Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira,
Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria
 Published by Research Publishing, Singapore.
doi: 10.3850/ESREL2026061419_esrel26-p26597-cd

Security Management of Interdependent Critical Infrastructures by Deep Reinforcement Learning

Maria Valentina Clavijo Mesa

Energy Department, Politecnico di Milano, Italy. E-mail: mariavalentina.clavijo@polimi.it

Jinming Zhang

Energy Department, Politecnico di Milano, Italy. E-mail: jinming.zhang@polimi.it

Luca Pinciroli

Energy Department, Politecnico di Milano, Italy. E-mail: luca.pinciroli@polimi.it

Critical Infrastructures (CIs) are increasingly exposed to intentional attacks that can disrupt essential services and propagate across interdependent systems. Traditional approaches to security management often rely on static game-theoretical models or optimization schemes, which have limitations in capturing the dynamic and adaptive nature of adversarial interactions and the complexity of system-of-systems interdependences. This work introduces a novel framework for the security management of interdependent multi-state CIs using Deep Reinforcement Learning (DRL). By leveraging Adversarial Reinforcement Learning (ARL), where attacker and defender are modeled as competing agents with asymmetric knowledge and budgets, the security management policy is tested against intelligent adversaries, improving its robustness and operability. The proposed approach addresses two central challenges: *i*) the combinatorial explosion of possible attack scenarios, and *ii*) the need for effective decision-making in high-dimensional, uncertain environments. Through continuous interaction and learning, DRL provides adaptive defense strategies that go beyond static optimization, dynamically allocating resources to mitigate risks and reduce cascading effects across CIs. The framework is demonstrated through a case study involving interdependent power and water networks. Results illustrate how the proposed DRL-based framework can support operators in designing robust, adaptive, and resource-aware security management policies.

Keywords: Critical Infrastructures, Interdependent Systems, Security Management, Adversarial Reinforcement Learning.

1. Introduction

Critical Infrastructures (CIs), such as power grids, transportation systems, and water supply systems, are essential to the normal functioning of modern society [1]. With the acceleration of urbanization, these systems have become increasingly complex and interdependent, making them more vulnerable to cascading failures.

Recent incidents, including the 2021 Texas power grid collapse [2] and the 2025 Spain-Portugal power outage [3], which disrupted transport, communications and other essential services, have exposed the limitations of traditional single-point protection strategies, showing that safeguarding individual components while neglecting interdependencies is insufficient to prevent cascading failures in highly interconnected CIs [4]. Meanwhile, intentional cyber and physical attacks on CIs are becoming increasingly strategic and coordinated. Attackers exploit the interdependencies between infrastructures to amplify disruptions, while defenders must allocate limited resources to maintain overall stability [5]. Traditional game-theoretical and optimization-based defense models have been widely applied to analyze such adversarial interactions, modeling attackers and defenders as rational agents seeking equilibrium strategies [6]. Typical approaches, such as binary attack–defense and resource allocation games [7, 8], help identify critical nodes and determine optimal protection strategies using resource-based payoff formulations like contest success functions. Although these models provide valuable analytical insights, they generally assume static environments and complete

information, which limit their ability to capture the dynamic, uncertain, and adaptive nature of real-world attack–defense processes.

Consequently, integrating game theory with artificial intelligence, especially Deep Reinforcement Learning (DRL), offers a promising pathway for developing adaptive, data-driven defense strategies against intelligent and evolving threats.

In recent years, DRL has shown great potential for solving complex decision-making problems under uncertainty and has achieved notable success in applications such as autonomous driving [9], Internet of Things (IoT) systems [10], smart manufacturing [11], and the operation and maintenance of complex systems [12].

Then, this study proposes a game-theoretic DRL framework for the security management of interdependent multi-state CIs, combining game theory and artificial intelligence to model complex attack–defense dynamics. The framework employs Adversarial Reinforcement Learning (ARL) [13], in which attackers and defenders are modeled as intelligent agents operating under asymmetric information and resource constraints. To handle large-scale state and action spaces while exploiting the graph-structured representation of the CIs, the framework further integrates Graph Neural Networks (GNNs) [14].

The framework addresses two challenges: (*i*) the combinatorial explosion of potential attack scenarios and (*ii*) the need for efficient decision-making in high-dimensional, uncertain environments.

A case study on interdependent power and water networks illustrates how the proposed framework mitigates cascading failures and enhances system resilience, showing that the DRL-based approach enables defenders to design robust, adaptive, and resource-aware security policies for modern CIs.

2. Problem formulation

The security management of interdependent CIs is modeled as a two-agent adversarial stochastic game, in which an attacker and a defender repeatedly interact with a System of Systems (SoS) environment composed of interconnected CIs. At each time step t the interaction unfolds in two phases:

i. Attacker phase: the attacker selects a target node and allocates attack effort.
ii. Defender phase: the defender allocates defensive or recovery resources to a selected node.

The game is asymmetrical in both information and action scope: the attacker can act only on a single target infrastructure, while the defender observes and protects the entire SoS.

2.1 State representation

The environment state summarizes the operational conditions of the interdependent CIs and captures the potential for cascading effects arising from their interconnections. While the defender has system-wide visibility of the SoS, the information available about component conditions may be imperfect due to sensing limitations, delays, or noise. As a result, the security management problem is naturally formulated as a partially observable Markov decision process (POMDP), where decisions are made based on incomplete or uncertain observations of the underlying system state.

2.2 Attacker and defender actions

At each decision step, the attacker selects a single component to target and allocates an attack effort, subject to a limited resource budget. The defender similarly selects a component and allocates protective or recovery resources, with the objective of reducing attack impact and limiting cascading effects across the SoS. The resulting action spaces are discrete and resource-constrained, reflecting the strategic trade-off faced by both agents.

2.3 Reward function

The attacker and defender pursue opposing objectives defined at the SoS level. The defender aims to preserve functionality and resilience by limiting service disruption and cascading failures, while accounting for resource constraints. Conversely, the attacker seeks to degrade SoS performance by exploiting interdependencies to amplify disruption. These opposing objectives induce an adversarial interaction that can be represented within a zero-sum or general-sum game framework, depending on the specific reward formulation adopted.

2.4 Learning objective

The defender's objective is to learn a policy that maximizes expected cumulative rewards when interacting with an adaptive adversary. Adversarial training exposes the defender to a diverse set of attack behaviors, promoting robustness against strategic and evolving threats. The attacker, in turn, adapts its strategy in response to the defender's actions, resulting in a dynamic learning process that approximates worst-case or highly challenging attack conditions.

3. Methodology

Building on the problem formulation introduced in Section 2, this section details the modeling assumptions and learning framework used to implement the proposed adversarial security management approach. The methodology specifies how interdependent CIs are represented, how attack and defense actions drive stochastic system evolution, and how policies are learned through DRL.

3.1. *SoS representation*

CIs are modeled within a SoS structure, where multiple infrastructures operate in an interdependent manner and local degradations may propagate across systems through structural and functional couplings.

Each CI is represented as a direct graph $G^i \equiv \left(N^i, E^{(i,i)}\right)$, where N^i denotes the set of nodes corresponding to components or subsystems of the i-th CI, and $E^{(i,i)}$ captures their internal connections. Each z-th node $n_z^i \in N^i$ is characterized by a discrete multi-state variable $\bar{x}^{n_z^i}$, taking values in an ordered set of operational states $\bar{x}^{n_z^i} \in \left\{x_1^{n_z^i}, x_2^{n_z^i}, .., x_{K_{\left(n_z^i\right)}}^{n_z^i}\right\}$, where $K_{n_z^i}$ denotes the total number of admissible states of node n_z^i. The state $\left(x_1^{n_z^i}\right)$ corresponds to full operability, while $\left(x_{K_{\left(n_z^i\right)}}^{n_z^i}\right)$ represents complete failure.

The collection of node states $\bar{x}^{n_z^i}\ \forall n_z^i \in N^i$ defines the operational state of the i-th CI, denoted by X_l^i, where l indexes the CI-level state. Under the normal operational conditions, corresponding to the full operational state X_1^i, (i.e., $x_1^{n_z^i}\ \forall n_z^i \in N^i$), the demand served by the infrastructure is denoted by $D\left(X_1^i\right)$. For a generic degraded state X_l^i, the inoperability of the CI is quantified as:

$$O\left(X_l^i\right) = 1 - \frac{D(X_l^i)}{D(X_1^i)} \quad (1)$$

where $D(X_l^i)$ is the demand satisfied in state X_l^i. This metric ranges from 0 (full operability), to 1 (complete inoperability) and provides a normalized measure of CI-level performance degradation [4].

At the SoS level, overall performance is defined by aggregating the inoperability of the individual CIs. Let M denote the set of CIs composing the SoS. The SoS inoperability in a given state is expressed as

$$O_{SoS} = \mathcal{F}\left(\left\{O\left(X_l^i\right)\right\}_{i \in M}\right) \quad (2)$$

where $\mathcal{F}(\cdot)$ is an aggregation operator reflecting the desired notion of systemic performance, such as average inoperability or worst-case criterion emphasizing the most CI.

3.2. *Sequential attacker-defender dynamics*

The security management problem is formulated as a sequential attacker-defender interaction evolving over a finite discrete-time horizon $t = 0, 1, \ldots, T$.

At the beginning of each episode, the SoS structure is fixed, including CI network topologies, interdependencies, node state spaces, and the SoS performance aggregation operator. The environment is initialized to a given global operational state, typically corresponding to nominal operating conditions.

At each time step t, the environment evolves through two ordered decision stages. First, the attacker acts, producing an intermediate SoS state denoted by t^{postA}. Second, the defender acts, leading to a final state t^{postD}, which becomes the initial state for the subsequent time step. This ordered structure enforces sequentiality and allows the cumulative effects of attack and defense decisions to unfold over time.

Both agents operate under finite resource budgets R_a and R_d. At each stage, the attacker selects a node to target and allocate an effort $e_A(t) \in [0,1]$, while the defender selects a node to protect or repair and allocates an effort $e_D(t) \in [0,1]$. Effort represents the fraction of the agent's

available resources deployed at that stage, constraining each agent to localized interventions while permitting system-wide consequences through cascading effects.

The interaction between attack and defense resources is mapped into a node-level probability of failure. For a node n_k^i, let $\theta_k^i(t)$ and $\gamma_k^i(t)$ denote the normalized attacker and defender allocations, respectively. The probability that node n_k^i experiences a failure event during the attacker stage is defined as

$$P_{n_k^i}(t) = \frac{R_a \times \theta_k^i(t)}{R_a \times \theta_k^i(t) + R_d \times \gamma_k^i(t)} \quad (3)$$

Under the single-node action formulation, $\theta_k^i(t) = e_A(t)$ for the targeted node and zero otherwise, with an analogous definition for $\gamma_k^i(t)$.

To capture cumulative degradation and recovery while preserving the discrete multi-state representation introduced in Section 3.1, each node n_k^i is associated with a continuous health variable $h_{n_k^i}(t) \in [0,1]$. This variable governs stochastic transitions between discrete operational states: higher values correspond to a greater likelihood of remaining in healthy operational states, while lower values increase the probability of transitioning toward degraded or failed states. During the attacker stage, the health of the targeted node is reduced as a function of the applied effort, whereas during the defender stage the health of the selected node is increased, subject to saturation bounds. Nodes not selected at a given stage retain their previous health values.

Failure and repair are modeled as stochastic events conditioned on node health and the probability $P_{n_k^i}(t)$. Realized failures and recoveries determine the set of failed nodes, which in turn triggers cascading effects within and across interdependent CIs. Given the resulting SoS state, CI inoperability and SoS inoperability are evaluated using Eqs. (1)-(2).

The resulting attacker-defender interaction defines a stochastic, high-dimensional sequential decision problem. Due to cascading dynamics, recovery processes, and adversarial objectives, analytical solution methods are intractable. The framework therefore adopts DRL, with policies learned through repeated interactions with the SoS environment.

3.3 Reinforcement learning formulation

Proximal Policy Optimization (PPO) is adopted as the RL algorithm [15], motivated by its balance between sample efficiency, implementation simplicity, and training stability, while maintaining strong empirical performance across a wide range of continuous and discrete control problems.

At each decision step t, the attacker and defender receive observations of the environment state, denoted by $s_A(t)$ and $s_D(t)$, respectively. These state representations reflect the asymmetric information structure of the problem and are defined as:

$$s_A(t) = (Y^i, \widehat{H}^i, \Omega^i, E^{(i,i)}, O(X_l^i), R_A) \quad (4)$$

$$s_D(t) = (Y_{SoS}, H_{Sos}, \Delta_{\mathrm{SoS}}, E_{Sos}, \{O(X_l^i)\}_{i \in M}, O_{SoS}, R_D) \quad (5)$$

where Y^i represents the set of operational states of the nodes in the $i-th$ CI, $\widehat{H}^i$ represents noisy estimates of node health obtained by perturbing the true health values with noise $\sim \mathcal{N}(0,0.15)$ and clipping the resulting observations to the interval $[0,1]$, and Ω^i captures empirical attack success rates. The defender observes the full SoS state, including node operational states Y_{SoS}, health variables H_{SoS}, accumulated protection resources Δ_{SoS}, and the SoS topology E_{SoS}.

Furthermore, at each decision step t, both agents select an action consisting of a target node and an associated resources allocation. The attacker and defender actions are defined as:

$$a_A(t) = (n_k^i, \theta_k^i(t)) \quad (6)$$

$$a_D(t) = (n_k^i, \gamma_k^i(t)) \quad (7)$$

where $\theta_k^i(t)$ and $\gamma_k^i(t)$ denote the fraction of available resources allocated by the attacker and defender, respectively. Actions are constrained by finite episode-level resource budgets, enforcing realistic strategic trade-offs between early aggressive interventions and long-term sustainability.

Agent rewards are defined in terms of incremental changes in SoS inoperability, providing dense and causally grounded learning signals at each decision step. The attacker reward at time t, $r_A(t)$, is defined as

$$r_A(t) = [O_{SoS}(t^{postA}) - O_{SoS}(t-1)] - \lambda_A e_A(t) - P_{budget} - P_{idle} \quad (8)$$

while the defender reward, $r_D(t)$, is given by

$$r_D(t) = [O_{SoS}(t^{postD}) - O_{SoS}(t^{postA})] - \lambda_D e_D(t) - P_{budget} - P_{idle} \quad (9)$$

The coefficients λ_A and λ_D penalize excessive resource usage, whereas P^{budget} discourages premature budget depletion, promoting resource allocation over the full decision horizon, and P^{idle} discourages prolonged inactivity. This reward structure preserves the adversarial nature of the interaction—encouraging the attacker to maximize systemic degradation and the defender to minimize it—while providing shaped feedback that improves learning stability in large-scale interdependent environments with stochastic cascading dynamics.

3.4 Policy architecture and function approximation

Both attacker and defender policies are parameterized using neural architectures that exploit the relational structure of the underlying SoS network. Since the environment state is naturally represented as a graph, with nodes corresponding to CI components and edges encoding functional interdependencies, GNNs are adopted as feature extractors. In particular, the GraphSAGE (SAmple and aggreGatE) architecture [16] is used to compute node-level and graph-level embeddings from the raw state representations. GraphSAGE supports inductive learning through neighborhood aggregation and remains scalable to large and dynamically changing graphs, makin it suitable for interdependent CI environments.

The learned embeddings are passed to task-specific MultiLayer Perceptrons (MLPs) that parameterize the actor and critic networks within the PPO framework. This GNN–MLP architecture, improves sample efficiency by exploiting structural priors in the SoS representation and supports stable learning in high-dimensional adversarial settings. The GraphSAGE feature extractor consists of three message-passing layers with hidden dimension $d_h = 256$ and output embedding dimension $d_{out} = 128$. Mean pooling is used as the aggregation function, with a dropout rate $P_d = 0.1$ applied during training. Node embeddings are then provided to separate actor and critic MLPs, each composed of two hidden layers with 64 units.

4. Case study

The SoS considered in this study consists of a power network (PN) and a water network (WN) that are mutually dependent. The PN requires water resources for cooling processes, while the WN depends on electrical power for pumping and treatment operations. This bidirectional dependency structure enables the propagation of disruptions between CIs over time.

The PN is modeled using the IEEE57-bus test system [17], while the WN topology is derived from the IEEE39-bus system [18]. In both CIs, nodes are classified as supplier or demand nodes according to their functional role in commodity delivery. The node sets are therefore defined as $N^{PN} = N_S^{PN} \cup N_D^{PN}$ and $N^{WN} = N_S^{WN} \cup N_D^{WN}$.

Supplier nodes in both CIs are modeled using four states: fully operational, disconnected due to loss of connectivity, overloaded when capacity limits are exceeded, and completely inoperable. Demand nodes are characterized by three states: fully operational, disconnected due to loss of supply and inoperable. These states follow the multi-state representation introduced in the methodology and are used to evaluate CI-level and SoS-level performance during sequential interaction.

Under nominal operating conditions, PN comprises seven supplier nodes delivering electricity to fifty demand nodes, with a total supplied demand $D(X_1^{PN}) = 428.8\,MW$. The WN includes ten supplier nodes providing a nominal flow of $D(X_1^{WN}) = 5141.03\,\frac{m^3}{h}$ to twenty-nine demand nodes. The topologies of G^{PN} and G^{WN} are shown in Figures 1, while their physical interconnections are described in [5].

Interdependencies between the two CI are modeled through redundant functional links. Each supplier node in the WN depends on electrical power provided by two nodes in the PN, while each supplier node in the PN relies on water resources redundantly supplied by two demand nodes in the WN. This redundancy implies that a dependent node remains fully operational if at least one of its supplying nodes is in a fully functional state. Cascading failures across infrastructures are therefore evidenced only when all supplying nodes associated with a dependency link lose operability, making the SoS response sensitive to the sequence and localization of disruptions.

For the sequential attacker-defender interaction, one of the two CIs is designated as the target CI. At each discrete time step, the attacker and defender act sequentially on the target CI by selecting a single node and allocating a normalized effort $e_A, e_D \in [0,1]$, subject to resource budgets $R_a = 10$ and $R_d = 10$, respectively. The interactions unfolds over a finite horizon $T = 25$. Attack and defense efforts jointly determine node failure and recovery probabilities, while cascading effects propagate through the PN-WN interdependencies described above. SoS performance is evaluated using an average aggregation operator $\mathcal{F}(\cdot)$. The DRL agents are trained with PPO using the hyperparameters reported in Table 1. These values were selected through preliminary experiments to ensure stable learning in the stochastic adversarial setting while keeping the computational cost tractable. The same setting was used for both PN and WN scenarios to ensure consistency of the comparison.

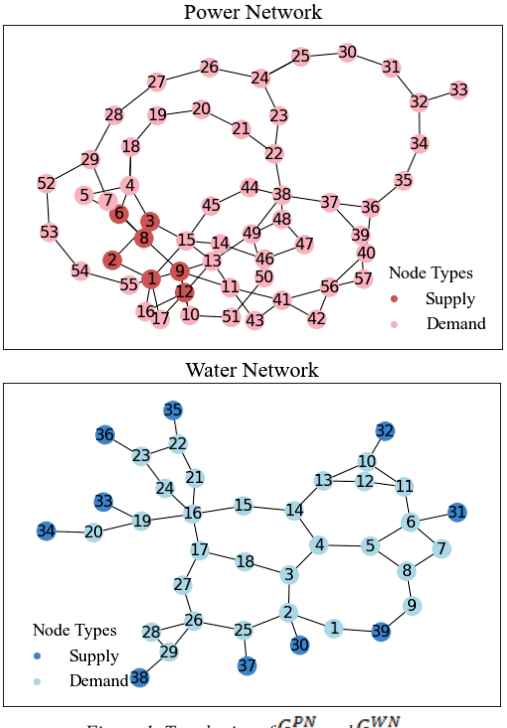

Figure 1. Topologies of G^{PN} and G^{WN}

Table 1. Parameters setting.

Parameter	Value
Learning rate	5×10^{-5}
Batch size	64
PPO epochs	10
Entropy coefficient	$0.3 \rightarrow 0.05$
Parallel agents	32
Total training episodes	42000000
P_{budget}	-5
P_{idle}	-3
λ_A, λ_D	0.5

The model is implemented in Python using StableBaselines3, PyTorch and PyTorch Geometric and the computation are carried out on the CINECA Leonardo supercomputer [19]. Performance is evaluated over 1000 testing episodes initialized with different random seeds.

5. Results

The framework is evaluated under two scenarios, corresponding to attacks targeting the PN and the WN. Figure 3 reports the episodic reward evolution for the PN-targeting scenario. The defender exhibits steady learning progress, while the attacker's rewards remain comparatively stable, indicating that defensive strategies increasingly mitigate attack effectiveness. The high variance observed in both reward trajectories reflects stochastic cascading effects and continued adversarial adaptation. Overall, the defender achieves higher final rewards, suggesting a relative advantage in protecting the PN.

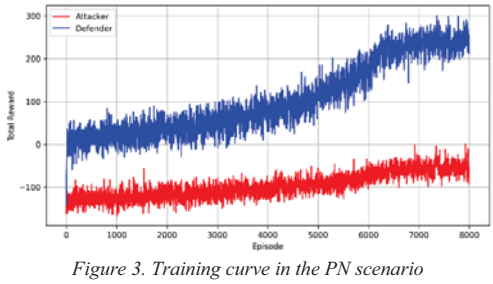

Figure 3. Training curve in the PN scenario

Figure 4 shows the reward evolution when the WN is targeted. In this case, both agents attain larger reward magnitudes, with the attacker exhibiting stronger relative performance. This behavior indicates that defending the WN is more challenging in the considered case study, not because of its smaller size alone, but because of the way service is concentrated in the WN and because of its dependence on the PN-WN

interdependency structure. Consequently, disruptions affecting a limited number of critical WN nodes can lead to larger service degradation and fewer opportunities for defensive mitigation.

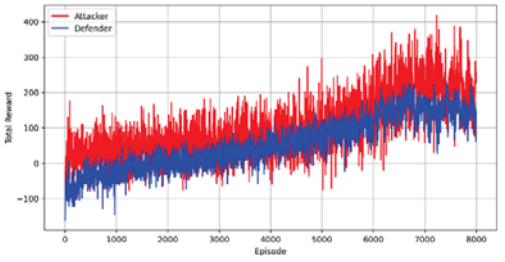

Figure 4. Training curve in the WN scenario

The learned policies are compared with random and expert baselines. These baselines are implemented as simple heuristics with fixed effort allocation: the expert attacker selects the most vulnerable operational node, whereas the expert defender prioritizes the node most frequently targeted by previous attacks. Figure 5 presents final SoS inoperability distributions for PN matchups. Random defense produces the worst results with wide distributions and catastrophic outliers, demonstrating severe vulnerability. In contrast, expert and learned defenders achieve very low median inoperability with minimal variance, confirming learned defender effectively mitigates attacks regardless of attacker sophistication and performs comparably to expert strategy.

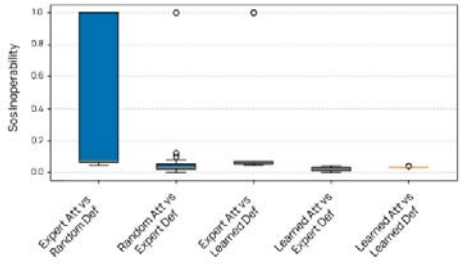

Figure 5. SoS inoperability of different attacker-defender matchups in the PN scenario

Figure 6 reveals more heterogeneous results for WN. The Expert Attacker vs. Learned Defender matchup produces notably higher inoperability than any PN case, confirming expert heuristics better exploit WN vulnerabilities. Random Attacker vs. Expert Defender shows similar median to PN scenarios but with extreme outliers, indicating higher outcome variance in particular cases. Notably, both expert and learned defenders achieve higher median inoperability against the learned attacker compared to PN equivalents, suggesting the attacker has learned to better exploit WN structure. Three critical findings emerge: *i*) learned defenders substantially outperform random strategies across both infrastructures, reducing median inoperability by 80-90% and performing similarly to ad-hoc defenses; *ii*) Despite smaller size, WN exhibits higher vulnerability and greater sensitivity to attack selection, caused by its topology; *iii*) Learned equilibria differ between infrastructures: tight, low-inoperability for PN versus moderate for WN, reflecting fundamental topological differences in defensive opportunities.

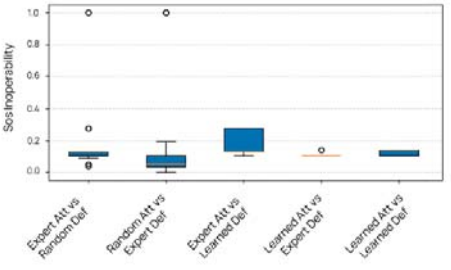

Figure 6. SoS inoperability of different attacker-defender matchups in the WN scenario

6. Conclusions

This work introduced a DRL-based framework for the security management of interdependent CIs, modeling the attack–defense problem as a sequential adversarial game embedded in a multi-state SoS environment. By combining PPO with GNN-based state representation, the framework learns adaptive strategies that account for infrastructure topology, resource constraints, and cascading effects, elements that static game-theoretic or optimization approaches cannot fully capture.

Results on the interdependent PN-WN SoS highlight three main findings. First, learned defender policies consistently outperform random baselines and reach performance comparable to expert-designed heuristics, reducing median SoS inoperability by up to 80–90% across infrastructures. Second, the comparative analysis of PN versus WN scenarios reveals structural asymmetries: the PN enables tighter control and

lower residual inoperability, while the WN exhibits higher vulnerability and larger variance due to its smaller size and topology, confirming the attacker's higher leverage on that network. Third, the attacker and defender reach distinct equilibria in the two infrastructures, demonstrating that the framework captures system-specific strategic behaviors rather than converging to generic patterns.

Acknowledgements

The authors gratefully acknowledge Dr. Matteo Broggi (Leibniz University Hannover) and Prof. Piero Baraldi (Politecnico di Milano) for their valuable contributions to the initial formulation of this research during the *1st Young Researchers Workshop on Reliability and Risk Assessment in Uncertain Environments*, where the study was first conceived.

Maria Valentina Clavijo Mesa was supported by the Italian National Recovery and Resilience Programme (PNRR). Luca Pinciroli was supported by the FAIR (Future Artificial Intelligence Research) project, funded by NextGenerationEU within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence). Jinming Zhang acknowledges financial support from the China Scholarship Council (No. 202406680014).

The authors also acknowledge the CINECA award (WIND-GNN - HP10C2186W) under the ISCRA initiative for access to the LEONARDO supercomputer.

References

[1] Rinaldi SM, Peerenboom JP, Kelly TK. (2001). Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems Magazine*;21(6):11–25. https://doi.org/10.1109/37.969131

[2] Flores, N. M., McBrien, H., Do, V., Kiang, M. V., Schlegelmilch, J., & Casey, J. A. (2022). The 2021 Texas Power Crisis: distribution, duration, and disparities. *Journal of Exposure Science & Environmental Epidemiology*, 33(1), 21 31. https://doi.org/10.1038/s41370-022-00462-5

[3] González-Pozo, R. (2025). Social profiles and response patterns during the 2025 Iberian Peninsula power outage. The case of Spain. *International Journal of Disaster Risk Reduction*, 130,105813. https://doi.org/10.1016/j.ijdrr.2025.105813

[4] Mesa, M. V. C., Di Maio, F., & Zio, E. (2025). Dynamic inoperability input-output modeling of a system of systems made of multi-state interdependent critical infrastructures. *Reliability Engineering & System Safety*, 264, 111303. https://doi.org/10.1016/j.ress.2025.111303

[5] Clavijo Mesa, M. V., Wu, Y., & Zio, E. (2025). Game theory-based defense strategies against coordinated attacks on multi-state interdependent critical infrastructures. *Proceedings of the 35th European Safety and Reliability Conference (ESREL 2025) and the 33rd Society for Risk Analysis Europe Conference (SRA-E 2025)*. https://doi.org/10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P7627-cd

[6] Wu, Y., Guo, P., Wang, Y., & Zio, E. (2024). Attack-defense game modeling framework from a vulnerability perspective to protect critical infrastructure systems. *Reliability Engineering & System Safety*, 256, 110740. https://doi.org/10.1016/j.ress.2024.110740

[7] Peng, R., Wu, D., Sun, M., & Wu, S. (2020). An attack-defense game on interdependent networks. *Journal of The Operational Research Society*, 72(10), 2331-2341. https://doi.org/10.1080/01605682.2020.1784048

[8] Lin, C., Xiao, H., Peng, R., & Xiang, Y. (2021). Optimal defense-attack strategies between M defenders and N attackers: A method based on cumulative prospect theory. *Reliability Engineering & System Safety*, 210, 107510. https://doi.org/10.1016/j.ress.2021.107510

[9] Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S., & Perez, P. (2021). Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6), 4909-4926. https://doi.org/10.1109/tits.2021.3054625

[10] Lei, L., Tan, Y., Zheng, K., Liu, S., Zhang, K., & Shen, X. (2020). Deep Reinforcement Learning for Autonomous Internet of Things: Model, Applications and Challenges. *IEEE Communications Surveys & Tutorials*, 22(3), 1722-1760. https://doi.org/10.1109/comst.2020.2988367

[11] Lu, R., Li, Y., Li, Y., Jiang, J., & Ding, Y. (2020). Multi-agent deep reinforcement learning based demand response for discrete manufacturing systems energy management. *Applied Energy*, *276*, 115473. https://doi.org/10.1016/j.apenergy.2020.115473

[12] Pinciroli, L., Baraldi, P., Compare, M., & Zio, E. (2023). Optimal operation and maintenance of energy storage systems in grid-connected microgrids by deep reinforcement learning. *Applied Energy*, *352*, 121947. https://doi.org/10.1016/j.apenergy.2023.121947

[13] Pinto, L., Davidson, J., Sukthankar, R., & Gupta, A. (2017). Robust adversarial reinforcement learning. *arXiv (Cornell University)*, 2817-2826. https://doi.org/10.48550/arxiv.1703.02702

[14] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The Graph Neural Network model. *IEEE Transactions on Neural Networks*, *20*(1), 61-80. https://doi.org/10.1109/tnn.2008.2005605

[15] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. (2017). Proximal policy optimization algorithms. https://doi.org/10.48550/arXiv.1707.06347

[16] Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. *arXiv (Cornell University)*, *30*, 1024-1034. https://doi.org/10.48550/arxiv.1706.02216

[17] MATPOWER. (2014a). Case 57 Power flow data for IEEE 57 bus test case. https://matpower50/Case57html

[18] MATPOWER. (2014b). Case 39 Power flow data for 39 bus New England system 2014. https://matpower.org/docs/ref/matpower5.0/case39.html

[19] Turisini, M., Cestari, M., & Amati, G. (2024). LEONARDO. *Journal of Large-scale Research Facilities JLSRF*, *9*(1). https://doi.org/10.17815/jlsrf-8-186