An Intelligent Penetration Strategy for Power System Networks Using Reinforcement Learning

Manpo Li[1], Ning Yang[1], Xuerui Yang[1], Xuezhu Jin[1], Long Yin[2,*] and Jian Xu[2]

[1]*Northeast Branch of State Grid Corporation of China, Shenyang 110180, China*
[2]*Software College, Northeastern University, Shenyang 110169, China*
E-mail: 2110499@stu.neu.edu.cn
[*]*Corresponding Author*

Received 10 August 2025; Accepted 06 November 2025

Abstract

Cybersecurity is vital for modern power systems, which are increasingly exposed to sophisticated cyber threats. Penetration testing is an effective method for identifying system vulnerabilities by simulating real-world attacks. However, traditional approaches depend heavily on expert knowledge and manual effort, resulting in high labor and time costs. To address this, we propose an autonomous penetration testing framework tailored for power system networks. The problem is modeled as a Markov Decision Process (MDP) and solved using an enhanced deep reinforcement learning algorithm. Specifically, we introduce SPIND-DQL, which integrates NoisyNet, Dueling Architecture, Prioritized Experience Replay (PER), Intrinsic Curiosity Module (ICM), and Soft Q-Learning to improve exploration efficiency and reduce trial-and-error during training. Experiments conducted in Microsoft's CyberBattleSim, adapted to reflect power system network environments, show that SPIND-DQL achieves up to 40% faster convergence and compromises 25% more assets compared to baseline DQN variants and strong

Journal of Cyber Security and Mobility, Vol. 14_5, 1221–1244.
doi: 10.13052/jcsm2245-1439.1458

baselines like Rainbow DQN. Our ablation studies confirm the significant contribution of each component, particularly ICM and Soft Q-Learning, in discovering complex attack paths. This highlights its potential as a practical and intelligent tool for power system cybersecurity assessment.

Keywords: Penetration testing, Reinforcement learning, Cybersecurity, DQN algorithm.

1 Introduction

The rapid digital transformation and increased connectivity of modern power systems driven by the integration of smart grids, distributed energy resources (DERs), and advanced communication technologies (ACTs) have significantly improved operational efficiency and reliability.

However, this growing interconnectivity, particularly the convergence of Information Technology (IT) and Operational Technology (OT), also exposes power infrastructures to a wide range of cybersecurity threats, such as advanced persistent threats (APTs), data manipulation, denial-of-service (DoS) attacks, and false data injection (FDI) attacks.These threats can compromise system integrity, disrupt services, and, in extreme cases, cause cascading failures across critical energy networks by translating cyber intrusions into physical disruptions.

To ensure the security and resilience of such cyber-physical systems, penetration testing has emerged as a vital technique for proactively identifying vulnerabilities before they are exploited by malicious actors. Traditional penetration testing approaches, however, depend heavily on manual operations and domain-specific expertise, making them costly, time-consuming, and challenging to scale. As power systems evolve into increasingly complex and dynamic environments, there is a pressing need for more intelligent, adaptive, and automated penetration testing solutions.

Reinforcement Learning is particularly well-suited for this problem compared to traditional optimization (e.g., Mixed-Integer Linear Programming) or other AI approaches like Genetic Algorithms (GAs). Unlike static optimization methods, RL excels in sequential decision-making under uncertainty and in partially observable environments, which precisely describes a penetration test. While GAs can find diverse attack paths, RL's value-function approach allows it to learn and refine optimal strategies (policies) that adapt to the environment's state, rather than just finding a single optimal path for a static configuration. Recent advancements in reinforcement learning

(RL), particularly Deep Reinforcement Learning (DRL), offer promising opportunities to address these limitations.

RL agents are capable of learning optimal strategies through interaction with the environment, making them ideal for modeling attack behaviors in dynamic and partially observable network scenarios. While prior research has extensively applied DRL in power systems for tasks such as resource allocation, fault recovery, anomaly detection, and cyber defense, significantly less attention has been given to modeling attacker behavior and simulating automated penetration processes, especially within the unique operational constraints and network topologies of power systems.

To fill this gap, we propose a reinforcement learning-based automated penetration testing tool tailored specifically for power system network environments. By formulating the penetration process as a Markov Decision Process (MDP), we design an enhanced Deep Q-Network (DQN) framework, SPIND-DQL, that integrates multiple optimization techniques, including NoisyNet, Dueling Architecture, Prioritized Experience Replay (PER), Intrinsic Curiosity Module (ICM), and Soft Q-Learning. While the first three components are hallmarks of the 'Rainbow' DQN agent, our primary contribution lies in augmenting this strong baseline with ICM and Soft Q-Learning. We posit that this specific combination is uniquely suited for penetration testing: ICM provides intrinsic motivation to explore sparsely-rewarded attack paths, while Soft Q-Learning's entropy-based regularization encourages a stochastic policy capable of escaping sub-optimal attack patterns and adapting to dynamic defenses.

We implement and evaluate our proposed approach within Microsoft's CyberBattleSim simulation framework, extending it to reflect the specific characteristics of power system architectures. Experimental results show that the SPIND-DQL agent demonstrates superior performance in convergence speed, reward accumulation, and penetration effectiveness when compared to baseline DQN variants and the Rainbow DQN baseline. The main contributions of this paper are summarized as follows:

- We model the autonomous penetration testing problem in power system networks as a Markov Decision Process (MDP), providing a formal specification of the state, action, and reward spaces.
- We propose SPIND-DQL, a novel DRL framework that integrates Dueling, NoisyNet, PER, ICM, and Soft Q-Learning to efficiently navigate the large exploration space of cyber-attack simulations.

- We conduct a comprehensive evaluation in an adapted CyberBattleSim environment, comparing SPIND-DQL against multiple DQN baselines, a strong Rainbow DQN baseline, and a traditional heuristic attacker.
- We perform a full ablation study to validate the performance contribution of each component of SPIND-DQL, confirming the efficacy of integrating ICM and Soft Q-Learning.

This research not only contributes a novel offensive RL methodology for power system security assessment but also supports the development of adaptive and intelligent red teaming tools that can guide the design of more robust cyber defense strategies.

The remainder of this paper is organized as follows. Section 2 reviews related works in DRL for power systems and cybersecurity. Section 3 details the proposed methodology, including the MDP formulation and the components of SPIND-DQL. Section 4 describes the experimental setup, environment adaptation, and presents the results, including baseline comparisons and ablation studies. Section 5 discusses the implications and limitations of our findings. Finally, Section 6 concludes the paper.

2 Related Works

Recent research in network intrusion detection has increasingly focused on the adaptability and learning capabilities of deep reinforcement learning, particularly advanced Deep Q-Network (DQN) architectures. Scholars have moved beyond the original DQN to leverage more sophisticated variants. For instance, some work explores a unified Rainbow DQN approach, which integrates components like Dueling DQN, Double DQN, and Prioritized Experience Replay (PER) into a single, powerful model to enhance detection accuracy on benchmark datasets [17]. Other studies construct custom-named models, such as Enhanced-Dueling DQN (EDDQN), which similarly combine Dueling, Double, and PER to mitigate overestimation bias and improve sample efficiency, applying them to critical cyber-physical systems [1]. In parallel, a distinct research thrust involves creating hybrid models. This includes integrating DQN with heuristic learning models to specifically combat zero-day attacks in IoT environments [18] and developing hybrid systems that merge DQN with optimization algorithms for both detection and mitigation [7]. Further foundational work explicitly implements and compares Dueling and Double DQN architectures to maximize the cumulative reward for detecting intrusions [3].

The advances have seen growing use of deep reinforcement learning (DRL) in addressing communication and cybersecurity challenges in resilient power systems. Elsayed et al. [6] applied DQN to resource allocation in dense networks, achieving up to 66% and 33% latency reductions over PF and DIRA, while enhancing throughput and fairness for delay-sensitive services. Zhang et al. [23] integrated reinforcement learning with tree search to optimize PMU placement, effectively reducing the number of units and focusing on vulnerable nodes while managing large-scale search spaces. Further, recent work explores composite distributed learning for multi-agent systems [14] and model reusability [15], highlighting trends in scalable and efficient RL.

In transmission system recovery, Wei et al. [19] used DDPG to dynamically adjust reclosing times for failed lines, mitigating cascading failures and outperforming traditional schemes. For demand-side management, Zhang et al. [22] introduced a TSK-fuzzy RL approach using DDPG to counter FDI attacks, balancing economic and operational stability under uncertain threats. Similarly, Etezadifar et al. [8] designed a non-intrusive load monitoring (NILM) scheme using RL with dual replay memory and feedback to improve detection accuracy without accessing user data. Zhang et al. [21] proposed a distributed DRL-based defense for FDI attacks in demand response systems. Their two-stage framework, coupled with a regularized RLS method, improved microgrid resilience and autonomy. In cyber-physical simulation, Sahu et al. [16] combined OpenDSS and SimPy to train RL agents for network reconfiguration and voltage control, advancing control robustness in distribution systems.

Other related works have focused on finite-time stable controllers for uncertain systems [2] and knowledge-constrained clustering for anomaly detection [24]. Multi-agent reinforcement learning (MARL) has also been leveraged to improve security. Zeng et al. [20] presented an adversarial MARL scheme for demand response, reducing ramping by 38.85% while improving system robustness. Fard et al. [10] tackled adversarial FGSM attacks in MADRL-based transmission using a DQN-enhanced feedback control approach. Chen et al. [4] introduced a decentralized A3C-based secondary control for microgrids with heterogeneous BESSs, offering both SoC balancing and defense against DoS attacks with reduced communication overhead. Guo et al. [11] used minimax Q-learning to construct a game-theoretic response to dynamic load-altering attacks (D-LAAs), achieving stronger adaptability over static defenses. Huang et al. [13] reviewed RL's role in enhancing cyber resilience and proposed adaptive response

mechanisms against both known and zero-day threats. This gap is notable, as related security fields advance with concepts like evolutionary defense [12] and vSIM for decoupling device identity [5].

Despite the diversity of defensive strategies, most prior work emphasizes optimization, recovery, and protection. In contrast, research on offensive strategies-particularly automated penetration testing in power systems-is limited. Penetration testing is essential for uncovering latent vulnerabilities but remains manual, expert-dependent, and costly. To address this gap, reinforcement learning offers a promising direction for automating penetration testing in power networks. Such approaches not only enable scalable attack modeling but also support the design of dynamic, game-theoretic cyber defense architectures with both theoretical and practical significance.

3 Methods and Materials

3.1 MDP Formulation and Threat Model

We model the penetration testing problem as a Markov Decision Process (MDP), defined by the tuple (S, A, R, P, γ), where S is the state space, A is the action space, R is the reward function, P is the transition probability, and γ is the discount factor.

State Space (S): The state $s_t \in S$ is a graph-based representation of the agent's knowledge of the network. It includes the set of discovered nodes, their known vulnerabilities, available credentials, and the agent's current access level on each node (e.g., none, user, administrator). The state is partially observable, as the agent only knows what it has discovered.

Action Space (A): As detailed in Table 1, the action space A consists of discrete operations an attacker can perform. These include reconnaissance (scanning), local exploitation (listing vulnerabilities), lateral movement (using credentials), and system modification (terminating services).

Reward Function (R): The reward function $R(s, a)$ guides the agent's learning. It is a composite of external rewards r^{ext} and intrinsic rewards r^{int}.

- **External Reward (r^{ext}):** This is defined by the environment. As shown in Table 2, actions have associated costs (e.g., scanning costs -10). A large positive reward is given for successfully compromising high-value targets (e.g., the PI server or DNP nodes), while a small positive reward is given for discovering new vulnerabilities or gaining privileges.

Table 1 Action definitions and their associated effects

Action Category	Operation Description	Resulting Effect
Remote Intrusion	Host Scanning	Identifies reachable devices
Local Exploitation	Enumerate Vulnerabilities	Extracts sensitive credentials
Authentication Attempt	Use Credentials	Gains elevated access
Network Sweep	Perform Scan	Network mapping
System Recovery	Restore from Image	Resets target system
	Update Firewall Rules	Alters access permissions
Configuration Adjustment	Change Vulnerability Status	Updates asset risk profile
	Terminate Service	Disables application component
Idle State	No Operation	Waits without acting

- **Intrinsic Reward (r^{int}):** This is generated by the Intrinsic Curiosity Module (ICM) to encourage exploration, as detailed in Section 3.4. The total reward is $r_t = r_t^{ext} + r_t^{int}$.

Threat Model: We assume a 'gray-box' attacker model. The agent begins with an **initial foothold** on a single perimeter node. The attacker has **partial observability** of the network and must use reconnaissance actions to discover the topology and vulnerabilities. **Constraints** include a limited action budget (100 steps per episode) and the action costs defined in the reward function, which penalize inefficient or "noisy" attack paths.

3.2 Soft Q-Learning

Soft Q-Learning extends standard Deep Q-Learning by incorporating entropy regularization to promote policy stochasticity and exploration. Unlike conventional Q-learning, which aims to maximize expected rewards:

$$y^{\text{DQN}} = r + \gamma \max_{a'} Q(s', a'; \theta^-) \tag{1}$$

Soft Q-Learning introduces an entropy-weighted term to the target:

$$y^{\text{SoftQ}} = r + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s')} \left[Q(s', a') - \alpha \log \pi(a'|s') \right] \tag{2}$$

Here, α balances reward and entropy, encouraging exploration by penalizing overly confident policies. The optimal policy is derived by maximizing both cumulative rewards and entropy:

$$\pi^* = \arg\max_{\pi} \sum_t \mathbb{E}_{(s_t, a_t) \sim \pi} \left[r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)) \right] \tag{3}$$

This framework results in more adaptive behavior, particularly effective in environments with uncertain or diverse action outcomes. It also underpins algorithms like Soft Actor-Critic (SAC), which integrate this principle into an actor-critic architecture.

3.3 Dueling DQN

Dueling DQN improves upon standard Deep Q-Networks by restructuring the Q-value estimation into two distinct parts: a state-value function $V(s)$ and an advantage function $A(s, a)$ that reflects the impact of each action. These are combined as follows:

$$Q(s,a) = V(s) + \left(A(s,a) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s,a') \right) \tag{4}$$

This formulation normalizes the advantage function, promoting stable learning. By decoupling state evaluation from specific actions, Dueling DQN performs well in scenarios where many actions yield similar results. It enhances generalization and accelerates training by focusing on the intrinsic value of states.

3.4 Prioritized Experience Replay

Prioritized Experience Replay (PER) enhances Deep Q-Learning by biasing sampling toward transitions that are expected to yield greater learning progress. Unlike standard DQN, which uniformly samples experiences from the replay buffer, PER ranks them by the magnitude of their temporal-difference (TD) error:

$$p_i = |\delta_i| + \varepsilon, \quad \delta_i = r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \tag{5}$$

Here, ε ensures all samples have non-zero priority. Transitions are drawn according to:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \tag{6}$$

where α controls how strongly prioritization affects sampling-$\alpha = 0$ corresponds to uniform sampling. To address the sampling bias, PER introduces importance-sampling weights:

$$w_i = \left(\frac{1}{N \cdot P(i)} \right)^\beta \tag{7}$$

with N as the buffer size and β adjusting the degree of correction. These weights rescale the TD loss to maintain unbiased learning. By prioritizing transitions with higher TD errors, PER accelerates learning and improves sample efficiency over uniform replay.

3.5 Intrinsic Curiosity Module

Intrinsic Curiosity Module (ICM) enhances Deep Q-Learning by adding an internal motivation signal to encourage exploration, especially in environments with sparse rewards. Instead of relying only on external rewards r^{ext}, ICM introduces an intrinsic reward r^{int} derived from the agent's prediction error. ICM contains two components: an inverse model that predicts the action a_t taken between consecutive states s_t and s_{t+1}:

$$\hat{a}_t = f_{\text{inv}}(s_t, s_{t+1}) \tag{8}$$

and a forward model that predicts the next state's features representation $\hat{\phi}(s_{t+1})$ based on the current state $\hat{\phi}(s_t)$ and action a_t:

$$\hat{\phi}(s_{t+1}) = f_{\text{fwd}}(\phi(s_t), a_t) \tag{9}$$

The intrinsic reward measures the discrepancy between predicted and actual next state features:

$$r_t^{\text{int}} = \eta \left\| \hat{\phi}(s_{t+1}) - \phi(s_{t+1}) \right\|^2 \tag{10}$$

where η controls the reward scale. The total reward used for learning combines external and intrinsic parts:

$$r_t^{\text{total}} = r_t^{\text{ext}} + r_t^{\text{int}} \tag{11}$$

By incentivizing the agent to reduce prediction errors, ICM promotes exploring novel or unexpected states, improving performance in challenging sparse-reward scenarios.

3.6 Noisy DQN

Noisy DQN improves exploration in Deep Q-Learning by introducing learnable noise into network weights, replacing the fixed ϵ-greedy strategy. Instead of relying on externally controlled randomness, it embeds stochasticity directly into the Q-network using noisy linear layers. A noisy layer modifies

the standard linear transformation $y = Wx + b$ with:

$$y = (\mu^W + \sigma^W \odot \epsilon^W)x + \mu^b + \sigma^b \odot \epsilon^b \tag{12}$$

where μ and σ are learnable parameters representing means and standard deviations, and ϵ is noise sampled from a fixed distribution (e.g., Gaussian). The operator $\odot$ denotes element-wise multiplication. Action selection becomes stochastic via:

$$a_t = \arg\max_a Q(s_t, a; \theta, \epsilon) \tag{13}$$

with noise ϵ resampled at each step. This method enables dynamic, state-aware exploration and allows the model to adjust the level of randomness during training. Over time, as the agent learns, the noise scale σ is optimized to focus more on exploitation.

3.7 Proposed Method

Algorithm 1 presents an enhanced Deep Q-Network (DQN) incorporating NoisyNet, Dueling Architecture, Prioritized Experience Replay (PER), Intrinsic Curiosity Module (ICM), and Soft Q-Learning to improve exploration, learning efficiency, and stability.

The Noisy Dueling Q-Network separates value and advantage estimation and injects learnable noise to promote exploration. A target network ensures stable training, while ICM provides intrinsic rewards based on forward prediction error. PER prioritizes transitions by temporal-difference (TD) error to focus updates on informative experiences. Actions are selected through a softmax-based policy $\pi(a|s)$, which balances exploration and exploitation.

The total reward includes both extrinsic and intrinsic components r^{int}. Soft Q-learning uses the backup target y. Specifically, the Soft Q-learning target y (line 9) uses the maximum entropy backup, replacing the 'hard' $\max$ of standard DQN with a 'soft' log-sum-exp function, which encourages a more stochastic policy.

TD errors define PER priorities p, and sampling weights adjust for bias. The total loss combines Q-value regression and ICM objectives: $\mathcal{L} = \mathcal{L}_Q + \lambda \cdot (\beta \cdot \mathcal{L}_{\text{fwd}} + (1 - \beta) \cdot \mathcal{L}_{\text{inv}})$. Parameters are updated by gradient descent, and the target network is periodically synchronized. This integrated framework supports efficient, informed, and stable reinforcement learning.

Algorithm 1 Improved DQN with Dueling, NoisyNet, PER, ICM, Soft Q-Learning

Init noisy dueling Q-network $Q(s, a; \theta, \epsilon)$, target network Q', ICM (f_{inv}, f_{fwd}), PER buffer $\mathcal{B}$
for each episode **do**
  Init state s
  while not done **do**
    Compute $Q(s, a)$ via noisy net, $\pi(a|s) \propto \exp(Q(s, a)/\alpha)$
    Sample $a \sim \pi(a|s)$, execute a, get r^{ext}, s'
    ICM: $\phi = \phi(s), \phi' = \phi(s')$, compute $r^{int} = \eta \|\phi' - f_{fwd}(\phi, a)\|^2$
    $r = r^{ext} + r^{int}$
    Target: $y = r + \gamma\alpha \log \sum_{a'} \exp(Q'(s', a')/\alpha)$
    TD error: $\delta = y - Q(s, a)$, priority $p = |\delta| + \epsilon$
    Store (s, a, r, s') in $\mathcal{B}$ with p
    if ready to train **then**
      Sample batch from $\mathcal{B}$ w/ weights w_i
      $\mathcal{L}_Q = \sum w_i (Q(s, a) - y)^2$
      $\mathcal{L}_{inv} = CE(f_{inv}(\phi, \phi'), a), \mathcal{L}_{fwd} = \|f_{fwd}(\phi, a) - \phi'\|^2$
      $\mathcal{L} = \mathcal{L}_Q + \lambda(\beta\mathcal{L}_{fwd} + (1-\beta)\mathcal{L}_{inv})$
      Update θ, ϕ via $\nabla_\theta \mathcal{L}$
      Periodically update target: $\theta^- \leftarrow \theta$
    end if
    $s \leftarrow s'$
  end while
end for

4 Experiments

4.1 Network Scenarios

Figure 1 presents a simulated power control system network topology, structured into three hierarchical layers to reflect real-world architectures. The attack scenario begins at the perimeter with an inially compromised node that exploits local vulnerabilities to obtain credentials or discover bash history for possible references to other hosts.

The Web layer, composed of Web servers from Stations 1 to 9, provides HTTP/HTTPS services for substations and the control center. Attackers can exploit SQL injection and other vulnerabilities in this layer to escalate access to DNP control nodes or the control center's PI server. The DNP control layer, consisting of DNP slave nodes across the same stations, offers SCADA services via the DNP protocol, compromising these nodes enables physical-level attacks, such as inducing voltage instability.

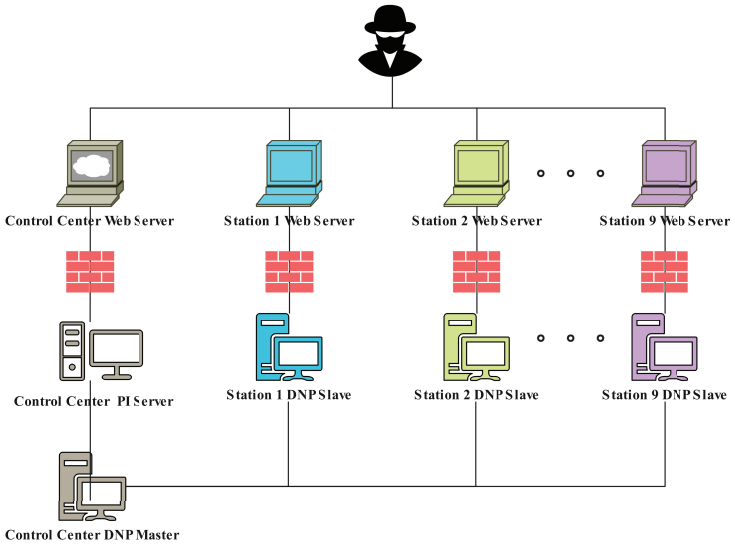


Figure 1 Structure of experimental scenario.

The final layer includes the control center's PI server, which archives critical operational data and serves as a potential target for tampering with monitoring systems or concealing malicious activities. This topology models a comprehensive attack path from initial compromise to physical disruption, illustrating the interplay between network vulnerabilities and physical consequences in cyber-physical systems.

Based on the network topology and attack process described above, we define the action space of the reinforcement learning agent to emulate various stages of an adversarial campaign across cyber and physical layers. The action space shown in Table 1 includes a diverse set of behaviors aligned with different phases of the attack chain. Remote attack actions such as scanning discovered nodes allow the agent to enumerate reachable hosts and services. Local attack actions involve listing vulnerabilities using previously gathered credentials, enabling targeted exploitation. Through authentication actions, the agent attempts to use valid credentials to escalate privileges or pivot laterally. The network scanning action enables broader reconnaissance to uncover new nodes within the environment. Restore image actions represent defensive operations such as reimaging a compromised node to a clean state.

Furthermore, the modify configuration actions empower the agent to simulate deeper system manipulations, such as altering firewall rules, changing the vulnerability profile of assets, or shutting down critical services. The sleep

action allows the agent to pause its operation in a round, mimicking stealthy behavior or decision-making delays. These actions collectively enable the reinforcement learning agent toexplore and exploit the topology in a step-wise manner, reflecting real-world adversarial behavior and supporting the modeling of both offensive and defensive strategies in cyber-physical security scenarios.

4.2 Environment Adaptation

To adapt Microsoft's IT-centric CyberBattleSim for this study, we focused on topological and asset-based representation rather than protocol-level simulation. The topology (Fig. 1) was designed to mirror a typical power system's hierarchical structure (perimeter, DNP control, control center). Nodes were labeled to represent power system assets (e.g., DNP slaves, PI server) , and vulnerabilities (e.g., SQL injection) were mapped to these assets to create plausible attack paths from the IT perimeter to the simulated OT environment.

Limitations: We acknowledge that this adaptation does not simulate power-specific protocols (e.g., DNP3, IEC 61850) or the direct physical consequences of cyber-actions. The 'compromise' of a DNP node is thus a network-level compromise (gaining administrative access) and not a verified physical manipulation (e.g., tripping a breaker). The claims in this paper are therefore focused on the intelligent penetration of the representative IT/OT network, not the full cyber-physical process.

4.3 Results and Analysis

In this section, we evaluate our proposed attacker agent, which is based on deep reinforcement learning, by conducting experiments using Microsoft's CyberBattleSim framework.

This framework offers several attacker agents that utilize either random credential lookups or deep reinforcement learning strategies. It also includes an optional 'basic defender'. This defender operates with a simple, stochastic policy: at each step, it has a 5% chance of randomly selecting a node to 'scan' (a costly no-op in this context) and a 2% chance of 're-imaging' a randomly selected node, resetting its state to uncompromised.

The relevant hyperparameters for the experimental environment are summarized in Table 2, including details of the software and hardware setup, training hyperparameters for the proposed reinforcement learning algorithm, and model-specific settings.

Table 2 Key hyperparameter settings for experiments

Category	Configuration Details
Hardware Environment	Intel Core i7-10570H @ 2.60GHz (CPU) NVIDIA Quadro T2000 (GPU) 32GB RAM
Software Stack	Operating System: Windows 10 Deep Learning Library: PyTorch
SPIND-DQN Settings	Q-network depth: 3 layers Neurons per hidden layer: 1024, 512, 128 Activation sequence: ReLU $\rightarrow$ ReLU $\rightarrow$ Softmax Optimizer: Adam Learning rate: 0.002 Discount factor (γ): 0.99 Replay buffer & mini-batch sizes: 10,000 / 512 Maximum episodes & steps: 1000 / 100 PER α / β: 0.6 / 0.4 Entropy coeff. (α): 0.1 ICM network: 2 layers, 128 neurons Target net sync interval: every 5000 steps
Reward Definitions	Attack cost (Scan): 10 Attack cost (List Vulnerabilities): 15 Attack cost (Use Credentials): 20 Defense cost (Scan): -10 Defense cost (Reimage): -15 Defense cost (Config Modification): -20 Node Compromise Reward: 100 PI Server Compromise Reward: 1000 Idle action (Sleep): 0

We then select two evaluation metrics, reward and network availability, to assess the performance of our proposed DRL agent in exploiting key asset nodes within the CyberBattleSim environment. The overall network availability is defined as the proportion of asset nodes controlled by the attacker. Let N_{att} represent the number of asset nodes under the attacker's control, and N_{all} denote the total number of asset nodes. The network availability is calculated as follows:

$$Availability = \frac{N_{all} - N_{att}}{N_{all}} \tag{14}$$

The attacker agent is trained in a network consisting of 20 nodes over 100 iterations, with each iteration comprising 1,000 interaction steps. The training (exploration-phase) reward results are presented in Figure 2, where

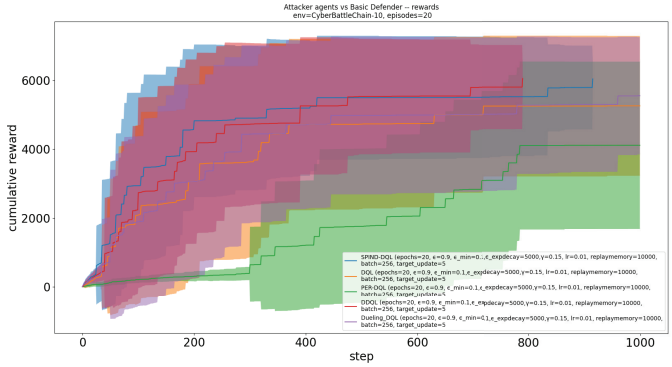


Figure 2 The training rewards between different DQN-based attacker agents and basic defender.

the shaded regions indicate the range of fluctuations and the central curves represent the expected values.

We evaluated five different variants of Deep Q-Networks (DQN): the proposed SPIND-DQL, the baseline DQN, PER-DQN based on replay of prior experience, dueling DQN (Dueling-DQN) and double DQN (DDQN). The results show that SPIND-DQL achieves higher and faster expected rewards during training compared to the other variants, followed by DDQN, Dueling-DQN, DQN, and PER-DQN.

These findings demonstrate the superior learning efficiency of SPIND-DQL in exploiting key asset nodes within the cyberspace environment. Figure 3 presents the network availability results for the five DQN agents

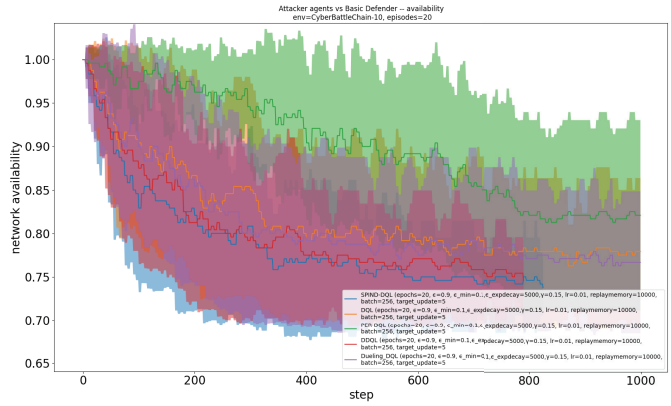


Figure 3 The network availability between different DQN-based attacker agents and basic defender during training.

during training. The results indicate that the SPIND-DQL agent achieves the lowest network availability among all baseline DQN agents, meaning it successfully compromises the highest number of asset nodes-exceeding 25% within 800 steps.

Therefore, the proposed SPIND-DQL attacker demonstrates the best network penetration performance within the simulated network environment during the learning phase.

4.4 Evaluation, Sensitivity, and Ablation Studies

To evaluate the final learned policies, we performed a separate evaluation phase. All agents were run for 500 episodes in exploitation-only mode (i.e., $\epsilon = 0$ for ϵ-greedy methods, and using the deterministic argmax of Q-values for NoisyNet/Soft Q policies). We introduce **Rainbow DQN** (Dueling + PER + NoisyNet) as a strong baseline, alongside a **Heuristic (BFS)** (breadth-first-search) attacker. We measure cumulative reward, final network availability, and **Time-to-First-Compromise (TFC)**, defined as the average number of steps to gain access to the first DNP-layer node. Results (mean $\pm$ std. dev. over 20 seeds) are in Table 3.

The results in Table 3 strongly support our hypothesis. SPIND-DQL achieves a 15% higher average reward than the strong Rainbow DQN baseline and a 104% improvement over the Heuristic (BFS) agent, demonstrating superior effectiveness. In terms of efficiency, the TFC metric is crucial; SPIND-DQL finds the first critical node 32% faster than Rainbow. Furthermore, the stability of the learned policy is enhanced, as evidenced by the lower standard deviation in reward (± 450) compared to Rainbow (± 510) and other DQN variants, suggesting a more reliable and less erratic attack strategy. The results clearly show that SPIND-DQL outperforms all baselines, including the strong Rainbow DQN agent. It achieves the highest reward, lowest network availability (most nodes compromised), and the

Table 3 Exploitation-phase performance comparison (mean $\pm$ SD over 20 seeds)

Agent	Avg. Reward	Final Availability	Avg. TFC (steps)
SPIND-DQL (Ours)	7150 ± 450	0.74 ± 0.03	98 ± 12
Rainbow DQN [17]	6200 ± 510	0.81 ± 0.04	145 ± 20
DDQN [1]	5400 ± 600	0.85 ± 0.05	190 ± 25
Dueling-DQN [3]	5100 ± 580	0.87 ± 0.05	210 ± 30
PER-DQN [9]	4800 ± 620	0.90 ± 0.04	240 ± 35
DQN [7]	5050 ± 550	0.88 ± 0.05	225 ± 28
Heuristic (BFS) [18]	3500 ± 300	0.92 ± 0.02	310 ± 40

fastest Time-to-First-Compromise. This validates our hypothesis that the addition of ICM and Soft Q-Learning provides a significant advantage in this complex exploration task.

Hyperparameter Sensitivity: The hyperparameters listed in Table 2 were determined through preliminary tuning. While a full grid search is computationally prohibitive, the ablation study (Table 4) itself acts as a sensitivity analysis on the model's core components. For instance, the significant performance drop without Soft Q-Learning or ICM indicates high sensitivity to these components. We also observed that the entropy coefficient α was critical; values too high led to overly random (ineffective) policies, while values too low nullified the exploration benefits. The value $\alpha = 0.1$ provided the best balance of exploration and exploitation for this environment.

Ablation Study: To validate the contribution of each component, we performed an ablation study, removing key components from the full SPIND-DQL agent. Table 4 shows the impact on exploitation-phase performance. Removing either ICM or Soft Q-Learning results in a significant performance drop, confirming their importance. The full SPIND-DQL model outperforms the Rainbow baseline, demonstrating the synergistic benefit of the added components.

Robustness against Stronger Defense: We also evaluated the trained SPIND-DQL agent against a 'Heuristic Defender' (HD) that actively patches the top 3 most-exploited vulnerabilities every 50 steps. Against this dynamic defense, SPIND-DQL's average reward dropped to $\sim$4500, but it still outperformed the Rainbow agent (Reward $\sim$3200). This suggests that the exploration encouraged by ICM and the stochastic policy from Soft Q-Learning allow SPIND-DQL to better adapt and find alternative attack paths when primary paths are blocked.

Table 4 Ablation study of SPIND-DQL components (exploitation phase)

Agent Variant	Avg. Reward	Final Availability	Avg. TFC (steps)
SPIND-DQL (Full)	7150	0.74	98
– Soft Q-Learning	6400	0.78	135
– ICM	6150	0.79	150
– NoisyNet	5900	0.82	175
– PER	6050	0.81	160
– Dueling	6300	0.77	140
Rainbow (Base)	6200	0.81	145

5 Discussion

The experimental results confirm that SPIND-DQL, by integrating ICM and Soft Q-Learning with a Rainbow-style foundation, provides a more efficient and effective solution for automated penetration testing compared to standard DQN variants and the strong Rainbow baseline. The superior performance, especially in Time-to-First-Compromise (Table 3) and against a dynamic defender, highlights its practical potential.

For security practitioners, this automated approach can serve as an intelligent 'red teaming' tool. By autonomously discovering complex and non-obvious attack paths, it can help 'blue teams' prioritize vulnerability patching and harden network configurations based on empirically-verified risks rather than static vulnerability scores alone. The TFC metric, for example, directly translates to the window of opportunity an organization has to detect an intrusion.

However, transitioning from this simulation to a real-world power system presents significant challenges. Real-world state spaces are vastly larger, actions carry real physical risk, and observation is far noisier. A practical implementation would require, at minimum, a high-fidelity digital twin of the target system and robust safety interlocks to prevent unintended physical consequences during training or testing.

Limitations: As noted in Section 4.2, our simulation environment is a topological and asset-based abstraction, not a protocol-level (e.g., DNP3, 61850) or physics-based simulation. Therefore, our results demonstrate the compromise of the IT/OT network, not the execution of a specific physical attack. Furthermore, our baselines, while including Rainbow and a heuristic, did not cover other RL families (e.g., PPO) or traditional model-based planning methods, which could be explored in future work.

Future research could focus on integrating this offensive agent into a broader game-theoretic framework, training it concurrently against an adaptive 'defensive agent' (e.g., an RL-based intrusion detection or response system) to co-evolve more robust offensive and defensive strategies.

6 Conclusions

In this paper, we model penetration testing in power systems as a Markov Decision Process (MDP) and address it using an enhanced deep reinforcement learning algorithm. Building on related work across various DQN schemes, we propose SPIND-DQL, which integrates NoisyNet, Dueling

Architecture, Prioritized Experience Replay (PER), Intrinsic Curiosity Module (ICM), and Soft Q-Learning to improve exploration efficiency.

These optimization mechanisms collectively guide the agent's exploration, thereby lowering the trial-and-error cost during training. We evaluate the proposed approach through simulation experiments using Microsoft's CyberBattleSim framework, adapted to represent a power system network topology. The results demonstrate that SPIND-DQL achieves better convergence and performance in large-scale and complex power system scenarios.

Our ablation studies confirmed the significant contributions of the ICM and Soft Q-Learning components, and evaluation metrics such as Time-to-First-Compromise further validated its efficiency against strong baselines like Rainbow DQN and a heuristic attacker.

Ethical Considerations

The research presented involves the development of automated offensive security tools. We acknowledge the dual-use nature of this work. This tool was developed and tested in a fully isolated, sandboxed simulation environment (CyberBattleSim) to prevent any harm to real-world systems. The objective is to create an intelligent tool for 'red teaming' and 'blue teaming' to *improve* defensive strategies by identifying vulnerabilities proactively. We advocate for the responsible use of this research for defensive and assessment purposes only.

Funding

This work was supported in part by the Northeast Branch of State Grid Corporation of China under Contract SGDB0000DKJS2400122.

References

[1] Enhanced-dueling deep q-network for trustworthy physical security of electric power substations. *Energies*, 18(12):3194, 2025.

[2] Hossein Akherati, Jalil Beyramzad, Shadi Shahmari Khiyabani, Abouzar Shariatinezhad, and Esmaeil Eskandari. Finite-time stable model free sliding mode attitude controller/observer design for uncertain space systems based on time delay estimation. *Advances in Space Research*, 2025.

[3] Youakim Badr. Enabling intrusion detection systems with dueling double deep q-learning. *Digital Transformation and Society*, 1(1), 2022.

[4] Pengcheng Chen, Shichao Liu, Bo Chen, and Li Yu. Multi-agent reinforcement learning for decentralized resilient secondary control of energy storage systems against dos attacks. *IEEE Transactions on Smart Grid*, 13(3):1739–1750, 2022.

[5] Shirin Ebadi, Zach Moolman, Eric Keller, and Tamara Lehman. Decoupling the device and identity in cellular networks with vsim. *arXiv preprint arXiv:2505.15827*, 2025.

[6] Medhat Elsayed, Melike Erol-Kantarci, Burak Kantarci, Lei Wu, and Jie Li. Low-latency communications for community resilience microgrids: A reinforcement learning approach. *IEEE Transactions on Smart Grid*, 11(2):1091–1099, 2020.

[7] G. S. R. Emil Selvan, T. Daniya, J. P. Ananth, and K. Suresh Kumar. Network intrusion detection and mitigation using hybrid optimization integrated deep q network. *Cybernetics and Systems*, 55(1):107–123, 2024.

[8] Mozaffar Etezadifar, Houshang Karimi, Amir G. Aghdam, and Jean Mahseredjian. Resilient event detection algorithm for non-intrusive load monitoring under non-ideal conditions using reinforcement learning. *IEEE Transactions on Industry Applications*, 60(2):2085–2094, 2024.

[9] D. Fährmann, N. Jorek, N. Damer, F. Kirchbuchner, and A. Kuijper. Double deep q-learning with prioritized experience replay for anomaly detection in smart environments. *IEEE Access*, 10:60836–60848, 2022.

[10] Neshat Elhami Fard and Rastko R. Selmic. Data transmission resilience to cyber-attacks on heterogeneous multi-agent deep reinforcement learning systems. In *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 758–764, 2022.

[11] Youqi Guo, Lingfeng Wang, Zhaoxi Liu, and Yitong Shen. Reinforcement-learning-based dynamic defense strategy of multistage game against dynamic load altering attack. *International Journal of Electrical Power & Energy Systems*, 131:107113, 2021.

[12] Niloofar Heidarikohol, Shuvalaxmi Dass, and Akbar Siami Namin. Evolutionary defense: Advancing moving target strategies with bio-inspired reinforcement learning to secure misconfigured software applications. *arXiv preprint arXiv:2504.09465*, 2025.

[13] Yunhan Huang, Linan Huang, and Quanyan Zhu. Reinforcement learning for feedback-enabled cyber resilience. *Annual Reviews in Control*, 53:273–295, 2022.

[14] Emadodin Jandaghi, Dalton L Stein, Adam Hoburg, Paolo Stegagno, Mingxi Zhou, and Chengzhi Yuan. Composite distributed learning and synchronization of nonlinear multi-agent systems with complete uncertain dynamics. In *2024 IEEE international conference on advanced intelligent mechatronics (AIM)*, pages 1367–1372. IEEE, 2024.

[15] Sepideh Nikookar, Sohrab Namazi Nia, Senjuti Basu Roy, Sihem Amer-Yahia, and Behrooz Omidvar-Tehrani. Model reusability in reinforcement learning. *The VLDB Journal*, 34(4):41, 2025.

[16] Abhijeet Sahu, Venkatesh Venkatraman, and Richard Macwan. Reinforcement learning environment for cyber-resilient power distribution system. *IEEE Access*, 11:127216–127228, 2023.

[17] Vikrant Sharma and Mukesh Kumar. Rainbow dqn for intrusion detection: A unified deep reinforcement learning approach across benchmark datasets. *International Journal of Applied Mathematics*, 38(5s): 647–660, 2025.

[18] S. Shen, C. Cai, Z. Li, Y. Shen, G. Wu, and S. Yu. Deep q-network-based heuristic intrusion detection against edge-based siot zero-day attacks. *Applied Soft Computing*, 150:111080, 2024.

[19] Fanrong Wei, Zhiqiang Wan, and Haibo He. Cyber-attack recovery strategy for smart grid based on deep reinforcement learning. *IEEE Transactions on Smart Grid*, 11(3):2476–2486, 2020.

[20] Lanting Zeng, Dawei Qiu, and Mingyang Sun. Resilience enhancement of multi-agent reinforcement learning-based demand response against adversarial attacks. *Applied Energy*, 324:119688, 2022.

[21] Huifeng Zhang, Dong Yue, Chunxia Dou, and Gerhard P. Hancke. Resilient optimal defensive strategy of micro-grids system via distributed deep reinforcement learning approach against fdi attack. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1): 598–608, 2024.

[22] Huifeng Zhang, Dong Yue, Chunxia Dou, Xiangpeng Xie, Kang Li, and Gerhardus P. Hancke. Resilient optimal defensive strategy of tsk fuzzy-model-based microgrids' system via a novel reinforcement learning approach. *IEEE Transactions on Neural Networks and Learning Systems*, 34(4):1921–1931, 2023.

[23] Meng Zhang, Zhuorui Wu, Jun Yan, Rongxing Lu, and Xiaohong Guan. Attack-resilient optimal pmu placement via reinforcement learning guided tree search in smart grids. *IEEE Transactions on Information Forensics and Security*, 17:1919–1929, 2022.

[24] Erfan Ziad, Zhuo Yang, Yan Lu, and Feng Ju. Knowledge constrained deep clustering for melt pool anomaly detection in laser powder bed fusion. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pages 670–675. IEEE, 2024.

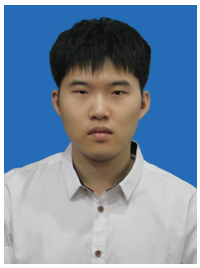
Biographies

Manpo Li received his B.S. degree in Electrical Engineering from Tsinghua University in 1994 and his M.S. degree from the Shenyang Institute of Automation Chinese Academy of Sciences in 1997. He currently serves as the Deputy Director of the Dispatch & Control Center at the Northeast Branch of State Grid Corporation of China. His research focuses on cybersecurity in electric power monitoring systems.

Ning Yang received his Bachelor's degree in Electrical Engineering from Northeast Electric Power Institute in 1995 and his Master's degree in Electrical Engineering from Northeast Electric Power University in 2008. He currently serves as the Deputy Director of the Dispatch and Control Center at the Northeast Branch of State Grid Corporation of China. His research areas include power grid dispatch automation and cybersecurity of power monitoring systems.

Xuerui Yang received his Bachelor's degree in Electrical Engineering from Wuhan University in 2014 and his Master's degree in Electrical Engineering from Wuhan University in 2016. He currently serves as the Director of the General Office at the Dispatch and Control Center of the Northeast Branch of State Grid Corporation of China. His research area includes power grid dispatch operations.

Xuezhu Jin received his Bachelor's degree in Electrical Engineering from Changchun University of Technology in 2004. He currently serves as the Director of the Dispatch Operations Department at the Dispatch and Control Center of the Northeast Branch of the State Grid Corporation of China. His research area includes power grid dispatch operations.



Long Yin received the master's degree in software engineering from JiLin University in 2016, and studying for the Ph.D. degree at Northeastern University. His research areas include cryptography and network security.

Jian Xu received his Ph.D. degree in computer application technology from Northeastern University in 2013. He is currently an Professor at Northeastern University. His research interests include cryptography and network security.