by the independent host to simulate the linear time-invariant system. For example, in the software, a valve control system has three system state quantities, three control quantities and four observations. Its dynamic matrix satisfies:

$$A = \begin{bmatrix} 0.9856 & 0 & 0 \\ 0.1023 & 0.9867 & 0 \\ -0.0030 & 0.0120 & 1 \end{bmatrix}, \quad (19)$$

$$B = \begin{bmatrix} 0.5301 & 0.1035 & 0 \\ 0.0056 & 1.025 & 0 \\ 0.0071 & -0.3162 & 0.2517 \end{bmatrix}, \quad (20)$$

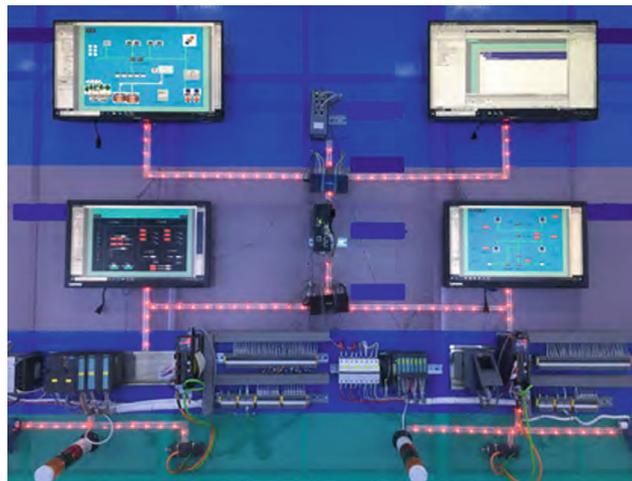


FIGURE 3 Photos of real environment.

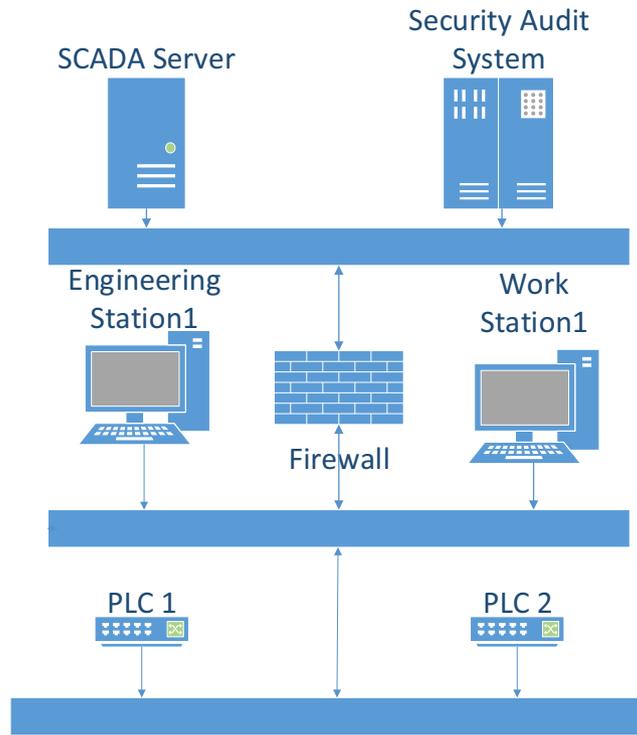


FIGURE 4 Network logic structure diagram.

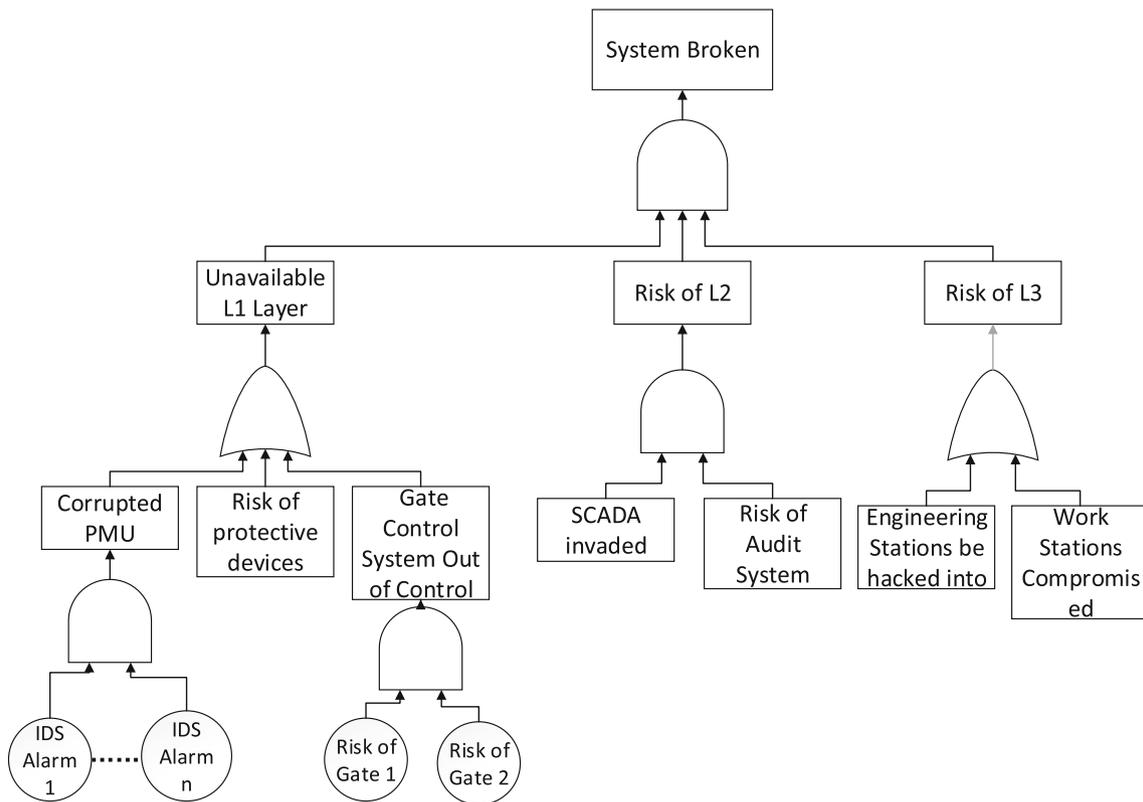
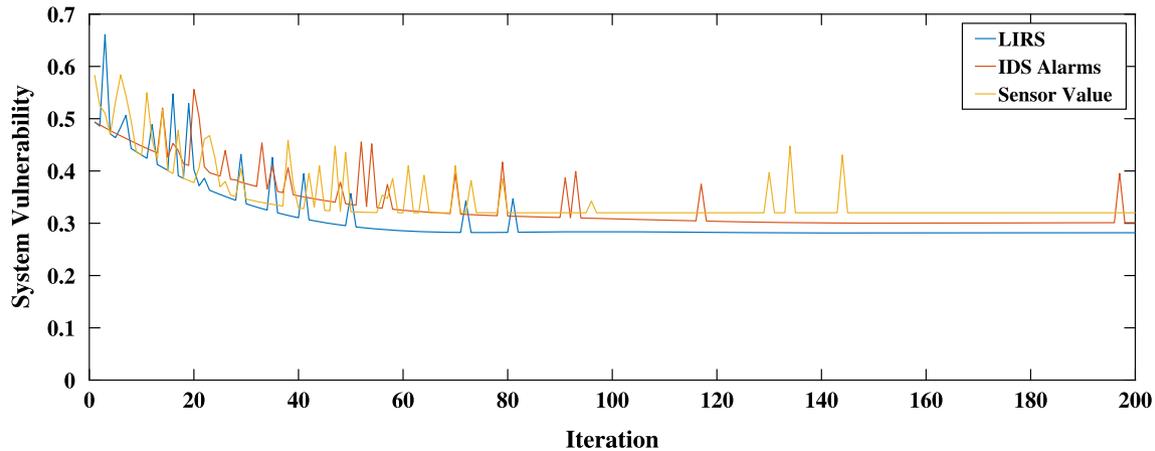


FIGURE 5 The logical tree of the simulated network.

TABLE 3 Part of vulnerabilities.

CVE	Target devices	Target version	CVSS score
CVE-2020-14580	Oracle Communications Session Border Controller	8.1.0–8.3.0	8.2 (CVSS 3.1based)
CVE-2017-3486	SQL*Plus component	11.2.0.4	7.2 (CVSS 3.0based)
CVE-2020-1860	NIP6800	V500R001 series	7.5 (CVSS 3.1based)
CVE-2019-5258	NIP6800	all versions	5.5 (CVSS 3.1based)
CVE-2020-16955 CVE-2020-16928	Microsoft Office Click-to-Run AppVLP	all versions	7.8 (CVSS 3.1based)
CVE-2013-3957	Web Navigator in Siemens SIMATIC WinCC	SIMATIC PCS7.8.0 SP1 and earlier	7.5 (CVSS 2.0based)

**FIGURE 6** The performance about evaluation system.

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.95 & 0.02 \\ 0.047 & 0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (21)$$

$$\mathbf{w}_k \sim \mathcal{N}(0, 0.05^2), \mathbf{v}_k \sim \mathcal{N}(0, 0.05^2). \quad (22)$$

5.2 | Results analysis

In this section, we will compare the experiments in terms of speed and vulnerability of the system after operation. And in experiments, we will compare our algorithm with the stochastic game model, POMDP and the cross-layer game model proposed by Huang et al.³⁵

5.2.1 | Performance of the evaluation system

In Section 4, we introduce our evaluation system to calculate the risk coefficient of leaf nodes in the expanded tree. Here we will show the improvement of our evaluation system by comparing its effects with the effects of the other two methods. According to the information in chapter 4, we set the number of IDS alarms and the number of sensor values in the LIRS to a random sequence at a ratio of approximately 5:1.

The results of three different risk evaluation systems are compared in this experiment. As shown in Figure 6, the abscissa axis is the number of iterations during the POMDP solution process of three evaluation systems, while the vertical axis represents the root node risk coefficient of the system implemented by the experimental environment. This coefficient is calculated by Equation (3).

Figure 6 shows that the evaluation system of our model achieves the best effect in the performance of the vulnerability of the experimental environment. Its performance is better than the IDS alarms set only by approximately 6.35% and is better than the sensor value set by approximately 11.93%. The number of iterations needed to reach a stable value is also the lowest.

TABLE 4 Speed of decisions.

model	CPU time of decisions (s)	Number of iteration rounds (times)
LIRS	3.1571	48.1785
Stochastic game	3.5177	60.2352
POMDP	3.6320	53.31
Cross-layer model	3.2547	50.3839

TABLE 5 Total time of response engines.

Model	Total time (s)
LIRS	18.7113
Stochastic game	14.180
POMDP (without parallel method)	25.8561
Cross-layer model	21.5715

5.2.2 | Speed performance

First, we test the response speed performance. In this experiment we test the response times of several algorithms in the same environment, which includes fifteen leaf nodes and fifteen non-leaf nodes. The results are shown in Tables 4 and 5 after taking the mean values of several experiments. Each experiment consists of 10 rounds of attack and defense decisions.

In Table 4, we can find that the average CPU time of each decision is 0.0655, 0.0584, 0.06782, and 0.06460 s, respectively. The Stochastic game model takes the shortest time among the three methods because the other two methods require several iterations, thus taking more time. And that the rate of the LIRS is about 4% higher than that of the POMDP decision and 1.37% higher than the cross-layer model by Huang. Furthermore, we find that in terms of the number of iterations, the LIRS shows 20.02%, 9.63%, and 4.38% fewer iterations than the other three methods, respectively. So in terms of total time, LIRS is the best by a narrow margin.

We can find that the stochastic game takes the least time out of three methods because the other two approaches need to update the belief state b correlation matrix after each attack and defense, and this process takes a long time to read and write the memory. However, the LIRS is 23.63% faster than the POMDP and 13.26% faster than the cross-layer model because of faster reward matrix computation speed.

5.2.3 | Functional representation

In this section, we focus on the comparison of the system through the process and take the vulnerability of the system as the evaluation indicator, which can be calculated by Equations (3)–(5). This experiment was conducted in the simulated environment of the small hydro power plant described above.

Because of the difference in the income function, measurement of the evaluation indicators and calculation methods between different algorithms, the rate of return is not suitable as the evaluation index of the results. At this point, we can choose some experimental environment related parameters, such as the node risk coefficient of the system, the MTTR, the TTSD and others, as income indexes instead. These parameters can effectively reflect the current state change of the system after the defense takes measures. Here, we take the root node risk coefficient as the evaluation coefficient of this experiment according to Section 4.1. The results are shown below:

In Figure 7, we can see that the experiment performed a total of 500 iterations. In Figure 7, the rate of convergence of the two methods is not very different. When $\epsilon=10^{-6}$, the LIRS converges at 76 rounds, the Stochastic game model converges at 103 rounds, and the Cross-Layer model by Huang converges at approximately 175 rounds. With a gap of 3.92% between the two approximations, the two curves eventually stabilize at 0.2818, 0.3010 and 0.2914. We also find that the proportion of V in the maximum income strategy of the LIRS is 99.15% and the ratio of the Stochastic game model and the cross-layer model are 98.25% and 97.38%, respectively. Therefore, in the back section of the curve, the LIRS produces less policy noise than the latter, and because of the algorithm, the noise produced by the LIRS is also lower than that of the latter. These data show that our model is better than the Stochastic game model.

Furthermore, we perform several experiments on the number of nodes and the number of iteration rounds needed for system stability, and the results are shown in Figure 8.

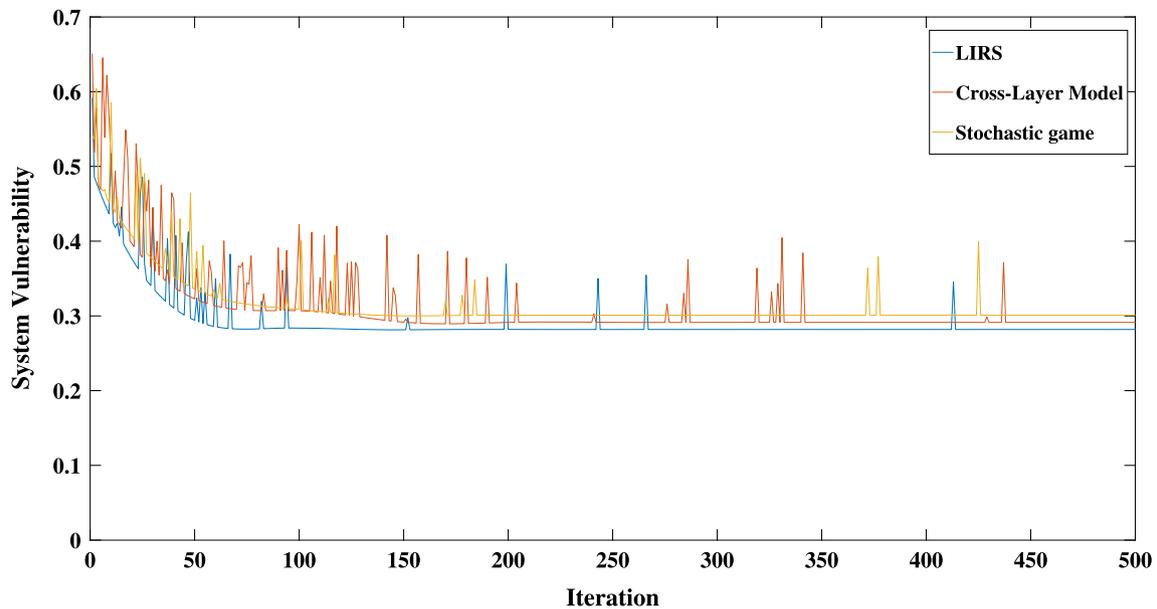


FIGURE 7 Vulnerability after defensive measures.

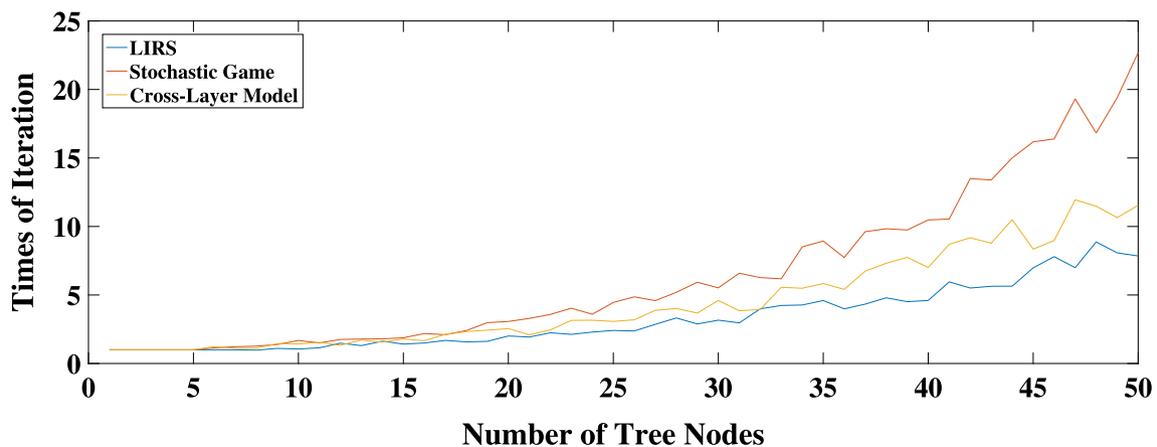


FIGURE 8 The curve of times of iteration.

In Figure 8, we can find that the relationship between the number of iteration rounds and the number of leaf nodes is polynomial in the LIRS, and the value of the points on our model's curve is lower than the corresponding values of the Stochastic game and the cross-layer model.

6 | CONCLUSION

In this paper, we developed a game theoretic model for intruder response in SCADA. First, we established an expanded attack tree model, and then we quantified the IDS alarm set and quantities into the probability of being attacked to form an evaluation system. Then, we accelerated the POMDP process by (1) using the MWUM algorithm to reduce the number of iteration rounds; (2) decomposing the solution process on the back of the game according to the number of states and using the parallel method to speed up this process. In the simulation, compared with the other two approaches, the LIRS achieves the best results in a short time.

In future work, we plan to extend the evaluation indicators to other physical quantities and test other ways to reduce the state space of our expanded attack tree. We also intend to integrate time data in the industrial control process into our model evaluation system.^{36,37}

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (Grant No. 2020YFB2104000), the NSFC (Grant No. 61802032) and the Key Area Research Program of Hunan (2019GK2091), the Program of National Natural Science Foundation of China (No. 62002114), the Program of Natural Science Foundation of Hunan (No. 2021JJ40108), the Program of Natural Science Foundation of Changsha (No. kq2202318), the Program of Natural Science Foundation of Hunan (General Program No. 2023JJ30102).

DATA AVAILABILITY STATEMENT

Research data are not shared.

ORCID

Siyang Yu  <https://orcid.org/0000-0003-4510-0729>

Fan Wu  <https://orcid.org/0000-0001-9392-2597>

REFERENCES

- Zolanvari M, Teixeira MA, Gupta L, Khan KM, Jain R. Machine learning-based network vulnerability analysis of industrial internet of things. *IEEE Internet Things J*. 2019;6(4):6822-6834.
- Rathore MM, Ahmad A, Paul A. Real time intrusion detection system for ultra-high-speed big data environments. *J Supercomput*. 2016;72:3489-3510.
- Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun Surv Tutor*. 2016;18(2):1153-1176.
- Paul A, Jeyaraj R. Internet of things: a primer. *Human Behavior Emerg Technol*. 2019;1(1):37-47.
- Khan K, Mehmood A, Khan S, Khan MA, Iqbal Z, Mashwani WK. A survey on intrusion detection and prevention in wireless ad-hoc networks. *J Syst Archit*. 2020;105:101701.
- Hu Y, Yang A, Li H, Sun Y, Sun L. A survey of intrusion detection on industrial control systems. *Int J Distribut Sensor Networks*. 2018;14(8):1550147718794615.
- Hasan S, Dubey A, Karsai G, Koutsoukos X. A game-theoretic approach for power systems defense against dynamic cyber-attacks. *Int J Electr Power Energy Syst*. 2020;115:105432.
- Chevalier R. Detecting and surviving intrusions: exploring new host-based intrusion detection. *Recovery Response Approaches*. 2019;2019CSUP0003.
- Liu S, Paul A, Zhang G, Jeon G. A game theory-based block image compression method in encryption domain. *J Supercomput*. 2015;71:3353-3372.
- Gyamfi E, Jurcut A. Intrusion detection in internet of things systems: a review on design approaches leveraging multi-access edge computing, machine learning, and datasets. *Sensors*. 2022;22(10):3744.
- Giraldo J, Sarkar E, Cardenas AA, Maniatakos M, Kantarcioglu M. Security and privacy in cyber-physical systems: a survey of surveys. *IEEE Design Test*. 2017;34(4):7-17.
- Inayat Z, Gani A, Anuar NB, Khan MK, Anwar S. Intrusion response systems: foundations, design, and challenges. *J Netw Comput Appl*. 2016;62:53-74.
- Zonouz SA, Khurana H, Sanders WH, Yardley TM. RRE: a game-theoretic intrusion response and recovery engine. *IEEE Trans Parallel Distribut Syst*. 2014;25(2):395-406.
- Miehling E, Rasouli M, Teneketzis D. Optimal Defense policies for partially observable spreading processes on Bayesian attack graphs. *Associat Comput Mach*. 2015;67-76.
- Shameli-Sendi A, Ezzati-Jivan N, Jabbarifar M, Dagenais M. Intrusion response systems: survey and taxonomy. *Int J Comput Sci Network Security*. 2012;12(1):1-14.
- Li X, Zhou C, Tian Y, Qin Y. A dynamic decision-making approach for intrusion response in industrial control systems. *IEEE Trans Indust Informat*. 2019;15(5):2544-2554.
- Ullah S, Shelly S, Hassanzadeh A, Nayak A, Hasan K. On the effectiveness of intrusion response systems against persistent threats. *2020 International Conference on Computing, Networking and Communications (ICNC)*. 2020:415-421.
- Kiernert C, Ismail Z, Debar H, Leneutre J. A survey on game-theoretic approaches for intrusion detection and response optimization. *ACM Comput Surv*. 2018;51(5):1-31.
- Gill KS, Saxena S, Sharma A. GTM-CSec: game theoretic model for cloud security based on IDS and honeypot. *Comput Secur*. 2020;92:101732.
- Liu G, Xiao Z, Tan G, Li K, Chronopoulos AT. Game theory-based optimization of distributed idle computing resources in cloud environments. *Theor Comput Sci*. 2020;806:468-488.
- Xingshuo A, Fuhong L, Shenggang X, Li M, Chao G. A novel differential game model-based intrusion response strategy in fog computing. *Secur Commun Networks*. 2018;2018:1-9.
- Orojloo H, Azgomi MA. Evaluating the complexity and impacts of attacks on cyber-physical systems. *CSI Symposium on Real-Time and Embedded Systems and Technologies (RTEST)*. Vol 2015. IEEE; 2015:1-8.
- Orojloo H, Abdollahi AM. Predicting the behavior of attackers and the consequences of attacks against cyber-physical systems. *Secur Commun Networks*. 2016;9(18):6111-6136.
- Wang C, Lei S, Ju P, Chen C, Peng C, Hou Y. MDP-based distribution network reconfiguration with renewable distributed generation: approximate dynamic programming approach. *IEEE Trans Smart Grid*. 2020;11(4):3620-3631.
- Zhao X, Bhuiyan MZA, Qi L, Nie H, Tang W, Dou W. TrCMP: a dependable app usage inference design for user behavior analysis through cyber-physical parameters. *J Syst Archit*. 2020;102:101665.
- Basile G, Marro G. On the observability of linear, time-invariant systems with unknown inputs. *J Optim Theory Appl*. 1969;3(6):410-415.
- Ali S, Balushi TA, Nadir Z, Hussain OK. ICS/SCADA system security for CPS. *Cyber Secur Cyber Phys Syst*. 2018;89-113.
- Antsaklis PJ, Koutsoukos XD. Hybrid systems: Review and recent progress[J]. *Software-Enabled Control: Information Technology for Dynamical Systems*. 2003:273-298.

29. Kitanidis PK. Unbiased minimum-variance linear state estimation. *Automatica*. 1987;23(6):775-778.
30. Gillijns S, Moor BD. Unbiased minimum-variance input and state estimation for linear discrete-time systems. *Automatica*. 2007;43(5):934-937.
31. Braziunas D, Boutilier C. Stochastic local search for POMDP controllers. *Proceedings of the 19th National Conference on Artificial Intelligence*. 2004:690-s696.
32. Lassaigne R, Peyronnet S. Approximate planning and verification for large Markov decision processes. *Int J Softw Tools Technol Transfer*. 2015;17(4):457-467.
33. Arora S, Hazan E, Kale S. The multiplicative weights update method: a meta algorithm and applications. *Theory Comput*. 2012;8(1):121-164.
34. Sondik SEJ. The optimal control of partially observable Markov processes over a finite horizon. *Oper Res*. 1973;21(5):1071-1088.
35. Huang K, Zhou C, Qin Y, Tu W. A game-theoretic approach to cross-layer security decision-making in industrial cyber-physical systems. *IEEE Trans Ind Electron*. 2020;67(3):2371-2379.
36. Chen J, Li K, Rong H, Bilal K, Li K, Yu PS. A periodicity-based parallel time series prediction algorithm in cloud computing environments. *Inf Sci*. 2019;496:506-537.
37. Goldschmidt T, Hauck-Stattelmann S, Malakuti S, Grüner S. Container-based architecture for flexible industrial control applications. *J Syst Archit*. 2018;84:28-36.

How to cite this article: Yu S, Wu F, Chen B, Cao R, Yang Z, Li K. A parallel game model-based intrusion response system for cross-layer security in industrial internet of things. *Concurrency Computat Pract Exper*. 2023;35(28):e7826. doi: 10.1002/cpe.7826