CIT. Journal of Computing and Information Technology, Vol. 34, No. 1, March 2026, 45–63
doi: 10.20532/cit.2026.1006258

A SARSA-Driven Cyber-Physical Risk Modeling Framework for Cloud-Based CBTC Systems

Lin Deng, Zhao Ping and Hui Xiong

Sichuan Vocational and Technical College, Suining City, Sichuan Province, China

After the communication-based train control system is migrated to a cloud computing environment, it will face new and complex security risks caused by the deep coupling characteristics of cyber-physics. To this end, based on complex network, fault tree analysis, and attack graph theory, the research constructs a cyber-physical coupling risk model that can quantify the importance of nodes and the probability of multi-step attacks. Subsequently, the "state-action-reward-state-action" (SARSA) algorithm is innovatively introduced to automatically search for the optimal attack path that can cause the greatest cumulative risk by simulating the attacker's decision-making process in the model environment. The results revealed that the node degree of the physical master node was as high as 36.14, and it was accurately identified as the most critical node of the system. Meanwhile, the cumulative risk value of the optimal attack path discovered by the SARSA algorithm was 85.4, which was higher than that of the comparison algorithm. In the application verification, after deploying security measures, the system risk value assessed by this method was significantly reduced from 85.4 to 31.2, a decrease of 63.5%. It shows that the proposed cyber-physical coupling risk modeling method can effectively identify the key risk nodes of the system, and the SARSA algorithm can effectively solve the optimal attack path optimization problem. The significance of the research is to provide a quantifiable dynamic risk assessment framework and a lightweight solution for security defense in an edge computing environment.

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

Keywords: CBTC, SARSA, Cloud computing, Cyber-physical coupling, Risk modeling

1. Introduction

With the development of urban rail transit towards networking and intelligence, communication-based train control systems (CBTCs) have evolved into typical large-scale cloud-based cyber-physical systems (CPSs) [1−3]. Although these systems achieve efficient train scheduling through wireless communication, their open network architecture also introduces complex network physical security threats. Traditional security analysis methods are difficult to cope with this highly dynamic and strongly coupled industrial control environment, posing serious challenges to existing security modeling and defense mechanisms [4−5]. In this context, an important challenge is how to use computational intelligence to accurately model and quantify network physical security risks and adaptively optimize them in the complex topology of cloud-based CPS. Existing research has difficulty considering the causal logic of attack paths and the real-time nature of defense decisions simultaneously [6−7]. Therefore, it is necessary to construct a hybrid computing framework that can integrate structured prior knowledge with data-driven learning.

"State-action-reward-state-action" (SARSA), as a classic and efficient same-strategy reinforcement learning (RL) algorithm, has been widely used to solve various complex sequence decision-making and optimization problems. Liu *et al.* proposed an intelligent scheduling method based on the improved SARSA frame-

work to solve the problem of a lack of efficient automation solutions for main production scheduling. The results showed that the convergence and scheduling efficiency of this method were significantly better than traditional methods and other RL algorithms [8]. To achieve an accurate prediction of battery power of rail transit workshop robots, Peng *et al.* proposed a hybrid prediction model based on the SARS algorithm. The results revealed that the hybrid model based on SARS had better accuracy, robustness, and adaptability on real data sets [9].

In summary, the SARSA algorithm has shown great potential in multiple fields such as production scheduling, equipment prediction, and autonomous driving. The research mainly proposes a modeling and evaluation method for information physical coupling security risks in CBTC systems. This method establishes an information physics model to comprehensively characterize the assets, functional dependencies, and attack vectors of cloud-based CBTC systems. Then, the SARSA algorithm is introduced to automatically search for the optimal attack path by simulating intelligent attacks. The cumulative risk of this path is used as the quantitative evaluation result of the overall system risk. This allows for the accurate identification of the most dangerous attack sequence faced by the system.

2. Contributions and Novelty

The research's innovation and contribution lie in constructing a weighted, complex network model that abstracts the physical and network topologies of CBTC. This solves the problem of mapping interactions between heterogeneous nodes. Meanwhile, a dynamic risk assessment algorithm combining fault tree analysis (FTA) and attack graph is proposed. This method uses FTA to identify the logical root cause of physical faults and determine the lateral movement path at the network level using attack graphs. This method achieves joint quantification of cross-domain risks. In addition, for dynamic attack scenarios, the study modeled defense strategy generation as a Markov decision process and designed an improved SARSA RL algorithm. This algorithm can automatically identify the optimal defense action sequence through online learning in uncertain network environments. This method is superior to traditional static defense strategies.

3. Research Methodology

In a cloud computing environment, in order to implement CBTC system security risk assessment, this research constructs a coupled risk model that can accurately describe the cyber-physical characteristics of the system. Based on this model, the advanced intelligent algorithm SARSA algorithm is used to conduct risk assessment on potential optimal attack paths.

3.1. Modeling for Cyber-physical Coupling Security Risks in CBTC Systems Based on Complex Networks and Attack Graphs

The overall architecture of cloud-assisted CBTC network physical security analysis designed for research is shown in Figure 1. The framework uses a hierarchical, modular design to address security risk quantification and path optimization issues in heterogeneous CBTC environments. The first layer is the complex network abstraction layer, which abstracts the physical entities and network entities of the CBTC system into heterogeneous complex network models. The second layer is the physical coupling network layer, which connects physical and network security. The node degree index of fusion control weights is calculated by introducing FTA, quantifying the logical roots of physical side accidents, and projecting physical risks to network nodes through mapping functions. The third layer is the attack graph generation layer. It combines lateral movement rules to dynamically generate state attack graphs. This process involves discretizing continuous attack processes into the state space of Markov decision processes. The fourth layer is the RL optimization layer, which deploys an improved SARSA algorithm.

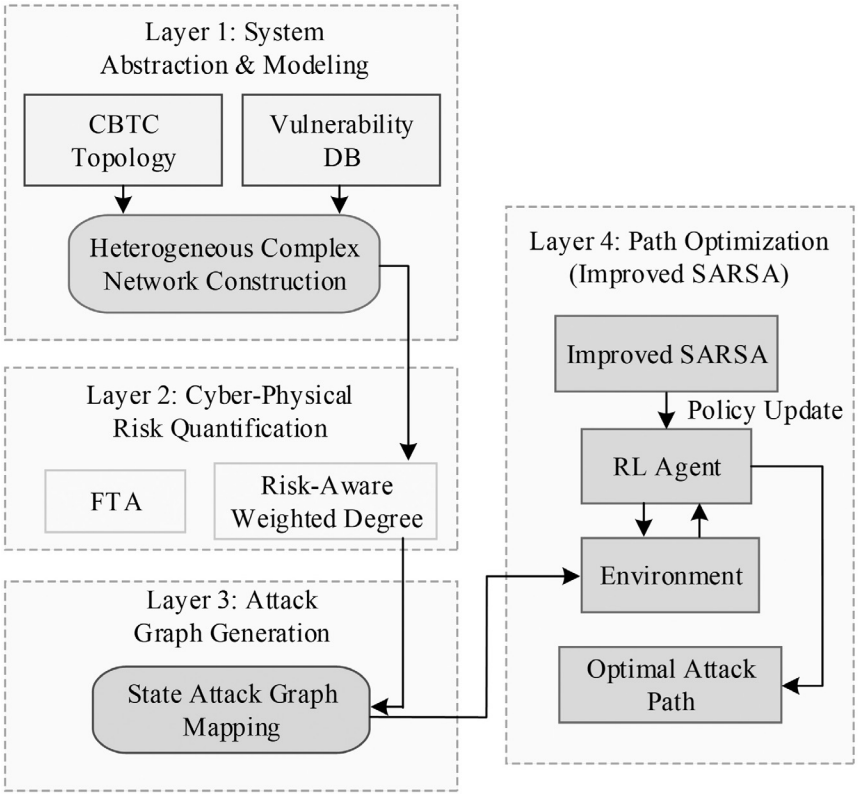


Figure 1. Overall architecture of cloud assisted CBTC network physical security analysis.

To accurately depict the complex characteristics of the cloud computing-oriented CBTC system in the structural, functional and security dimensions, the study first introduces complex network theory and abstracts the system into a dynamic network model containing multiple types of nodes and complex connection relationships. On this basis, FTA is used to analyze key business logic and quantify the influence of each network node in train operation control. Finally, combined with attack graph theory, a complex network risk model that can describe persistent network attack scenarios is constructed, as shown in Figure 2.

After the traditional CBTC system architecture is migrated to the cloud computing platform, its asset composition and interactive relationships have undergone fundamental changes. The system is no longer just a collection of signaling devices, but has evolved into a hybrid containing physical infrastructure, virtualized resources, and external on-board devices [10–12]. In Figure 2, a three-layer heterogeneous complex network model is constructed, denoted as $G_{CPS} = (N, E, \Phi)$. Among them, N represents the set of nodes. In this structure, P represents a set of physical nodes. V represents a set of cloud service virtual nodes, which are virtual machines running CBTC core business. Each computing node contains 4 subsystem virtual machines. O represents the vehicle mounted controller device node outside the cloud platform. It receives information domain instructions and executes physical domain control. E represents the set of edges, and Φ is the coupling function. If node 1 is attacked, the probability of damage to node 2 is determined by $\Phi(1, 2)$. In CBTC systems, the core element ensuring safe and efficient train operation is the computation and transmission of the movement authority (MA) [13–15]. Therefore, the study adopts the MA transmission path as a key metric for evaluating network functionality. The schematic diagram of the MA transmission path is shown in Figure 3.

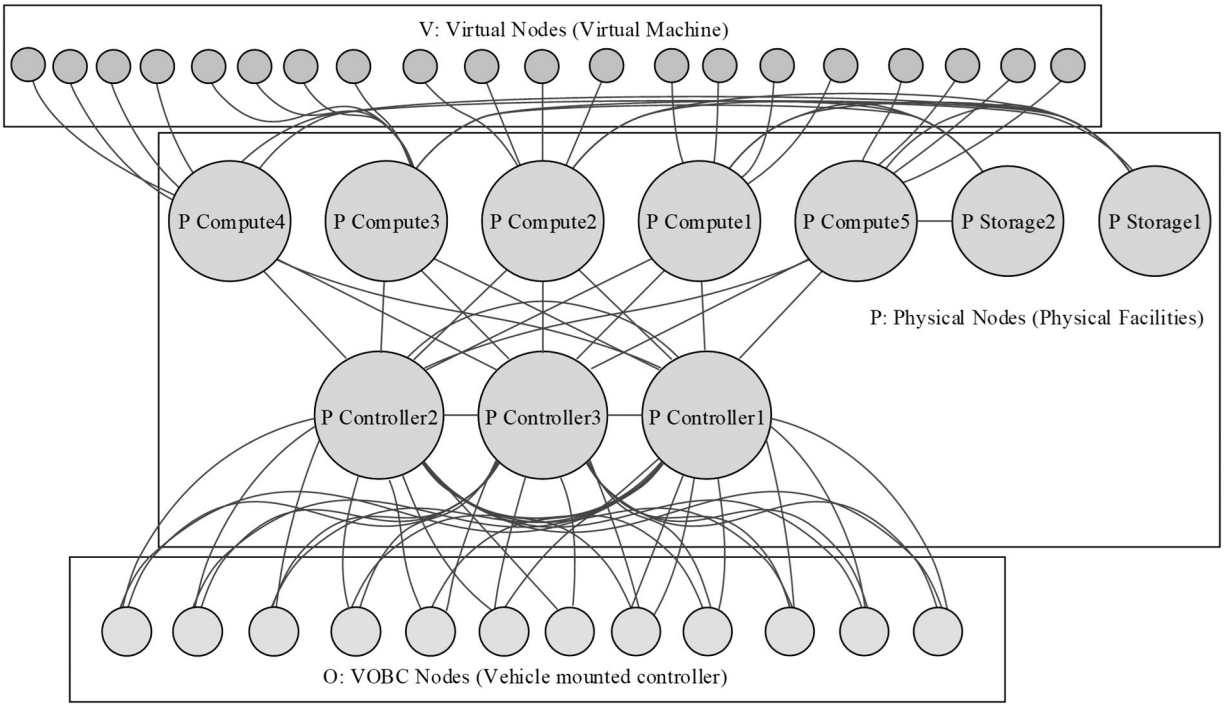


Figure 2. Complex network risk model.

Figure 3 depicts a set of nodes and edges forming an MA transmission path that originates from a virtual node, traverses physical nodes and network devices, and ultimately reaches the vehicle-mounted controller device node. In complex networks, node importance is typically measured by degree [16–18]. However, simple degree fails to reflect a node's functional significance in critical operations such as MA transmission. To address this, this study proposes a node degree calculation method that integrates control weights. The reason for using this indicator instead of directly using the standard network centrality indicator in the study is that the CBTC system has asymmetry. Standard graph theory indicators only focus on network topology connectivity, ignoring the physical control logic and consequences of accidents behind nodes. Among them, betweenness centrality measures the ability of nodes to act as information bridges. Many critical security execution devices are at the end of the network topology, with metric values close to 0. Once these nodes are compromised, it will directly lead to train derailment or collision. The node degree that integrates control weights can accurately identify key nodes at the topological edge and secure the core by introducing a physical importance weight, which is calculated by FTA. Meanwhile, proximity centrality indicators cannot distinguish the physical value of target nodes. In addition, in industrial control systems, a common sensor triggers the emergency stop action of the core controller directly through hard wired logic. The centrality index of feature vectors is difficult to capture this cross domain causal relationship. The degree of node is calculated as shown in Equation (1).

$$\delta_i = \alpha \times C_i + \beta \times \omega_i \tag{1}$$

In Equation (1), C_i represents the number of connecting edges for a node. ω_i denotes the control weight of node i. α and β are weight coefficients. For the calculation of ω_i, the study introduces the FTA method. For each MA transmission path, a fault tree is constructed that cannot be connected. The top event of this fault tree is termed MA transmission path interruption, while the bottom events represent failures of physical or virtual nodes within the network. The redundant architecture, cluster fault-tolerance mechanisms, and dependencies between virtual nodes and physical nodes are described using logic gates [19–20]. After completing system modeling and quantifying node impor-

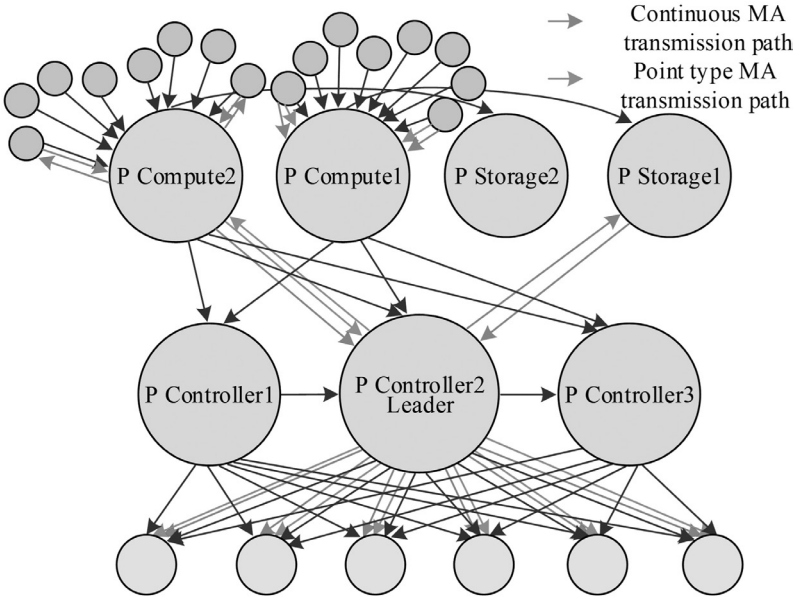


Figure 3. Schematic diagram of the MA transmission path.

tance, the attacker's perspective must be introduced to assess the system's security vulnerabilities. Attack graphs serve as an effective tool for describing such multi-step attack processes. An example is shown in Figure 4.

In Figure 4, gray circles represent different vulnerabilities, and directed edges represent attack associations between vulnerabilities. There are three attack paths that can reach the attack target, namely (attack starting point 1→vulnerability 1→vulnerability 2→target), (attack starting point 1→vulnerability 1→vulnerability 3→target), and (attack starting point 2→vulnerability 2→target). The study constructs an attack graph scenario based on complex networks, formally represented as a triplet $A = (I, L, T)$. Among these, I denotes the set of nodes within the complex network, L describes the attack association relationships between nodes, and T represents the set of exploit templates [21–22]. The study employs the common vulnerability scoring system (CVSS) to quantify the probability of exploitation for individual vulnerabilities. The probability $P(\varepsilon)$ of exploiting a vulnerability ε is determined by its exploitability metric, as shown in Equation (2).

$$P(\varepsilon) = M \times AV \times AC \times PR \times UI \qquad (2)$$

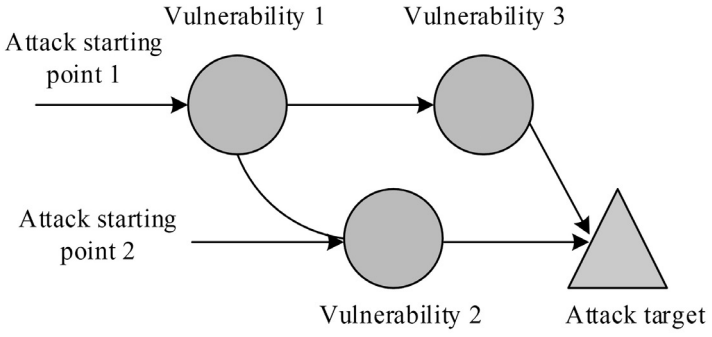


Figure 4. Example of an attack diagram.

In Equation (2), AV, AC, PR, and UI are metrics defined by CVSS, representing attack vector, attack complexity, privilege requirement, and user interaction, respectively. M is a constant. The success probability $P(A_u)$ of a unit attack A_u on a target node is defined as the maximum probability of exploitation among all vulnerabilities exploitable on that node by an attacker who satisfies the prerequisites. The calculation of $P(A_u)$ is shown in Equation (3).

$$P(A_u) = max\{P(\varepsilon 1), P(\varepsilon 2), ..., P(\varepsilon m)\} \quad (3)$$

Combining the above processes, the study constructs a cyber-physical coupling risk model. This model mainly uses complex networks to analyze the functions of the CBTC system in a cloud computing environment and also uses fault trees to quantify the importance of each node. In addition, the study uses attack graphs and CVSS to evaluate the risks and occurrence probabilities of the system.

3.2. Optimal Attack Path Risk Assessment Based on the SARSA Algorithm

After risk modeling is completed, the next step is to address the issue of quantifying the impact of an attack and finding the greatest risks to the system. To quantify the impact of attacks on the system, the research is conducted from the information domain and physical domain. Among them, the impact of the information domain is seen as a decrease in network performance. The impact in the physical domain represents the degradation of train operating performance. The attack impact transmission model diagram is shown in Figure 5.

In Figure 5, the comprehensive impact $\mathcal{J}$ of a single unit attack is defined as the fusion of information domain impact and physical domain impact, as shown in Equation (4).

$$\mathcal{J} = \Delta\eta(\mathcal{N}) \times (\Pi - \Pi') \quad (4)$$

In Equation (4), $(\Pi - \Pi')$ represents the degradation in control performance, while $\Delta\eta(\mathcal{N})$ denotes the reduction in connectivity. The network of CBTC systems for cloud computing is vast and complex, featuring an enormous number of attack paths. Traditional exhaustive search methods struggle to identify the optimal attack path with the highest risk [23]. To address this, the study introduces RL to simulate the behavior of an attacker aiming to maximize disruption. This problem is formulated as an RL model, as illustrated in Figure 6.

Figure 6 illustrates the components of this attack model: agents, environment, state, actions, and rewards. The agent represents the attacker. The environment denotes the complex network attack graph scenario of the cloud-based CBTC system. The state comprises the set of currently compromised nodes and the network status. An action refers to initiating a unit attack on an accessible node that satisfies exploitation conditions, starting from the currently controlled node. The reward indicates the risk value obtained after executing a unit attack action. Con-

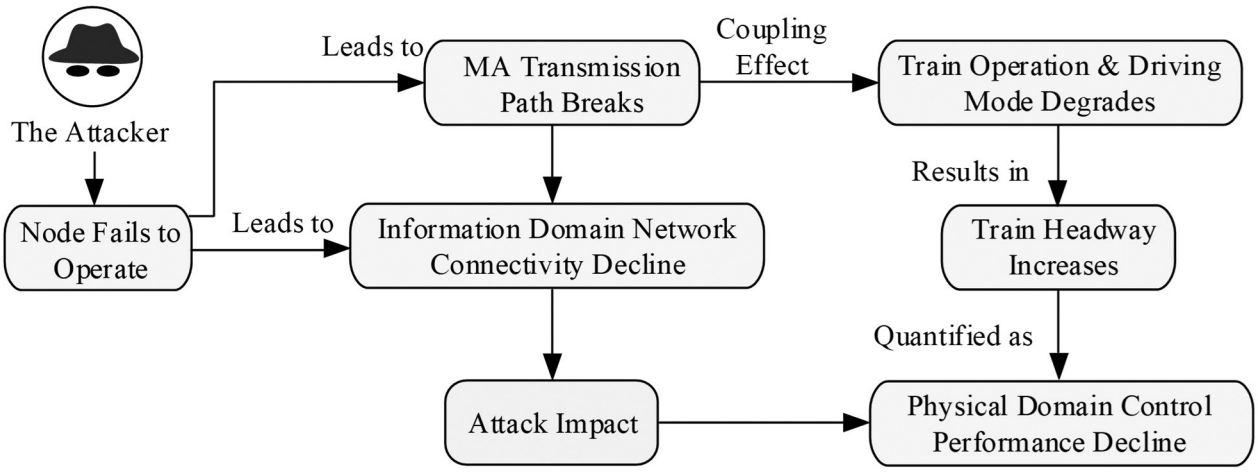


Figure 5. Attack impact transmission model.

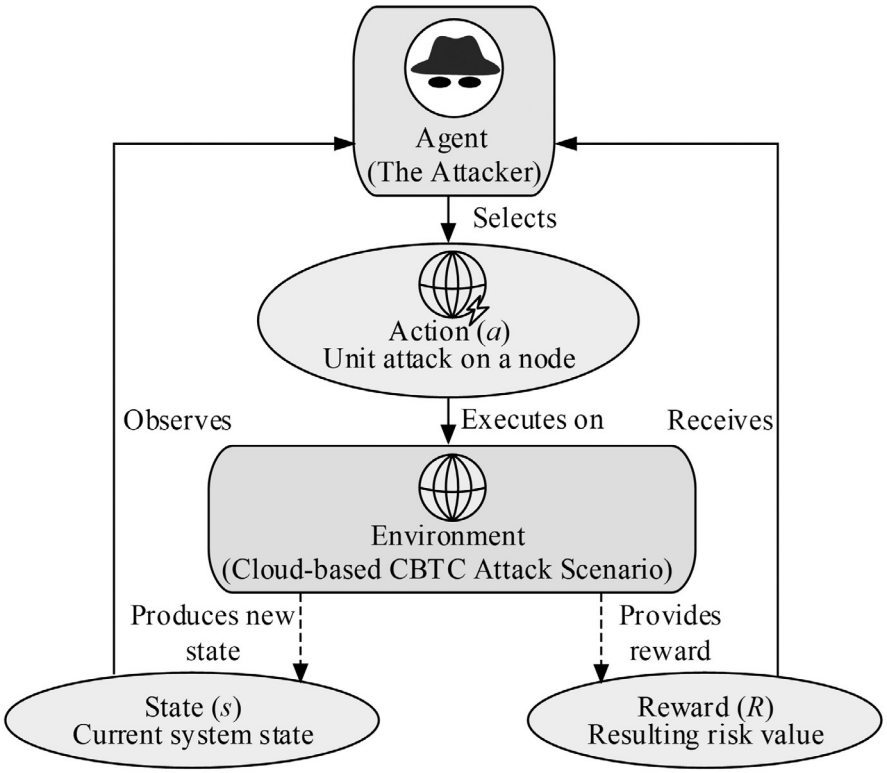


Figure 6. RL based attack model.

sidering that attack costs increase with path depth, the reward R_t for step t is defined as shown in Equation (5).

$$R_t = \begin{cases} P(A_{u1}) \times \mathcal{J}_1, & t = 1 \\ P(A_{ut}) \times \mathcal{J}_t \times \prod_{j=1}^{t-1} P(A_{uj}), & t > 1 \end{cases} \tag{5}$$

In Equation (5), $P(A_{u1})$ denotes the probability of executing the t-th attack action starting from the origin, with $\mathcal{J}_1$ representing the corresponding impact. The risk of subsequent attacks diminishes as the success probabilities of prior attacks are multiplied cumulatively. The agent's objective is to maximize the cumulative reward of the attack path, which is also expressed as maximizing the cumulative risk value R_{sys}, as shown in Equation (6).

$$R_{sys} = \max_{\pi} E\left[\sum_{t=1}^{T} R_t\right] \tag{6}$$

Although deep RL performs well in processing high-dimensional visual input tasks, in CBTC security defense decision-making scenarios, research mainly chooses SARSA to solve the optimal attack path model. The reason is that the deep Q-network (DQN) is a heterogeneous strategy algorithm which assumes the agent will always perform the optimal action. However, this is often too aggressive when exploring the attack path. On the contrary, SARSA is a same strategy algorithm that considers the actions performed by the agent when updating the Q-value. Second, the attack graph model essentially constitutes a finite discrete state space. Compared to the black box characteristics of neural networks, SARSA based on table method has a theoretically determined convergence proof in finite Markov decision processes. In addition, the computing resources of CBTC's edge devices are limited. SARSA's extremely low time complexity and space requirements enable direct deployment on embedded devices, such as regional controllers, for real-time inference without relying on high-computing-power GPU servers. The pseudocode of SARSA is shown in Table 1.

The flowchart of the optimal attack path search based on SARSA is shown in Figure 7.

Table 1. Pseudo code of SARSA.

Optimal Attack Path Search Algorithm Based on Improved SARSA
Input: Attack graph state space S, Action space A, Initial state s0, Learning rate α, Discount factor γ, Exploration rate ϵ, Maximum episodes Max_Episodes.
Output: Optimal Attack Path Pathopt.
1: Initialize Q(s, a) arbitrarily (*e.g.*, to 0) for all s ∈ S, a ∈ A
2: Initialize parameters α, γ, ϵ
3: For each episode do:
4: Initialize agent state st=s0
5: Step 1: First Action Selection
6: Choose action at from st using Improved ϵ-greedy policy
(Note: Incorporate Heuristic Exploration by prioritizing nodes with high network degree)
7: Repeat (for each step of episode):
8: Execute action at, observe reward Rt and new state st+1
9: Step 2: Next Action Selection (On-policy)
10: Determine available actions in new state st+1
11: Choose next action at+1 from st+1 using current policy (Improved ϵ-greedy)
12: Step 3: Q-table Update
13: Q(st,at)←Q(st,at)+α[Rt+γQ(st+1,at+1)−Q(st,at)]
14: Step 4: Transition
15: st←st+1
16: at←at+1
17: Increment time step t
18: Until st is terminal or Algorithm Converged
19: End For
20: Generate Pathopt by selecting the sequence of actions with maximum Q-values starting from s0
21: Return Pathopt

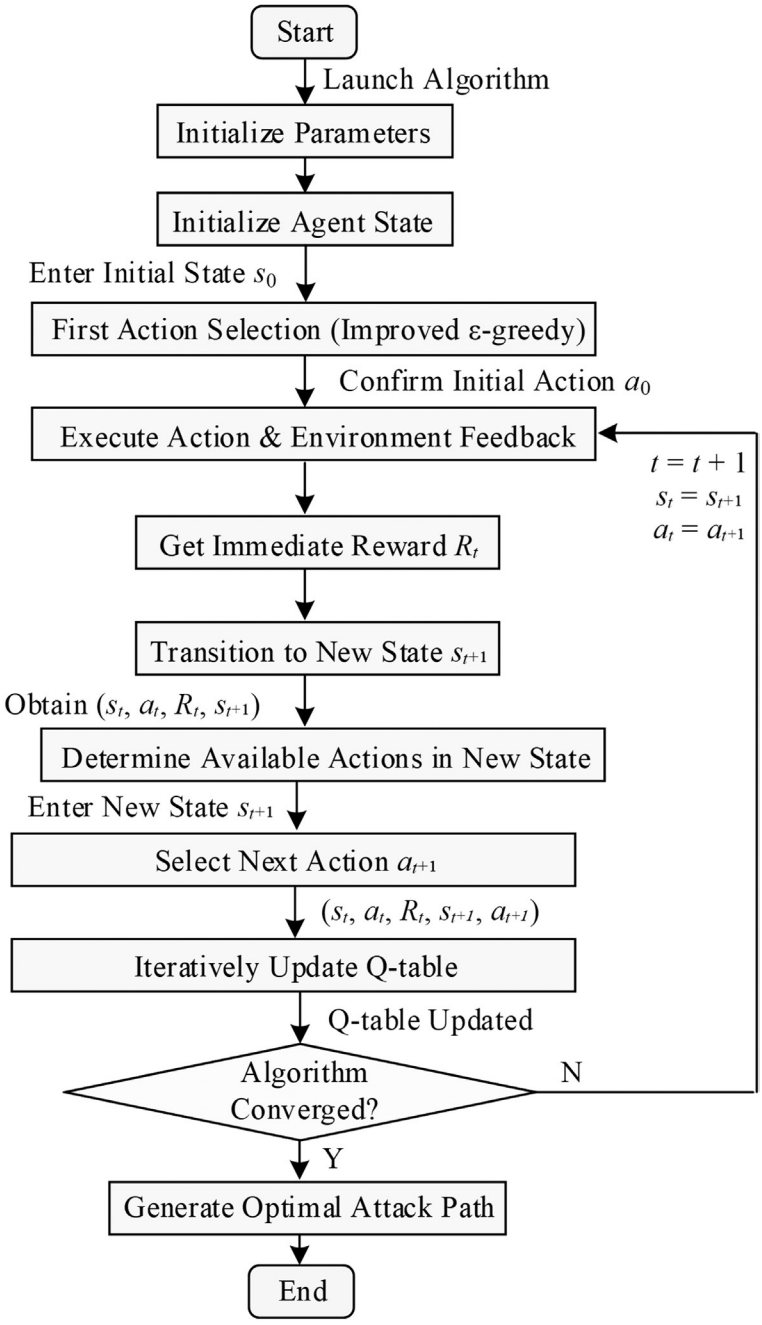


Figure 7. Flow chart of optimal attack path search based on SARSA.

In Figure 7, the algorithm's core idea is to learn an action-value function $V(s, a)$ that evaluates the maximum expected cumulative reward obtainable in the future after executing an action a in a specific state s. Unlike Q-learning, SARSA assesses the value of the current policy being executed rather than the absolute optimal policy value. The agent learns through repeated interactions with the environment [24]. In each interaction, the agent selects an action a_t based on its current policy. After execution, the environment provides an immediate reward R_t and transitions to the next state s_{t+1}. Subsequently, the agent must again select the next action a_{t+1} to execute from the new state s_{t+1} based on its current policy. Utilizing the complete informa-

tion quintuple (s_t, a_t, R_t, s_{t+1}, a_{t+1}), the agent iteratively updates its internal value table via the Bellman equation, progressively refining its valuation of state-action pairs. The update rule is expressed as in Equation (7) [25].

$$V_{new}(s_t, a_t) = (1-\lambda)V(s_t, a_t) + \lambda[R_t + \zeta V(s_{t+1}, a_{t+1})] \quad (7)$$

In Equation (7), λ is the learning rate. ζ is the discount factor. In terms of action selection, research has improved the traditional ε-greedy method. This method will give priority to nodes with larger δ_i in the network to help the agent quickly find high-risk areas. The calculation of the optimal attack strategy is shown in Equation (8).

$$\pi(s,i) = \begin{cases} 1-\varepsilon+\dfrac{\varepsilon}{|I_s|}, & \text{if } i = i^* \\ \dfrac{\varepsilon}{|I_s|}\cdot\dfrac{\delta_i}{\sum_{j\in I_s, j\neq i^*}\delta_j}\cdot(|I_s|-1), & \text{if } i \neq i^* \end{cases} \quad (8)$$

In Equation (8), i^* is the action with the highest current value. I_s is the set of all actions that can be selected in the current state. The SARSA algorithm maintains a Q-value table, the size of which depends on the state space $|S|$ and action space $|A|$. The state space corresponds to the number of nodes n in the attack graph, and the action space corresponds to the maximum node output D_{out}. Therefore, the space complexity is $O(|S| \cdot |A|) \approx O(n \cdot D_{out})$. This is completely within the memory capacity of the onboard computer for a typical subway line A topology. Meanwhile, the time complexity of single step Q-value update is $O(1)$. If the total number of training rounds is H and the average number of steps per round is B, then the total training time complexity is $O(H \cdot B)$.

4. Results and Discussion

To verify the effectiveness of the risk modeling and assessment method proposed in the study, the study conducts simulation verification. First, a simulation environment is constructed, and quantitative experimental analysis of the importance of key nodes and unit attack probability is conducted in this environment. Then, the optimal attack path search performance based on the SARS algorithm is verified and its superiority is verified.

4.1. System Attack Surface Analysis and Risk Quantification Modeling Applications

To ensure the reproducibility of the optimal attack path discovery algorithm proposed in this article, the implementation and training details of the Markov decision process tuples are thoroughly defined. The state space is defined as the set of all nodes in the attack graph. Each state represents the location of the highest privileged node currently held by the attacker. The initial state is randomly set based on external interface nodes. The action space consists of all directed edges starting from the current node. Each action corresponds to a lateral movement technique. At the same time, the experiment includes a composite reward function to balance the effectiveness of the attack. When the intelligent agent successfully captures the target node, it receives a sparse positive reward of +100. When performing any vulnerability exploitation action, deduct the corresponding score based on the availability score of CVSS to simulate the cost of the attack. If touching a honeypot node or a path that is highly likely to be detected by IDS, a heavy penalty (−50) will be imposed. Deduct 1 point for each step to drive the agent to find the shortest path. To prevent falling into local optima, a dynamic greedy strategy is implemented. The exploration rate ϵ is initialized to 1.0 and decreases at a decay rate of 0.995 per round until it reaches the lower limit of 0.05. When the fluctuation amplitude of the cumulative reward's sliding average within 50 consecutive rounds is less than 2%, the model is judged to converge. The study builds a simulation platform to simulate a real cloud computing CBTC environment. The platform uses several physical server nodes to form a private cloud and uses the network to communicate with external vehicle controllers. The configuration parameters of the platform are shown in Table 2.

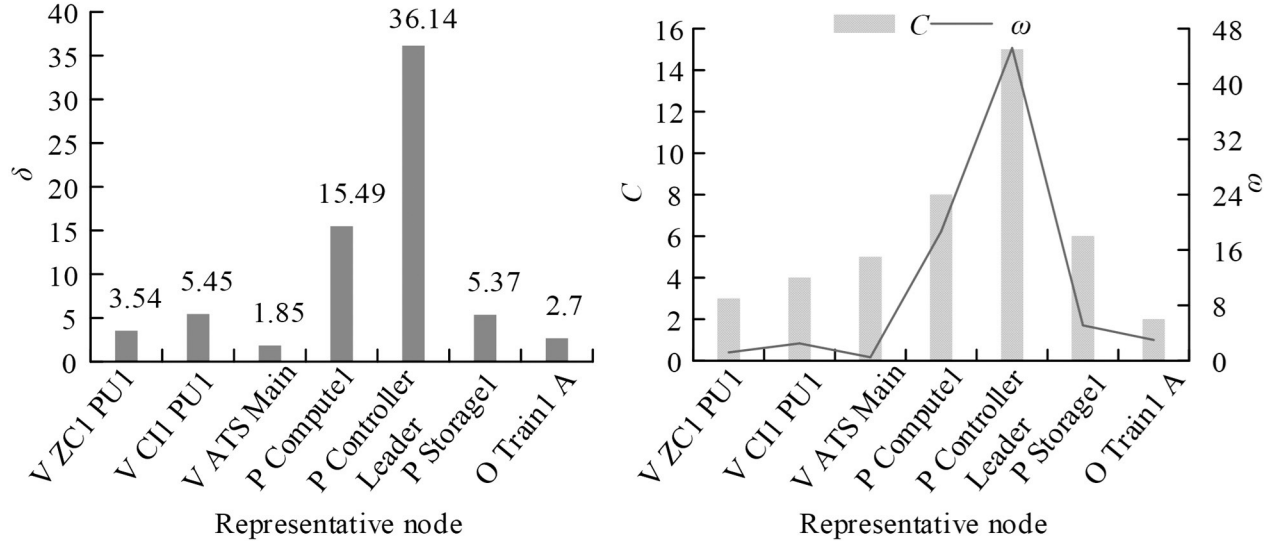
Table 2. Configuration of simulation experiment platform.

Category	Parameter	Configuration/Quantity
Physical resources (P)	Compute node	5 units (Dell R740)
	Control node	3 units (cluster)
	Storage node	2 units (Ceph cluster)
Virtual resources (V)	Operating system	CentOS 7.6/Windows Server 2012
	CI	8
	ZC	8
	ATS	2
Vehicle resources (O)	Vehicle mounted controller	10

Formal statistics are conducted on the structural features of the generated graph model, which contains 86 nodes and 214 directed edges. There is heterogeneity among the nodes, with 40.7% belonging to the network domain and 59.3% belonging to the physical/coupled domain. The network exhibits obvious scale-free characteristics, with an average node degree of 4.9 and a network diameter of 12 hops. At the same time, 12 typical industrial control system vulnerabilities based on CVE database mapping are injected into the figure. About 15% of vulnerabilities are high risk, mainly concentrated in the operating system layer. The proportion of medium risk vulnerabilities is 45%, and the proportion of low-risk vulnerabilities is 40%.

The control weight ω reflects a node's critical role in ensuring MA path connectivity. The node degree δ reflects a node's overall importance. The ω and δ values for each representative node are shown in Figure 8. In Figure 8(a), the P Controller Leader exhibits a δ value as high as 36.14, the highest among all nodes, indicating its status as the most critical node in the entire system. Figure 8(b) reveals the reason. The number of connecting edges C of this node is 15, which is the structural core of the network. At the same time, the node's ω is 45.2, which is far higher than other nodes, indicating that it plays a decisive functional role in the normal transmission of the MA path. The δ of the P Compute1 node is 15.49, and ω is as high as 18.7, ranking second, but C is only 8. This indicates that the node's high importance primarily stems from its functionality rather than the number of network connections. It demonstrates that δ, by integrating ω, can effectively identify critical risk nodes while avoiding interference from pseudo-important nodes.

A probabilistic assessment method based on CVSS analyzes the probability of unit attacks on two critical nodes within the system. Assuming a scan has identified high-severity vulnerabilities, it calculates the final attack success rate. By integrating the logic linking indicator values to attack difficulty in offensive-defensive practice, the method maps enumeration values for AV, AC, PR, and UI to probability coefficients, as shown in Table 3.



(a) The node degrees of each node.

(b) The number of connecting edges and control weights of each node.

Figure 1. The ω and δ of each representative node.

Table 3. CVSS probability mapping.

CVSS index	Enum value	Meaning	Probability coefficient
AV	N	Network (remote attack)	0.85
	A	Adjacent networks	0.65
	L	Local	0.40
	P	Physics	0.15
AC	L	Low	0.77
	H	Tall	0.35
PR	N	None	0.85
	L	Low	0.62
	H	Tall	0.27
UI	N	None	0.85
	R	Need	0.45

Table 4. Calculation results of attack probability for key node units.

Target node	Vulnerability	CVSS index (AV/AC/PR/UI)	Probability coefficient	Probability of single vulnerability exploitation $P(\varepsilon)$	Unit attack probability $P(A_u)$
V ATS Server	Web Framework RCE	N/L/N/N	0.85×0.77×0.85×0.85	0.473	0.473
	Default password for management port	N/L/L/N	0.85×0.77×0.62×0.85	0.345	
P Compute Host 1	virtual machine escape	L/H/L/N	0.40×0.35×0.62×0.85	0.074	0.162
	Dirty pipe	L/L/L/N	0.40×0.77×0.62×0.85	0.162	

The study conducts a unit attack probability analysis on the virtual application server (V ATS Server) and the physical compute host 1 (P Compute Host 1). Table 4 shows that the unit attack probability for the V ATS Server is significantly higher than that for P Compute Host 1. The former's risk is primarily determined by the remote code execution (RCE) vulnerability in the web framework, with a single vulnerability exploitation probability as high as 0.473. In contrast, the vulnerability with the highest exploitation probability on P Compute Host 1, Dirty Pipe, is only 0.162. Among these, AV is the key determinant of attack probability. Both vulnerabilities in V ATS Server are network-accessible (AV=N) with a coefficient of 0.85, indicating attackers can launch attacks directly from remote locations. Conversely, all vulnerabilities in P Compute Host 1 require local access (AV=L) with a coefficient of only 0.40, making the attack prerequisites significantly more stringent. This demonstrates that a node's network exposure is the primary factor determining its initial attack probability.

4.2. Optimal Attack Path Determination and Systemic Risk Assessment

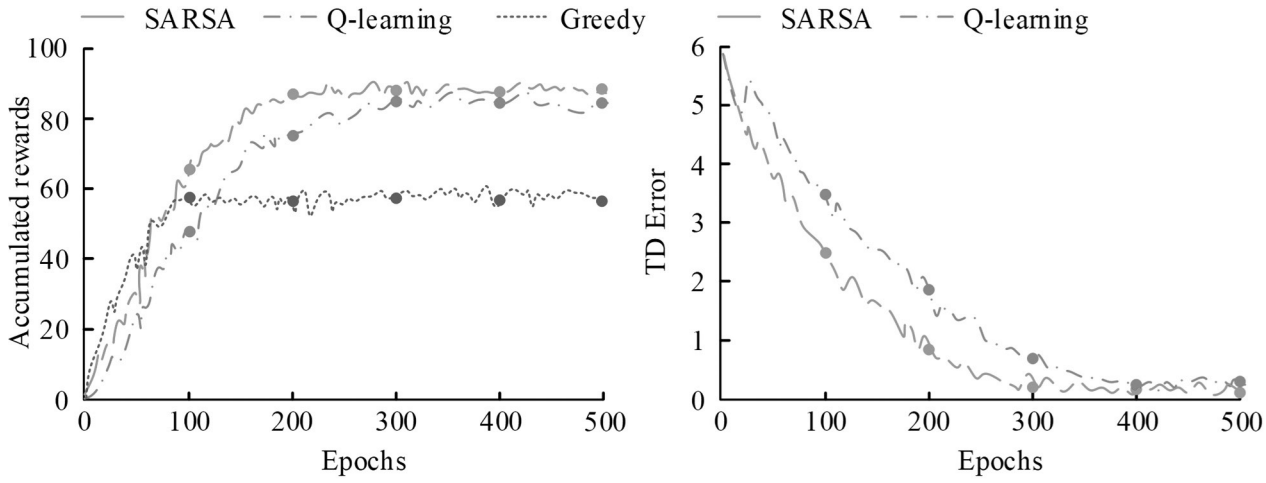
To identify the optimal attack path, the study employs the SARSA algorithm to train the attack model. For comparison, Q-learning and Greedy algorithms are also utilized. The learning rate is set to 0.1, the discount factor to 0.9, and the exploration rate linearly decays from 1.0 to 0.01. The agent is trained for a total of 500 rounds. The study compared the cumulative rewards and temporal difference error (TD Error) changes of each algorithm during the training process. Since the learning mechanism of the Greedy algorithm is very simple, its TD Error is not comparable, so it is not listed. In Figure 9(a), the Greedy algorithm rise rapidly in the initial stage but quickly enters a plateau period at about 50 rounds, and the final cumulative reward is only 58.36. The SARSA algorithm iterates for about 200 rounds and converges to 87.66. The Q-learning algorithm iterates about 300 times before it converges to 85.34. In Figure 9(b), throughout the learning process, the Q-learning curve is always above the SARSA curve, indicating that the prediction error of Q-learning is larger. Moreover, its TD Error finally converges to 0.24. The SARS algorithm finally converges to 0.17. This demonstrates that the SARSA algorithm outperforms the comparison algorithm in terms of learning efficiency, learning stability, and final performance.

After the algorithm converges, the study continues to extract the optimal attack path from the trained value table. The optimal attack paths discovered by the three algorithms are compared in Table 5. The SARSA algorithm identifies the attack path with the highest cumulative risk, reaching 85.4. Q-learning follows with a cumulative risk of 80.6, while the Greedy algorithm achieves only 35.5. The Greedy algo-

rithm becomes trapped in a local optimum in the second step, drawn to a minor objective V CI Backup with a high single-step risk. In contrast, both SARSA and Q-learning correctly identifies the critical sequence for the first two steps: first compromising the entry point V ATS Main, then escaping via the VM to control P Compute1. However, in the third step, SARSA opts to directly attack the system core P Controller Leader, which has the highest single-step risk at that moment. Q-learning choose to attack P Storage1 first. Its subsequent attack on P Controller Leader has lower returns, resulting in the total cumulative risk being slightly lower than SARS. In summary, the SARS algorithm has better risk assessment performance.

To verify the practical value of the evaluation method, the study simulates two system states, including the baseline state and the reinforced state. Among them, the hardened state deploys virtual machine isolation, updates vulnerability patches, and strengthens the password strength of physical nodes. The cumulative risk process in the two states is shown in Figure 10. In Figure 10(a), in the baseline state, the final cumulative risk of the Greedy algorithm is only 35.5. The SARSA and Q-learning algorithms completely overlap in the first two steps, and the cumulative risk reaches 40.6. Starting from the third step, the cumulative risk of the SARSA algorithm rapidly surges and converges to 85.4, while Q-learning increases at a slower pace, ultimately converging to 80.6. In Figure 10(b), under the hardened state, the final cumulative risks of all paths identified by the algorithms are significantly reduced. The SARSA algorithm's risk drops from 85.4 to 31.2, demonstrating the effectiveness of security hardening measures. In summary, the SARSA algorithm outperforms the comparison algorithms in identifying the highest-risk attack paths across different security environments, while also validating the efficacy of the security hardening measures.

The study then uses the Barabási-Albert model in the NetworkX library of the Python programming language to generate five sets of synthetic scale-free networks of different sizes. These networks have node numbers of 100, 300, 500, 800, and 1,000. They simulate the expansion process from a single subway line to a complex urban rail transit network. The scalability test results of algorithms under different network scales are shown in Table 6. Table 6 shows that as the node size increases from 100 to 1,000, the convergence time of the improved SARSA increases from 12.5 to 124.6 s. In contrast, the training time of DQN shows an exponential growth. In a topology with 1,000 nodes, it is difficult for DQN to converge within the specified time due to the explosion of neural network input dimensions and the memory overhead of experience replay. Compared with DQN, the improved SARSA algorithm is more feasible for edge deployment in real large-scale subway networks.

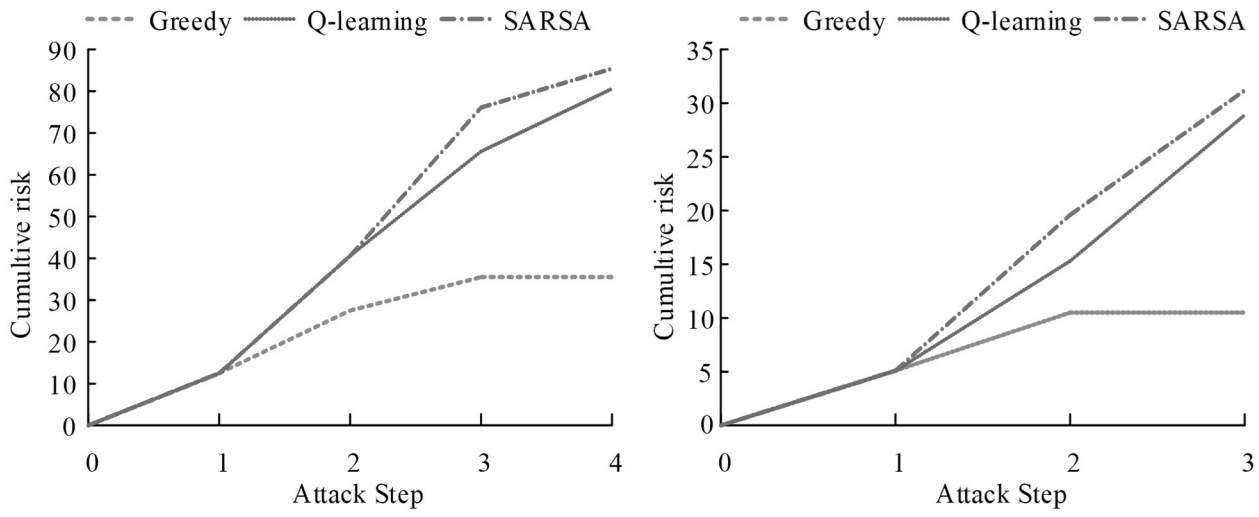


(a) Accumulated rewards. *(b)* TD Error.

Figure 9. Cumulative rewards of different algorithms and TD errors.

Table 5. Comparison of optimal attack paths discovered by three algorithms.

Algorithm	Step	Target node	Attack method	Single step risk	Cumulative risk
Greedy	1	V ATS Main	CVE-2019-0708	12.5	12.5
	2	V CI Backup	Weak password	15	27.5
	3	P Storage1	Unauthorized access to internal services	8	35.5
Q-learning	1	V ATS Main	CVE-2019-0708	12.5	12.5
	2	P Compute1	CVE-2020-1472	28.1	40.6
	3	P Storage1	Unauthorized access to internal services	25	65.6
	4	P Controller Leader	weak password	15	80.6
SARSA	1	V ATS Main	CVE-2019-0708	12.5	12.5
	2	P Compute1	CVE-2020-1472	28.1	40.6
	3	P Controller Leader	weak password	35.5	76.1
	4	P Storage1	Unauthorized access to internal services	9.3	85.4



(a) Baseline status. *(b)* Reinforced state.

Figure 10. Accumulated risk processes under different states.

Table 6. Scalability test results of algorithms under different network scales.

Number of nodes	Edge number	Average convergence time (s)		Average number of convergence rounds	
		SARSA	DQN	SARSA	DQN
100	294	12.5	45.2	320	850
300	890	38.4	186.5	450	1420
500	1490	65.2	412.8	580	2100
800	2390	98.1	985.4	720	3500
1000	2990	124.6	1850.2	860	Not converged

To verify the effectiveness of the various innovative modules proposed in the study, ablation experiments are designed. Q-Learning algorithm is based on Baseline 1, without FTA or heuristics. Baseline 2 is based on DQN emission, without FTA or heuristics. The mutation variant uses the SARSA algorithm but removes the node degrees of FTA and the fusion control weights. These are replaced with standard network centrality indicators. The average convergence steps and high-risk path detection rate of each algorithm are shown in Table 7. According to Table 7, compared to the Mutation Variant, the convergence rounds of the proposed algorithm decreases from 620 to 480, and the high-risk path detection rate increases by 13.2%. The reason is that the standard network centrality index only directs attackers to focus on network hubs. In contrast, the node degree that integrates control weights directs agents to prioritize exploring key actuators with remote topology and serious physical consequences. This approach can more accurately locate the maximum risk path. Meanwhile, Baseline 1 and Baseline 2 only rely on CVSS scores and lack FTA, resulting in them being able to identify network attack paths but missing specific logical cut sets that lead to physical accidents. After introducing FTA, the model can effectively understand the logic of cross domain collaborative attacks and improve the detection rate. In addition, although DQN has a higher detection rate than Q-Learning, its computational cost is enormous. In contrast, the improved SARSA proposed in the study has the fastest convergence rate and the highest high-risk path detection rate (94.6%), demonstrating significant advantages.

5. Summary

To evaluate the cyber-physical coupling security risks of the CBTC system in the cloud computing environment, the study integrated the complex network, fault tree, and attack graph theories to construct a risk model that can quantify the node importance and attack probability. Meanwhile, the SARS algorithm was introduced to find the attack path with the highest risk. The results showed that the node degree δ of the physical master node P Controller Leader was as high as 36.14, which was significantly larger than of other nodes. It could help identify critical infrastructure nodes that had the greatest impact on physical security. In the performance comparison of the evaluation algorithms, compared with Q-learning and Greedy algorithms, the SARS algorithm could converge to the highest cumulative reward faster, with a cumulative

Table 7. The average convergence steps and high-risk path detection rate of each algorithm.

Experimental group	Algorithm core	Explore inspiring indicators	Average convergence rounds	High risk path detection rate
Baseline 1	Q-Learning	Random	850	72.5%
Baseline 2	DQN	ϵ-Greedy	1200	88.1%
Ablation variant	SARSA	Standard network centrality	620	81.4%
Proposed method	Improved SARSA	Node degree of fusion control weight	480	94.6%

reward as high as 87.66. Meanwhile, the TD Error of this algorithm was always lower than Q-learning during training, and finally converged to 0.17, which was lower than the other comparison algorithms. It showed that the algorithm had high efficiency and stability. On this basis, the cumulative risk of the optimal attack path extracted by the algorithm was as high as 85.4, which was higher than of the comparative algorithm. In actual applications, after deploying measures such as virtual machine isolation and patch updates, the maximum risk value faced by the system was reduced from 85.4 to 31.2, a decrease of 63.5%. Intuitively quantifies the effective return on security investment. It showed that the cyber-physical coupling risk modeling method proposed in the study could effectively identify the key risk nodes and attack paths of the CBTC system in the cloud computing environment. Meanwhile, the SARSA algorithm performed well in solving such optimal attack path optimization problems.

6. Limitations and Future Work

Although the method proposed in this article demonstrates superior performance in simulated environments, there are still certain limitations in terms of model assumptions, data dependencies, and experimental environments. First, the study models the attacker's lateral movement as a series of discrete actions. It assumes that, once an attack is successful, the attacker gains complete control of the node. However, in real advanced persistent threat scenarios, attack behavior has a high degree of temporal persistence, concealment, and uncertainty. The current SAG model has not fully captured these complex dynamic features. Second, the risk quantification module highly relies on the logical structure definition of FTA and CVSS vulnerability rating data. The construction of FTA usually requires profound domain expert knowledge, which can easily introduce subjective bias. However, the universal CVSS score may not fully reflect the actual difficulty of utilization in specific industrial scenarios. In addition, due to legal and ethical considerations for the safe operation of critical infrastructure, research cannot conduct penetration testing on real operating subway lines. The experimental results are entirely based on a simulation environment that complies with the IEEE 1474.1 standard. The robustness of the algorithm in practical deployment is affected by the difficulty of fully reproducing the hardware fingerprints, network jitter, and complex background traffic noise of real physical devices.

In future research, the plan is to utilize the feature extraction capabilities of graph convolutional networks or graph attention networks to achieve end-to-end learning and generalization of large-scale, dynamically changing network topologies. Meanwhile, this study considers expanding from a single-attack perspective to a dual-layer game model of attack and defense. It investigates how defense agents can dynami-

cally generate optimal network isolation or traffic cleaning strategies based on predicted attack paths. Finally, it considers combining threat intelligence with honeypot log data to dynamically adjust CVSS scores and FTA weights, thereby reducing reliance on static expert knowledge.

Declaration of Competing Interests

The authors declare no conflict of interest.

Funding

This research received no funding.

Data Availability

Data used in this study are proprietary.

References

[1] L. Zhu *et al.*, "Machine Learning in Urban Rail Transit Systems: A Survey", *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 3, pp. 2182–2207, 2023. http://dx.doi.org/10.1109/TITS.2023.3319135

[2] R. Nagy *et al.*, "Innovative Approaches in Railway Management: Leveraging Big Data and Artificial Intelligence for Predictive Maintenance of Track Geometry", *Tehnički vjesnik*, vol. 31, no. 4, pp. 1245–1259, 2024. http://dx.doi.org/10.17559/TV-20240420001479

[3] S. Ma *et al.*, "Joint Security and Resilience Control in IIoT-Based Virtual Control Train Sets Under Jamming Attacks", *IEEE Transactions on Vehicular Technology*, vol. 72, no. 9, pp. 11196–11212, 2023. http://dx.doi.org/10.1109/ICCC62609.2024.10941915

[4] Q. Zhang *et al.*, "A Holistic Solution to Virtual Coupling Based Urban Rail Train Control System", *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 238, no. 6, pp. 603–615, 2024. http://dx.doi.org/10.1177/09544097231215519

[5] X. Ge *et al.*, "Secure Virtual Coupling Control of Connected Train Platoons Under Cyber Attacks", *Vehicle System Dynamics*, vol. 63, no. 1, pp. 93–120, 2025. http://dx.doi.org/10.1080/00423114.2024.2410909

[6] X. Y. Kong *et al.*, "Secure State Estimation for Train-to-Train Communication Systems: A Neural Network-Aided Robust EKF Approach", *IEEE Transactions on Industrial Electronics*, vol. 71, no. 10, pp. 13092–13102, 2024. http://dx.doi.org/10.1109/TIE.2023.3347834

[7] Y. Guo *et al.*, "A Particle Swarm Optimization-Based Online Optimization Approach for Virtual Coupling Trains with Communication Delay", *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 6, pp. 49–63, 2023. http://dx.doi.org/10.1109/MITS.2023.3293760

[8] Y. Liu *et al.*, "Efficient Master Production Scheduling for Manufacturing Systems Using an Enhanced SARSA Algorithm", *International Journal of Simulation and Process Modelling*, vol. 22, no. 1–2, pp. 60–74, 2025. http://dx.doi.org/10.1504/IJSPM.2025.148294

[9] F. Peng *et al.*, "A SARSA Reinforcement Learning Hybrid Ensemble Method for Robotic Battery Power Forecasting", *Journal of Central South University*, vol. 30, no. 11, pp. 3867–3880, 2023. http://dx.doi.org/10.1007/s11771-023-5451-0

[10] H. Lu *et al.*, "A Transfer Learning-Based Intrusion Detection System for Zero-Day Attack in Communication-Based Train Control System", *Cluster Computing*, vol. 27, no. 6, pp. 8477–8492, 2024. http://dx.doi.org/10.1007/s10586-024-04376-9

[11] X. Yang *et al.*, "Vehicle Driving Behavior Recognition and Optimization Strategies Based on Cloud Computing and SSA-BP Algorithm", *Studies in Informatics and Control*, vol. 33, no. 3, pp. 17–28, 2024. http://dx.doi.org/10.24846/v33i3y202402

[12] H. Du *et al.*, "An Improved Ant Colony Algorithm for New Energy Industry Resource Allocation in Cloud Environment", *Tehnički vjesnik*, vol. 30, no. 1, pp. 153–157, 2023. http://dx.doi.org/10.17559/TV-20220712164019

[13] X. Ge *et al.*, "Resilient Virtual Coupling Control of Automatic Train Convoys with Intermittent Communications", *IEEE Transactions on Vehicular Technology*, vol. 73, no. 5, pp. 6183–6195, 2023. http://dx.doi.org/10.1109/TVT.2023.3339854

[14] H. Dabbaghzadeh *et al.*, "CBTC Security and Reliability Enhancements by a Key-Based Direct Sequence Spread Spectrum Technique", *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 1, pp. 159–172, 2023. http://dx.doi.org/10.1109/TITS.2023.3312266

[15] B. Sun *et al.*, "Reliability Analysis of CTCS-3 Train-Ground Communication System Based on 5G-R", *IEEE Transactions on Vehicular Technology*, vol. 72, no. 10, pp. 12927–12940, 2023. http://dx.doi.org/10.1109/TVT.2023.3272019

[16] K. Ko *et al.*, "Field Verification of Wireless Cellular Communication-Based Subway Train Localization", *IEEE Transactions on Vehicular Technology*, vol. 73, no. 6, pp. 7681–7692, 2024. http://dx.doi.org/10.1109/TVT.2024.3355888

[17] Z. Song *et al.*, "Stacked Denoised Auto-Encoding Network-Based Kernel Principal Component Analysis for Cyber Physical Systems Intrusion Detection in Business Management", *Computer Science and Information Systems*, vol. 21, no. 4, pp. 1725–1743, 2024. http://dx.doi.org/10.2298/CSIS240314055S

[18] A. Gligor *et al.*, "Augmented Cyber-Physical Model for Real-Time Smart-Grid Co-Simulation", *International Journal of Computers, Communications and Control*, vol. 20, no. 1, 2025. http://dx.doi.org/10.15837/ijccc.2025.1.6914

[19] S. Cao *et al.*, "Robust Offline Actor-Critic with On-Policy Regularized Policy Evaluation", *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 12, pp. 2497–2511, 2024. http://dx.doi.org/10.1109/JAS.2024.124494

[20] M. S. Rais *et al.*, "Decision Making for Autonomous Vehicles in Highway Scenarios Using Harmonic SK Deep SARSA", *Applied Intelligence*, vol. 53, no. 3, pp. 2488–2505, 2023. http://dx.doi.org/10.1007/s10489-022-03357-y

[21] H. R. Xing *et al.*, "Optimizing Human-Machine Systems in Automated Environments", *International Journal of Simulation Modelling (IJSIMM)*, vol. 23, no. 4, 2024. http://dx.doi.org/10.2507/IJSIMM23-4-CO19

[22] Z. Jiang *et al.*, "A Graph-Based PPO Approach in Multi-UAV Navigation for Communication Coverage", *International Journal of Computers Communications & Control*, vol. 18, no. 6, 2023. http://dx.doi.org/10.15837/ijccc.2023.6.5505

[23] H. U. Gobbi *et al.*, "Comparing Reinforcement Learning Algorithms for a Trip Building Task: A Multi-Objective Approach Using Non-Local Information", *Computer Science and Information Systems*, vol. 21, no. 1, pp. 291–308, 2024. http://dx.doi.org/10.2298/CSIS221210072G

[24] Z. H. Wei *et al.*, "Optimizing Production with Deep Reinforcement Learning", *International Journal of Simulation Modelling (IJSIMM)*, vol. 23, no. 4, 2024. http://dx.doi.org/10.2507/IJSIMM23-4-C017

[25] R. Hu *et al.*, "Autonomous Driving Decision-Making Based on an Improved Actor-Critic Algorithm", *Studies in Informatics and Control*, vol. 33, no. 4, pp. 37–50, 2024. http://dx.doi.org/10.24846/33i4y202404

Received: November 2025
Revised: December 2025
Accepted: December 2025

Contact addresses:
Lin Deng
Sichuan Vocational and Technical College
Suining City
Sichuan Province
China
e-mail: yashifc@126.com

Zhao Ping*
Sichuan Vocational and Technical College
Suining City
Sichuan Province
China
e-mail: zhaoping667788@163.com
*Corresponding author

Hui Xiong
Sichuan Vocational and Technical College
Suining City
Sichuan Province
China
e-mail: xionghui811129@163.com

Lin Deng received his master's degree in computer application technology from Southwest Jiaotong University, China. He currently holds the position of lecturer at Sichuan Vocational and Technical College, China. His primary research interest lies in artificial intelligence.

Zhao Ping received his bachelor's degree in computer science and technology from Mianyang Teachers' College, China. He currently holds the position of lecturer at Sichuan Vocational and Technical College, China. His primary research interest lies in artificial intelligence and big data.

Hui Xiong his master's degree in computer application technology from the XinJiang Technical Institute of Physics & Chemistry, CAS, China. He currently holds the position of lecturer at Sichuan Vocational and Technical College, China. His primary research interest lies in artificial intelligence.