Automated Adversary-in-the-Loop Cyber-Physical Defense Planning

SANDEEP BANIK, Michigan State University, USA
THIAGARAJAN RAMACHANDRAN and ARNAB BHATTACHARYA, Pacific Northwest National Laboratory, USA
SHAUNAK D. BOPARDIKAR, Michigan State University, USA

Security of cyber-physical systems (CPS) continues to pose new challenges due to the tight integration and operational complexity of the cyber and physical components. To address these challenges, this article presents a domain-aware, optimization-based approach to determine an effective defense strategy for CPS in an automated fashion—by emulating a strategic adversary in the loop that exploits system vulnerabilities, interconnection of the CPS, and the dynamics of the physical components. Our approach builds on an adversarial decision-making model based on a Markov Decision Process (MDP) that determines the optimal cyber (discrete) and physical (continuous) attack actions over a CPS attack graph. The defense planning problem is modeled as a non-zero-sum game between the adversary and defender. We use a model-free reinforcement learning method to solve the adversary's problem as a function of the defense strategy. We then employ Bayesian optimization (BO) to find an approximate *best-response* for the defender to harden the network against the resulting adversary policy. This process is iterated multiple times to improve the strategy for both players. We demonstrate the effectiveness of our approach on a ransomware-inspired graph with a smart building system as the physical process. Numerical studies show that our method converges to a Nash equilibrium for various defender-specific costs of network hardening.

CCS Concepts: • **Computer systems organization** → **Sensors and actuators**; • **Computing methodologies** → **Reinforcement learning**; • **Security and privacy** → *Embedded systems security;*

ACM Reference format:
Sandeep Banik, Thiagarajan Ramachandran, Arnab Bhattacharya, and Shaunak D. Bopardikar. 2023. Automated Adversary-in-the-Loop Cyber-Physical Defense Planning. *ACM Trans. Cyber-Phys. Syst.* 7, 3, Article 18 (July 2023), 25 pages.
https://doi.org/10.1145/3596222

1 INTRODUCTION

A majority of the world's critical infrastructures depends on **Cyber-Physical Systems** (**CPS**) to manage essential and complex, domain-specific operational processes. Historically, CPS

This research was supported in part by a sponsored project under the Mathematics for Artificial Reasoning in Science (MARS) initiative at the Pacific Northwest National Laboratory, USA, and in part by the NSF Award CNS-2134076 under the Secure and Trustworthy Cyberspace (SaTC) program.

Authors' addresses: S. Banik and S. D. Bopardikar, Michigan State University, Electrical and Computer Engineering Department, East Lansing, Michigan 48823, USA; emails: baniksan@msu.edu, shaunak@egr.msu.edu; T. Ramachandranand and A. Bhattacharya, Pacific Northwest National Laboratory, PO Box 999, Richland, WA 99352, USA; emails: thiagarajan.ramachandran@pnnl.gov, arnab.bhattacharya@pnnl.gov.

2378-962X/2023/07-ART18 $15.00
https://doi.org/10.1145/3596222

operational risk could be attributed to human operator errors, natural disasters, and acts of physical sabotage. However, with the rapid integration of physical and cyber-security processes and increased reliance on internet-based networks, CPS is now vulnerable to sophisticated cyber attacks that can result in significant equipment damage, service disruptions, and potential loss of life. These attacks vary in severity and application; well-known examples include the StuxNet attack [37] on **supervisory control and data acquisition** (**SCADA**) systems, the German steel mill attack [38] caused by **advanced persistent threats** (**APTs**), the Ukrainian grid attack [60] via **Denial of service** (**DOS**) tactics and derailment of trams [39] using basic network access methods. In each instance, strategic threat actors used a sequence of atomic attack actions to exploit known vulnerabilities in both the cyber and physical layers of the system. MITRE ATT&CK framework is a continuously growing database of such atomic actions corresponding to specific goals on different platforms, primarily used to characterize post-compromise adversarial behavior in cybersecurity and in **Industrial Control System** (**ICS**) [5].

The core challenge in CPS security is the tight (often nebulous) integration of the cyber, physical, and computational elements. Such an integration, which can expand the CPS to arbitrary dimensions proportional to the complexity of the real-world system, necessitates a scalable framework for developing defense policies. Riding on recent successes, **Machine Learning** (**ML**)-based methods use parametric representations to create computational models to represent multi-level abstraction from data. ML has replaced hand-engineered tasks with computational models that offer high accuracy and performance. Although ML is being increasingly used in specific aspects of CPS security, such as anomaly detection [35], malware detection, intrusion detection [16], prevention of blackouts, attacks, and destruction [73], *the explicit consideration of the hybrid dynamics governing a CPS is relatively unexplored.*

This article proposes a general framework for modeling and uncovering an adversary's movements using a **hybrid attack graph** (**HAG**) and relating the security status of the cyber with the physical layer, while effectively configuring the HAG to ensure resilient operation of the CPS. The proposed framework has two components: (a) an adversary's model and policy and (b) a defender's network hardening policy. The adversary's movement is modeled using a **Markov Decision Process** (**MDP**) on the HAG, while the policy is determined using an ML method. The defender evaluates the security of the CPS using partial observations of the HAG. The security of the CPS is quantified by the adversary's movements and disruption in some *measurable services* of the physical processes. The defender uses partial observations to reason about the security of the CPS and to balance reconfiguring the HAG via network hardening and the corresponding costs. This article extends the linear parameterized ML method, introduced in our preliminary work [14], with a defender using **Bayesian optimization** (**BO**) to achieve successful network hardening. The proposed framework can be applied to a wide range of CPS and enhances the security of the system by preventing attacks and ensuring resilient operation.

1.1 Literature Review

There is a large body of work on securing CPS from an attack prevention perspective in the cyber layer, categorized broadly into (a) resilience-by-design and (b) resilience-by-reaction [18]. To position our work in literature, we organize the literature in appropriate categories as below.

Control-Theoretic Methods: The use of control theory for securing CPS has been extensively studied in the literature. For instance, [50] proposes a sampling-based worst-case design approach to overcome observation challenges and develop corresponding policies. Similarly, the work in [51] proposes a system identification and control-theoretical framework to ensure safety-critical operations in CPS. Extensive surveys of control-theoretic methods used for securing CPS are presented in [24] and [44]. Recently, [48] developed a model to explicitly links the security status between the cyber and physical layers to design an **intrusion response system** (**IRS**). *However, all of these approaches require knowledge of the system model at the cyber or at the physical level or both, making them challenging to apply in scenarios where the system model is unknown.*

Attack-Graphs: Attack graphs are commonly used to model the movement of adversaries in a cyber environment, allowing for the quantification of attack path vulnerabilities using a **common vulnerability scoring system** (**CVSS**), designed by [72]. In [78], Bayesian attack graph are used to determine the cyber attack scenarios on SCADA and **energy management system** (**EMS**) of wind farms. Petri nets, with their increased flexibility and resolution compared with attack graphs, have been a long standing tool for a range of application, including modeling cyber attacks, as demonstrated in References. [19, 46].

Adversarial Identification Frameworks: The MITRE **Adversarial Tactics, Techniques, and Common Knowledge** (**ATT&CK**) [1] framework provides a knowledge database to characterize post-compromise detection of an adversary targeting a given platform. The MITRE ATT&CK has recently been extended for **Industrial Control System** (**ICS**) [5]. Using MITRE ATT&CK framework, a **cyber kill chain** (**CKC**) has been developed and evaluated to determine the resiliency of **Distributed Energy Resouces** (**DER**) [2, 55]. Similar models characterizing the security attributes of a CPS are presented in [9] using a post-compromise database, like MITRE ATT&CK.

Attack Detection via Machine Learning: ML methods have shown significant success in enhancing the security of CPS in various applications [73]. Some of these methods include attack detection, which have been used in Reference [47] to detect false data injection attacks. To capture the temporal and spatial structure of an anomaly, convolutional and memory based encoder-decoder models are used in Reference [45]. A survey on ML based attack detectors can be found in Reference [54]. However, these methods are only used for detecting attacks and lack a defense mechanism to counteract an attack on the system.

Defense Mechanism via Reinforcement Learning: Reinforcement learning (**RL**), a subfield of ML, has been used to develop a variety of defense mechanism in CPS [40, 52]. For instance, RL has been used to develop anti-jamming [21], and anti-spoofing policies, such as the use of dynamic threshold hypothesis testing for authentic user verification in References. [41, 74]. Moreover, RL methods have also been used to indentify vulnerabilities in smart grid CPS [20, 76]. However, the described RL methods assume a fixed policy for the CPS or the adversary, and do not account for any deviation while identifying system vulnerability or developing a defense mechanism.

Game-Theoretic Methods and Network Hardening: Game-theoretic formulation in conjunction with RL have been used in Reference [53], where an adversary-defender zero-sum dynamic game is formulated to determine an optimal actions for damaging (resp. protecting) transmission lines in smart grids. Two-player games have been used to model the security policies of CPS in **Vehicular ad-hoc networks** (**VANETs**) [43] that are vulnerable to jamming attacks. Game theory has been used to model preemptive defender measures (e.g., anti-virus software or honeypot mechanisms [26]) to secure the IT systems before allowing access to potential users. Furthermore, game theory has been used in analyzing APTs [62, 64–66], where a defender can resort to **Dynamic Information Flow Tracking** (**DIFT**)—a mechanism developed to dynamically track the usage of information flows during program executions [69].

In addition to game theory, network hardening techniques have been used to secure CPS. However, the problem of network hardening has been shown to be NP-hard [67] and only resorting to heuristic solutions. Identifying system vulnerability along with attack graph-based hardening is proposed in [63], and offer efficient algorithms with provable guarantees along with the tradeoffs between hardening cost and damages inflicted on the system. In this article, we present a novel approach to securing CPS through a non-zero-sum game between an adversary and a defender.

The adversary's policy is dynamic in nature and is determined using a RL agent, while the defender's policy is static and chooses to sequentially harden the network. For a principled approach to updating the defender actions, we resort to BO methods [68].

Blackbox Optimization: Blackbox (particularly, Bayesian) optimization has its roots in early methods such as Taguchi techniques [8, 29]. Techniques for blackbox optimization can be classified into two categories, deterministic [11–13] and stochastic. Among stochastic approaches to blackbox optimization, a popular approach is based on the assumption that the unknown function can be represented as a Gaussian process [68]. Recent research has applied BO to compute approximate Nash equilibria of general sum games with continuous action spaces [3, 58] or potential games [7].

1.2 Contributions

This article presents a framework to design network hardening strategies for CPS by integrating a learning-based adversarial attack modeling approach [14] with the defense planning process. The contributions of this work are three-fold.

# 1 – A game-theoretic formulation under information asymmetry and partial system observability: This work presents a game-theoretic formulation for a CPS using an HAG model to capture probabilistic transitions of the adversary. We assume the defender does not have direct access to the adversary's actions (policy) and rewards during an attack, and operate solely on a *belief* of the cyber layer security status and some measurable attributes in the physical layer (e.g., temperature measurements in smart buildings). The interaction between the adversary and the defender is modeled as a non-zero-sum game, where the goal is to find a defense strategy solely based on appropriately modeled reward/cost functions. By formulating the adversary's and defender's problems as an MDP [14] with cyber (discrete) and physical (continuous) states, the defender's actions correspond to hardening the network, i.e., to impact the success probabilities on the cyber exploits. The solution concept that we seek is that of a *Nash equilibrium*, i.e., a pair of policies from which neither player has any incentive to deviate.

# 2 – Data-driven adversarial network hardening: Our work starts by demonstrating that the network hardening problem is equivalent to designing a slow **absorbing Markov chain** (**AMC**) that represents the progression of an attack in any CPS. Such a slow AMC design is cast into a constrained optimization problem, which is non-convex, and hence, a global solution is not guaranteed using standard optimization methods. To address this, we propose a data-driven approach to compute a best response for each player iteratively, and then find an approximate NE using the best iterated response.

Given a security policy of the defender, we adapt an **Actor Critic** (**AC**) algorithm—an RL method to solve the adversary's problem and extract the corresponding policy. To solve for the defender's best response, BO is used given the adversary's policy. Neither the AC nor the BO require explicit knowledge of the underlying dynamics of the physical or cyber processes, making them attractive for joint attack and defense planning (referred to as purple teaming) of *any complex CPS*.

# 3 – Evaluation on a smart building case-study: We evaluate our proposed approach on a smart building system, where the dynamics of the physical process were obtained from a highly accurate truncated model based on *real-world* measurements. The cyber layer of the CPS is modeled as a truncated version of a ransomware graph, created using an information flow graph [65]. The simulation results demonstrate the effectiveness of our approach in hardening the network, while also characterizing a tradeoff between hardening costs and security status of the CPS. Furthermore, we observe that the adversary and defender objectives exhibit a *diminishing marginal improvement* with increasing number of iterations of our approach, suggesting proximity to an approximate NE of the game.

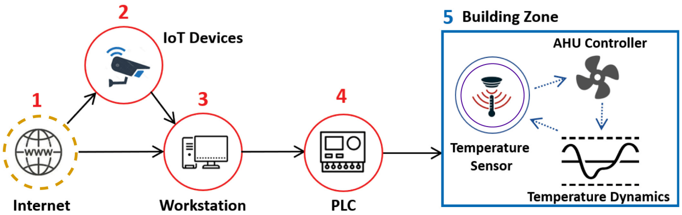


Fig. 1. A hybrid attack graph for a single-zone building with four cyber nodes (in red) and one physical node (in blue) [14]. An adversary infiltrates the leaf node (node 1) and progressively secures additional security attributes (nodes 2–4) before attacking the zone temperature controller by perturbing sensor measurements at the root node 5.

1.3 Outline

The article is organized as follows. The model formulation of the HAG describing the dynamics of cyber (discrete) and physical (continuous) components and their interactions are presented along in Section 2. Solution approaches for the adversary and defender problems are described in Section 3. Numerical experiments with the description of the cyber layer, physical layer, defense layer and results of proposed approaches in a smart building case-study in presented in Section 4. Finally, we conclude this article and outline future directions in Section 5.

2 MODEL FORMULATION

In this section, we present our adversarial threat model that characterizes the cross-layer coupling between the cyber and physical vulnerabilities in a CPS using an HAG [6, 14, 31, 33, 36, 42, 49]. An HAG is a directed acyclic graph, which represent exploitable security attributes and physical processes as nodes, and adversarial exploits (or actions) as edges. The leaf nodes represent exploitable cyber attributes as an attack entry-point (e.g., malware download into a local workstation), while root nodes denote an adversary's target set of physical-layer attributes (e.g., energy consumption, thermal comfort, traffic-lane assist). An HAG models the space of all possible attack paths available to a *strategic* adversary aiming to compromise cyber and physical components. Figure 1 illustrates a representative HAG used in Reference [14] to model cross-layer sensor-deception attacks in buildings.

The success probability of each cyber exploit along the edge of an HAG is dependent on the defense configuration. For example, the probability of detecting adversarial activity (equivalent to an unsuccessful attack action) is a function of the number of honeypots installed in the network [32]. The cyber exploits are represented using techniques from the MITRE ATT&CK framework for ICS, as done in previous works such as References. [5, 22]. The authors in Reference [22] developed an automated attack sequence generator represented as a **hidden Markov model** (**HMM**) using the same framework, with transition probabilities between tactics (nodes) and emission probabilities from tactics to a techniques. In our work, we use similar representations, i.e., the nodes can be presented as equivalent tactics and exploitable edges as techniques. Once a root node is breached, every attack action in the physical system is assumed to be successful with probability 1, and the adversary earns a corresponding reward. The adversary's objective is to progressively learn the best attack path(s) in the HAG to reach the target root node and maximize the cumulative rewards earned over a finite attack horizon. This learning problem is posed as a MDP. On the other hand, the defender's objective is to preemptively minimize any costs incurred due to the adversary compromising any physical attributes at the root node(s), such as any disruption of physical processes and the cost of network hardening. This is achieved by selecting the success probabilities on the cyber exploits appropriately. Next, we present the modeling assumptions in our problem setup.

2.1 Modeling Assumptions

Assumption 1. *The adversary has full knowledge of the HAG* topology *but has limited (no) knowledge of the success probabilities (set by the defender) at the onset of an attack.*

Assumption 2. *The defender has complete knowledge of the cyber exploits (edges in the HAG) and can allocate resources to* harden *the cyber network but not at the physical layer.*

Assumption 3. *The defender cannot observe the adversary's sequence of actions and rewards while the system is under attack.*[1]

Assumption 4. *The HAG exhibits the well-known monotonicity property, which states that an adversary never willingly relinquish attributes once obtained [49]. This simplifies our analysis by avoiding any attack paths with self-loops.*

In what follows, $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is used to denote an HAG, where $\mathcal{N}$ and $\mathcal{E}$ are the set of nodes and edges in $\mathcal{G}$, respectively. For notational clarity, we assume that $\mathcal{G}$ has only one root node; however, this assumption can be relaxed. Next, we discuss the preliminaries for the adversary's MDP model.

2.2 Preliminaries

States, Actions, and Rewards. We define Φ as the set of attack success probabilities over all edges of $\mathcal{G}$. The success probability of a cyber exploit $e \in \mathcal{E}$, conditioned on the adversary using e is denoted by $\Phi_e \in \Phi$, and is given by

$$\Phi_e \doteq \alpha_e w_e,$$

where $\alpha_e \in [\underline{\alpha}, 1]$ is chosen by the defender, $\underline{\alpha} \in (0, 1)$ is a positive lower bound on α_e, and w_e is a default (nominal) value. The defender can adjust α_e to control the success probability of e; as α_e increases, so does the success probability of e. Note that, $\underline{\alpha} > 0$ ensures that an exploit e is not made redundant by assigning a zero success probability. Let

$$\boldsymbol{\alpha} = (\alpha_e : e \in \mathcal{E})$$

be the tuple of all defender-assigned weights in $\mathcal{G}$; henceforth, we will refer to $\boldsymbol{\alpha}$ as the *defender's policy*. Note that $\boldsymbol{\alpha}$ is set prior to the onset of an attack and is constant over the attack horizon. Hardening an exploitable edge corresponds to improving defense mechanisms over the techniques (MITRE ATT&CK for ICS) used by the adversary. For instance, a cyber node such as *impair process control* (a tactic) can be hardened over exploitable techniques such as *alarm suppression, denial of service,* and others that require corresponding costs. Let $\mathcal{T} = \{1, 2, \ldots, T\}$ be a finite attack horizon. The security state of the CPS at time t is denoted by a hybrid state variable $s_t = (\gamma_t, x_t)$, where (a) $\gamma_t \in \{0, 1\}^{|\mathcal{N}|}$ is the *discrete* security state describing the current state of compromise of each node (1 means node is compromised and 0 means otherwise), and (b) $x_t \in \mathbb{R}^m$ is the *continuous* state of the physical process at the root node. The set of available attack actions in the cyber and physical layers at time t is denoted by $\mathcal{A}(s_t)$. Let Y be the total number of root nodes in $\mathcal{G}$, and $\gamma_t = \gamma_{\text{root}, i}$, for any $i \in \{1, 2, \ldots, \Upsilon\}$ represent the breach of the i^{th} physical node. Let $a(s_t) \in \mathcal{A}(s_t)$ denote an attack action taken in state s_t for a given defense policy $\boldsymbol{\alpha}$. Then, we denote the adversary's instantaneous net reward at time t by $r(s_t, a(s_t), \boldsymbol{\alpha}) \in \mathbb{R}$. Note that the net reward includes the cost incurred to launch an exploit, irrespective of whether it is successful or not.

[1] Under full information scenario between the adversary and defender, the defender's cost and adversary's net reward would be interchangeable.

CPS State Transitions. Suppose a non-root node n is compromised at time t, and there are $\mathcal{E}_{n,n'}$ exploits available to compromise a neighboring node n'. Assuming independence between different exploits, the probability that n' is compromised at time $t+1$ is given by $1 - \prod_{e \in \mathcal{E}_{n,n'}} (1 - \Phi_e)$. Such transitions represent various techniques from MITRE ATT&CK [4], and the graph nodes $\mathcal{N}$ represent equivalent tactics. For instance, an entry leaf node can be represented as an *initial access* (tactic), connected to *lateral movement* (another tactic) via cyber exploits (techniques), such as *default credentials, I/O module discovery*, and so on. Thus, the success probabilities Φ (or equivalently the defender policy $\boldsymbol{\alpha}$) influence the probabilistic evolution of the discrete state γ_t; this dependence is compactly expressed as

$$\gamma_{t+1} = g_{\text{cyb}}(\gamma_t, a(s_t), \boldsymbol{\alpha}), \tag{1}$$

where g_{cyb} is an appropriate probability transition kernel. Moreover, the physical-process dynamics at the root node is represented using a **state-space model** (**SSM**) of the form:

$$x_{t+1} = g_{\text{phy}}(x_t, u_t, w_t, a(s_t)), \tag{2}$$

$$y_t = H(x_t, w_t, a(s_t), \boldsymbol{\alpha}), \tag{3}$$

where g_{phy} is the state transition function, y_t is the measurements, H is the measurement function, u_t is a suitably designed control, and w_t is the disturbance. Note that the attack term $a(s_t)$ in Equations (1) and (2) accounts for the attack impact on the root (physical) node, only after the root node is compromised. Combining Equations (1) and (2), the security state s_t transition can be compactly denoted as

$$s_{t+1} = g(s_t, a(s_t), \boldsymbol{\alpha}), \tag{4}$$

where g comprises g_{cyb} and g_{phy}. A detailed version of the HAG and its components are described in Reference [14]. Next, we formally present the adversary's MDP model.

2.3 Adversary's Learning Problem

Let $\pi(s_t)$ denote a stationary attack policy that assigns a probability to each action in the set $\mathcal{A}(s_t)$ for a given state s_t and a defender's policy $\boldsymbol{\alpha}$. If s_t is the physical node, then π is a distribution over a finite set of actions on the physical dynamics. Let Π be the space of all feasible attack policies. Starting from an initial state $s_0 \in \mathcal{S}$ and for a given defender policy $\boldsymbol{\alpha}$, the adversary seeks a policy $\pi^* \in \Pi$ that maximizes the objective function J_{att} comprising the cumulative net reward over the attack horizon $\mathcal{T}$,

$$J_{\text{att}}(s_0, \pi, \boldsymbol{\alpha}) := \mathbb{E}\left[\sum_{t \in \mathcal{T}} r(s_t, \pi, \boldsymbol{\alpha})\right], \tag{5}$$

$$\pi^*(s_0, \boldsymbol{\alpha}) \in \arg\max_{\pi \in \Pi} J_{\text{att}}(s_0, \pi, \boldsymbol{\alpha}), \tag{6}$$

where the expectation is taken with respect to the transition kernel that defines the evolution in Equation (4).

2.4 Defender's Cyber Network Hardening Problem

The defender's objective is to minimize the combined impact of cyber attacks on the CPS and the cost of network hardening by choosing its actions $\boldsymbol{\alpha}$. Let $c^d(s, \pi(s), \boldsymbol{\alpha})$ be the cost incurred by the defender under an attack policy $\pi(.)$ for a given choice of defense action $\boldsymbol{\alpha}$. The cost may depend on the cyber states and/or physical layer attributes (discomfort or temperature fluctuations). Given a tuple of non-negative weights $\boldsymbol{\alpha}$, the network hardening cost is computed as

$$h(\boldsymbol{\alpha}) = \sum_{e \in \mathcal{E}} d_e \left(\frac{1 - \alpha_e}{\alpha_e}\right), \tag{7}$$

where d_e is a *hardening cost factor*, which will be studied in Section 4. If a cyber exploit e is not hardened, then the corresponding cost is zero, i.e., $\alpha_e = 1$.

We seek to minimize the defender's objective J_{def} over the attack horizon $\mathcal{T}$, given any initial state $s_0 \in \mathcal{S}$ and an attack policy $\pi \in \Pi$. The objective function is defined as follows:

$$J_{\text{def}}(s_0, \pi, \boldsymbol{\alpha}) := \mathbb{E}\left[\sum_{t \in \mathcal{T}} c^d(s_t, \pi(s_t), \boldsymbol{\alpha})\right] + h(\boldsymbol{\alpha}), \tag{8}$$

$$\boldsymbol{\alpha}^*(s_0, \pi) \in \operatorname*{arg\,min}_{\boldsymbol{\alpha} \in [\underline{\alpha}, 1]^{|\mathcal{E}|}} J_{\text{def}}(s_0, \pi, \boldsymbol{\alpha}), \tag{9}$$

where the expectation is taken with respect to the transition kernel in Equation (4).

Using Equations (4), (5), and (8), we define a *non-zero-sum stochastic game* being played between the defender and the adversary. The desired solution concept is that of an open-loop *Nash equilibrium* [10], , where we find a pair of attack-defense policies $\{\pi^*, \boldsymbol{\alpha}^*\}$ that are *best-responses* to each other, i.e, for which Equations (6) and (9) hold simultaneously, given any s_0. We identify sufficient conditions such as the stochastic game being zero-sum or having a specific structure (such as additive rewards for one player while the transitions are controlled by the other [34]) that guarantee the existence of Nash equilibrium policies. In particular, we adopt an iterative approach to find the best response of one player by fixing the policy of the other. We formally characterize technical conditions on the cost functions that ensures our proposed approach converges to a Nash equilibrium in a zero-sum and non zero-sum settings.

The defender's and adversary's objectives are interdependent through each other's policy, creating a paradox for solving either of the problems. For a *non-zero-sum game*, the defender's and adversary's objectives should be evaluated and optimized simultaneously. Since simultaneously solving non-zero-sum games is challenging, we propose an iterative approach to tackle the joint problem, i.e., by fixing the policy of a player first (e.g., the defender), solving for an optimal attack policy, then optimizing over the defender's policies. We numerically investigate the convergence of this approach on a CPS example in Section 4.

2.5 Computational Challenges

We elaborate on the major challenges in solving both, the adversary's and defender's problems. The adversary's problem focuses on solving the MDP (5). Traditional dynamic programming algorithms, such as value-iteration and policy-iteration [70], are infeasible for solving the optimality equation in each state due to the uncountable hybrid state space $\mathcal{S}$. Moreover, these methods assume perfect knowledge of the system and transition probabilities. However, an adversary usually has limited knowledge of the dynamics in Equation (2) and the attack success probabilities.

Similarly, the defender's objective is to solve Equation (9) using the HAG and adversary's policy. However, the defender also lacks explicit knowledge of the dynamics in the HAG and adversary's policy. This motivates the need for an *automated purple teaming* process, wherein both players solve their respective problems sequentially, until an equilibrium is reached or a specified number of iterations have been completed. In the next section, we discuss how an AC RL algorithm is used to approximately solve the adversary's problem Equation (5), as also described in our recent work [14]. For the defender's problem, we propose the use of BO to efficiently explore the defender's search space and identify a potential solution.

3 SOLUTION APPROACHES

In this section, we will begin by deriving an analytical expression for the expected time required by the adversary to reach the physical node(s), utilizing the properties of Markov chains.

The expected time to reach the physical node(s) is a function of the cyber exploits, meaning hardening the network results in a longer expected time to reach. However, we will see that the underlying problem of network hardening is non-convex, which necessitates the use of efficient search methods, such as BO, for the defender.

3.1 Markov Chain Hardening Using Expected Time

The attributes of the HAG namely, (a) directed acyclic nature of the defined attack graph, (b) the presence of leaf and root node acting as source (cyber) and sink (physical) nodes, respectively, and (c) a probabilistic distribution over the cyber exploits, make it ideal for modeling as an AMC. Using the defender's actions $\boldsymbol{\alpha}$ and the adversary's policy π, we determine the transition probabilities of the AMC states. We elaborate the components of the AMC and how the network hardening is posed as a constrained optimization problem.

Given $\mathcal{N}$ nodes and $\mathcal{E}$ edges in an HAG, we define a Markov chain M with a transition probability matrix $\widetilde{P} \in [0,1]^{|\mathcal{N}|\times|\mathcal{N}|}$. The Markov chain M defined by g_{cyb} is naturally absorbing due to the presence of sink nodes (physical nodes). Let $\widetilde{S} \subseteq \mathcal{N}$ be the set of absorbing states, and $\widetilde{T} \subseteq \mathcal{N}$ be the set of transient states, such that $\mathcal{N} = \widetilde{S} \cup \widetilde{T}$. The canonical form of the transition probability $\widetilde{P}$ is given by

$$\widetilde{P} = \begin{pmatrix} \widetilde{Q} & \mathbf{0} \\ \widetilde{R} & I \end{pmatrix}, \tag{10}$$

where $\widetilde{Q} \in \mathbb{R}^{|\widetilde{T}|\times|\widetilde{T}|}$, is the matrix corresponding to the transient states, $\widetilde{R} \in \mathbb{R}^{|\widetilde{S}|\times|\widetilde{T}|}$ is the matrix corresponding to the absorbing states, $\mathbf{0} \in \mathbb{R}^{|\widetilde{T}|\times|\widetilde{S}|}$ zero matrix, and $I \in \mathbb{R}^{|\widetilde{S}|\times|\widetilde{S}|}$ identity matrix corresponding to the absorbing states.

Let $\zeta_0 \in \Gamma$ be the initial state distribution of the Markov chain. Note that ζ_0 only contains the cyber state and represents a distribution over the transient states. For the transition probability $\widetilde{P}$, the expected absorption time [27] starting at the state ζ_0 is given by

$$\mathrm{E}[\mathrm{t}_{\mathrm{absorb}}(\widetilde{\mathrm{P}})] = J_{\mathrm{AMC}}(\widetilde{Q}, \zeta_0) := \mathbf{1}^T (I - \widetilde{Q})^{-1} \zeta_0. \tag{11}$$

The expected time governs how quickly the adversary can reach the physical node(s). The work in Reference [27] focuses on designing fast AMCs, such that the absorbing state is reached as soon as possible. However, hardening the network requires designing the matrix $\widetilde{Q}$ to deter the adversary from reaching the sink node. The optimization problem for modifying the matrix $\widetilde{Q}$ through the defender actions $\boldsymbol{\alpha}$ is given by

$$\max_{\boldsymbol{\alpha}} \; J_{\mathrm{AMC}}(\widetilde{Q}(\boldsymbol{\alpha}), \zeta_0) = \mathbf{1}^T (I - \widetilde{Q}(\boldsymbol{\alpha}))^{-1} \zeta_0 \tag{12a}$$

$$\text{s.t. } \boldsymbol{\alpha} \in [\underline{\alpha}, 1]^{|\mathcal{E}|}, \tag{12b}$$

where $1 > \underline{\alpha} > 0$ is a user-defined lower bound for the cyber exploit cost. The directed acyclic structure of HAG makes the transition matrix $\widetilde{P}$ a block lower triangular, column stochastic matrix. The elements of the fundamental matrix $J_{\mathrm{FM}} := (I - \widetilde{Q}(\boldsymbol{\alpha}))^{-1}$ are given by

$$J_{\mathrm{FM}} = \begin{bmatrix} \sum\limits_{j,\forall(j,1)\in\mathcal{E}} \alpha_{j,1} p_{j,1} & 0 & \dots & 0 \\ -\alpha_{2,1} p_{2,1} & \dots & \dots & \dots \\ \dots & \dots & \sum\limits_{j,\forall(j,i)\in\mathcal{E}} \alpha_{j,i} p_{j,i} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}^{-1}. \tag{13}$$

Equation (12a) can be re-expressed as

$$J_{\text{AMC}}(\boldsymbol{\alpha}) = \mathbf{1}^T \frac{\text{adj}(I - \widetilde{Q}(\boldsymbol{\alpha}))}{\det(I - \widetilde{Q}(\boldsymbol{\alpha}))}\zeta_0, \tag{14}$$

where det(A) and adj(A) corresponds to the determinant and adjugate of the matrix A, respectively. Since $\widetilde{Q}$ is affine in $\boldsymbol{\alpha}$, we can express Equation (14) as the ratio of two polynomials in the entries of $\boldsymbol{\alpha}$ given by

$$J_{\text{AMC}}(\boldsymbol{\alpha}) = \frac{\mathscr{P}_{|\mathcal{N}|-1}(\boldsymbol{\alpha})}{\mathscr{P}_{|\mathcal{N}|}(\boldsymbol{\alpha})}, \tag{15}$$

where $\mathscr{P}_{|\mathcal{N}|-1}(x)$ is a polynomial in x of degree at most $|\mathcal{N}|-1$. Note that the denominator has at least one more degree than the numerator, so $JAMC$ tends to infinity if and only if α_e approaches zero for all $e \in \mathcal{E}$. This leads to a solution to make all the cyber exploit weights α_e set to zero. However, setting α_e to zero in practice can disconnect different components of a CPS, rendering the problem infeasible. Moreover, the optimization problem under constraints (12b) is non-convex, and hence, a global solution is not guaranteed.

Proposition 1 (Convexity of Cost). *Suppose all entries in $\boldsymbol{\alpha}$ are identical, i.e., $\alpha_e = \alpha_a, \forall e \in \mathcal{E}$. Then,*

(1) *$J_{AMC}(\alpha_a)$ is convex in $\alpha_a, \forall \alpha_a \in [\underline{\alpha}, 1]$;*
(2) *the optimizer of* (12a) *lies on the constraint boundary.*

Proof. Under the assumption of $\alpha_e = \alpha_a, \forall e \in \mathcal{E}$, (15) changes from a ratio of polynomial in α_a to a monomial in α_a, given as

$$J_{\text{AMC}}(\alpha_a) = \frac{k_1\alpha_a^{|\mathcal{N}|-1} + k_2\alpha_a^{|\mathcal{N}|-2} + \cdots + k_{|\mathcal{N}|}}{q_1\alpha_a^{|\mathcal{N}|}}, \tag{16}$$

where $k_i, i \in 1, 2, \ldots, |\mathcal{N}|$ and q_1 are positive coefficients. Since $\frac{1}{\alpha_a^k}$ is convex for $k \geq 1$ and $\alpha_a > 0$, J_{AMC} is a sum of convex functions and therefore, is convex. The second part follows from the fact that the maximizer of a convex function always lies at the boundary of the domain. □

Observe that we are yet to include an additional or marginal cost for hardening the network in the formulation. Under the assumption $\alpha_e = \alpha_a, \forall e \in \mathcal{E}$, we add an additional cost to harden the network to obtain the hardening Markov chain objective J_{HMC} given by

$$J_{\text{HMC}}(\alpha_a) \doteq J_{\text{AMC}}(\alpha_a) + h(\alpha_a), \tag{17}$$

where $h(\alpha_a)$ is the cost of hardening. If $h(\alpha_a)$ is also convex, then J_{HMC} remains convex over α_e. Therefore, by Proposition 1, the solution will always lie at the boundary, i.e., for a given topology and costs, it will choose the cyber exploits α_e to *either completely harden or not harden at all.*

In order to model more general reward functions that also include the physical attributes, we present the use of BO for efficiently searching for non-trivial solutions. However, before we describe the approach, we briefly review the technique used to compute the optimal attack policy.

3.2 Model-Free Reinforcement Learning for Adversarial Policy Learning

AC is a model-free RL approach that learns an agent's (in this case, the adversary) policy without explicit knowledge of the probabilistic dynamics of the system (2), even for hybrid MDP state spaces. AC concurrently trains two models (called the actor and the critic) to learn a parametric form of a policy in an interactive setting with the environment (HAG). Let $\theta \in \Theta$ be a vector used to represent a parameterized value function of the form

$$V^*(s_t, \boldsymbol{\alpha}) = \max_{\pi \in \Pi} \mathbb{E}\left[r(s_t, \pi(s_t), \boldsymbol{\alpha}) + V^*(s_{t+1}, \boldsymbol{\alpha})\right], \tag{18}$$

where $V^*(s_t, \boldsymbol{\alpha})$ is the optimal value function for the state s_t and Θ has much lower dimensions as compared to $\mathcal{S}$. The AC aims to learn $\theta^* \in \Theta$ such that $\forall s \in \mathcal{S}, |V^*(s, \boldsymbol{\alpha}) - J_{\text{att}}(s, \boldsymbol{\alpha}; \theta^*)| < \epsilon$, where $J_{\text{att}}(s, \boldsymbol{\alpha}; \theta)$ is a parametrized value function, and $\epsilon > 0$ is an error tolerance. Analogous to the parameterized value function, let $\pi(s, \boldsymbol{\alpha}; \psi)$ denote a parameterized stochastic policy by $\psi \in \Psi$. At each time step, the *critic* updates the value-function parameters θ using sampled actions and successor states, while the *actor* updates the policy parameters ψ in a direction suggested by the critic. The parameters ψ and θ are updated using a stochastic gradient scheme of the form

$$\theta \leftarrow \theta + \beta^{\theta}\left(r_t + \eta J_{\text{att}}(s', \boldsymbol{\alpha}; \theta) - J_{\text{att}}(s, \boldsymbol{\alpha}; \theta)\right) \nabla_{\theta}, \tag{19a}$$

$$\psi \leftarrow \psi + \beta^{\psi}\left(r_t + \eta J_{\text{att}}(s', \boldsymbol{\alpha}; \theta) - J_{\text{att}}(s, \boldsymbol{\alpha}; \theta)\right) \nabla_{\psi} \ln \pi(s, \boldsymbol{\alpha}; \psi), \tag{19b}$$

where $\beta^{\psi} > 0$ and $\beta^{\theta} > 0$ are step-sizes for the actor and critic, respectively, that vary over the iterations, and ∇_{θ} is the gradient of J_{att} with respect to θ evaluated at $(s, \boldsymbol{\alpha}, \theta)$, η is the discount factor, and s' is the next state. The process is repeated until θ converges or a prescribed number of iterations is completed. To apply the AC algorithm in the MDP Equation (5) with discrete actions, we use an exponential softmax distribution

$$\pi(s, \boldsymbol{\alpha}; \psi) = \frac{e^{h(s,a,\psi)}}{\sum_{b \in \mathcal{A}_t(s)} e^{h(s,b,\psi)}}, \quad \forall a \in \mathcal{A}_t(s), \tag{20}$$

where e is the Euler constant. Here, the function $h(s, a, \psi)$ denotes a real-valued parametric preference defined for each state-action pair, which can be determined using tile coding or deep neural networks. The complete steps of various AC algorithms are described in Reference [70]. To implement the AC algorithm, we use an on-policy linear function approximation [70]. We use tile coding to represent multi-dimensional continuous state space, where the receptive fields of the features are grouped into partitions of state space. The convergence of **temporal difference** (**TD**) (λ) with probability 1 when the learning rates follow certain properties was demonstrated in Reference [23]. Similarly, the author in Reference [15] proved the convergence of on-line TD(0) with probability 1 while using a linear function approximator. Reference [71] introduced fast convergence algorithms for both on-line and offline policy training with linear function approximation. A comprehensive list of RL using function approximation and its convergence were reported in Reference [75]. We use the policy obtained from AC algorithm to determine an effective sequence of attacks to eventually reach the physical node(s) causing damage or disruption in service. Next, we present the solution to the defender's problem while keeping the obtained adversary policy fixed.

3.3 Bayesian Optimization for Network Hardening

Recall that our best-response based solution approach is iterative in nature: We begin with a defender policy, compute the optimal policy for the adversary (using the AC algorithm in Section 3.2), update the defender policy and repeat the process. Due to lack of knowledge of the underlying physical dynamics Equation (2) along with requiring multiple evaluations (expected value), we treat the problem as a black box and use BO [56] to solve the defender's problem. To account for the computational complexity of the defender's problem using BO, we evaluate the expectation with limited samples, to average out any measurement noise.

We initialize the defender's policies with $\alpha_e = 1, \forall e \in \mathcal{E}$, and train the adversary's policy using AC algorithm with weights θ and ψ. Once we learn an attack policy, we determine the defender's

ALGORITHM 1: Adversarial Network Hardening

```
Input: HAG, $T$ (Time horizon), $\{w_e\}, \forall e \in \mathcal{E}$ (default success probabilities), $K$ (Hardening
        iteration)
Result: Attack policy $\pi^*$, Defender's actions $\boldsymbol{\alpha}^*$
Initialize $\boldsymbol{\alpha}_1$ ($\alpha_e := 1, \forall e \in \mathcal{E}$)
for $k \leftarrow 1$ to $K$: do
    # Actor Critic for adversary
    Initialize Actor Critic weights;
    # Number of episodes of the attack
    for episode $\leftarrow 1$ to $N$: do
        Initialize $s_0 \in \mathcal{S}$
        for $t \leftarrow 1$ to $T$: do
            $a_t \sim \pi_k(s_t; \psi)$
            $s_{t+1} = g(s_t, a_t)$
            Update $\theta$ and $\psi$
        end
    end
    # Bayesian optimization for defender
    Initialize surrogate model parameters: $\mu_0(\cdot)$, $\sigma_0(\cdot)$, $k(\cdot,\cdot)$, $\rho$, $D_0 = \emptyset$
    for $b \leftarrow 1$ to $B$: do
        Obtain $\xi_b = F(\boldsymbol{\alpha}_{k,b}, \pi_k)$
        Augment data, $D_b = D_{b-1} \cup \{\boldsymbol{\alpha}_{k,b}, \xi_b\}$
        Update the GP parameters $\mu_b(\cdot)$, $\sigma_b(\cdot)$ using (22)
        Choose $\boldsymbol{\alpha}_{k,b+1} \in \arg\min_{\boldsymbol{\alpha}} q(\boldsymbol{\alpha}|D_b)$,
    end
    Choose $\boldsymbol{\alpha}_{k+1} = \arg\min q(\boldsymbol{\alpha}|D_B)$
end
Output: $\pi^* = \pi_K, \boldsymbol{\alpha}^* = \boldsymbol{\alpha}_{K+1}$
```

best response with respect to each exploit using BO. The goal of a BO process is to minimize an unknown function given by Equation (7) expressed by

$$F(\boldsymbol{\alpha}, \pi) = \mathbb{E}\left[\sum_{t \in \mathcal{T}} c^d(s_t, \pi(s_t), \boldsymbol{\alpha})\right] + h(\boldsymbol{\alpha}). \tag{21}$$

At each BO iteration b we select a tuple $\boldsymbol{\alpha}_{k,b}$ and evaluate the corresponding function value $F(\boldsymbol{\alpha}k, b, \pi_k)$, where π_k is the attack policy for the kth hardening epoch. The main idea behind BO is to maintain a surrogate function of F, such as a Gaussian process,[2] which is updated with noisy observations $\xi := [\xi_1, \ldots, \xi_B]'$ of F at the set $A_B := \{\boldsymbol{\alpha}_{k,1}, \ldots, \boldsymbol{\alpha}_{k,B}\}$ using an *acquisition function* $q(\boldsymbol{\alpha})$. The posterior over F is a Gaussian distribution with mean $\mu_B(\boldsymbol{\alpha})$ and covariance $k_B(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$ given by

$$\begin{aligned}
\mu_B(\boldsymbol{\alpha}) &= \mathbf{k}_B(\boldsymbol{\alpha})^T (K_B + \rho I)^{-1} \xi, \\
k_B(\boldsymbol{\alpha}, \boldsymbol{\alpha}') &= k(\boldsymbol{\alpha}, \boldsymbol{\alpha}') - \mathbf{k}_B(\boldsymbol{\alpha})^T (K_B + \rho I)^{-1} \mathbf{k}_B(\boldsymbol{\alpha}'), \\
\sigma_B(\boldsymbol{\alpha})^2 &= k_B(\boldsymbol{\alpha}, \boldsymbol{\alpha}'),
\end{aligned} \tag{22}$$

[2] A Gaussian process is a stochastic process, i.e., random variables indexed by space and time, such that any finite collection of those random variables has a multivariate normal distribution.

where $k : \mathrm{A} \times \mathrm{A} \to \mathbb{R}_{\geq 0}$ is the kernel function, the vector $\mathbf{k}_B(\boldsymbol{\alpha}) := [k(\boldsymbol{\alpha}_{k,1}, \boldsymbol{\alpha}) \dots k(\boldsymbol{\alpha}_{k,B}, \boldsymbol{\alpha})]^T$, K_B is the positive semi-definite kernel matrix $[k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')]_{\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in A_n}$, $\rho \geq 0$, and $\sigma_B(\boldsymbol{\alpha})$ is the standard deviation of the Gaussian measurement noise for the samples ξ. In this work, we use the *expected improvement* as the acquisition function, which is defined by Equation (23). Let $F'_B(\boldsymbol{\alpha}) := \min_{m \leq B} F(\boldsymbol{\alpha}_{k,m})$ represent the minimal observed value of $F()$ at the current iterate B, then expected improvement is defined as

$$q(\boldsymbol{\alpha}) = \mathrm{EI}_B(\boldsymbol{\alpha}) := \mathbb{E}\left[\left(F'_B(\boldsymbol{\alpha}) - F(\boldsymbol{\alpha})\right)^+ \,\middle|\, \boldsymbol{\alpha}_{k,1:B}, \xi_{1:B}\right], \tag{23}$$

where $x^+ \doteq \max\{x, 0\}$. To obtain theoretical guarantees on the suboptimality of α after B iterations, we also use the **upper confidence bound** (**UCB**) [68], which is given by

$$q_B(\boldsymbol{\alpha}) := \mu_B(\boldsymbol{\alpha}) + \sqrt{\beta_B}\sigma_B(\boldsymbol{\alpha}),$$

where, for a discrete choice of $\boldsymbol{\alpha}$, $\beta_B := 2\ln(|\boldsymbol{\alpha}|\xi_B/v)$ with an user-defined $v \in (0,1)$, and ξ_k is a sequence such that $\sum_{k=1}^{\infty} \xi_k^{-1} = 1$.

The BO algorithm in conjunction with AC is summarized in Algorithm 1, where K is the total number of BO iterations, N is the total number of episodes of the AC algorithm and T is total time duration for the system. At each BO iteration k, we return the updated cyber exploits which are used to re-train the adversary's policy with the new set of success probabilities and repeat the same process for the defined number of iterations K. Once this process terminates, we obtain the best set of defender's actions (non-negative weights) $\alpha_e^*, \forall e \in \mathcal{E}$ and the corresponding adversary policy π^*.

3.4 Analytic Properties for Zero-Sum Games

We provide analytical guarantees for our proposed approach, which involves analyzing Algorithm 1 in a zero-sum scenario by considering a finite set of pure policies for each player. For the zero-sum analysis, we swap the minimizer and maximizer. In particular, the adversary (minimizer) picks out of the set $\{\pi_1, \pi_2, \dots, \pi_m\}$ and the defender (maximizer) picks out of the set $\{\boldsymbol{\alpha_1}, \boldsymbol{\alpha_2}, \dots, \boldsymbol{\alpha_n}\}$. The cost of player policy π_i against $\boldsymbol{\alpha}_j$ equals $M_{ij}(s_0)$, where $M(s_0) \in \mathbb{R}^{m \times n}$ is the cost/payoff matrix. In what follows, we will drop the explicit dependence of M on s_0 for ease of notation.

Any Hannan consistent algorithm has properties of (i) time-average convergence to the best response policy, and (ii) $2\varepsilon-$ approximate Nash equilibrium with $\varepsilon \geq 0$ when both players update their policy using a Hannan consistent algorithm [30]. As such, our proposed approach employs a single-agent RL (adversary) to determine Nash equilibria for such repeated zero-sum games [77].

Assuming K iterations of Algorithm 1, we will leverage the following properties :

Proposition 2 ([17] Theorem 4.1 and 7.2). *Given K as the number of iterations of Algorithm 1, and let $\{P_1, \dots, P_K\}$ and $\{j_1, \dots, j_K\}$ be the possibly mixed adversary policies and pure defender policies at the corresponding iterations, respectively. Then, the adversary algorithm satisfies the following inequality*

$$\frac{1}{K}\sum_{k=1}^{K} P_k^T M e_{j_k} \leq \frac{1}{K} \min_{\bar{P} \in \Delta^m} \bar{P}^T \sum_{k=1}^{K} M e_{j_k} + \delta(m, K),$$

where e_{j_k} is the j_kth basis vector in $\mathbb{R}^n$, Δ^m is the probability simplex in m dimensions, $\delta(m,K) \geq 0$ is a Hannan consistent regret that depends on the number of adversary actions m and number of iterations K, obtained using any fixed distribution $\bar{P}$. $\delta(m,K) \geq 0$ corresponds to regret when the adversary uses a Hannan consistent [30] algorithm to update its policy every iteration.

There exist many Hannan consistent algorithms, such as exponential weighted average [17] or multiplicative weight update [28], where $\delta(m,K) = O(\sqrt{\log(m)/K})$. Before we proceed with the defender's analysis, we need to make the following assumption on the entries of M.

Assumption 5. *Each row of M is assumed to be drawn out of a Gaussian process with a given mean (typically equal to zero) and prior covariance defined by a kernel matrix $K^i(j,\ell) \geq 0$, for the ith row.*

Note that this assumption automatically implies that any linear combination of the rows is also a sample of a Gaussian process with a mean and a linear combination of the kernel matrices.

Proposition 3 ([68] Theorem 1 and Lemma 7.6). *Suppose that Assumption 5 holds. Then, against any attack distribution P_k, BO yields a pure policy e_{j_k}, such that*

$$\max_{\boldsymbol{\alpha}} P_k^T M\boldsymbol{\alpha} \leq P_k^T M e_{j_k} + \epsilon_k,$$

with probability of at least $1 - \upsilon$, where

$$\epsilon_k \in O\left(\sqrt{\frac{\gamma_B(P_k^T M)\beta_B(n)}{B}}\right).$$

Recall that $\beta_B(n) = 2\ln(n\xi_B/\upsilon)$, where the sequence ξ_k is such that $\sum_{k=1}^{\infty} \xi_k^{-1} = 1$. The information gain $\gamma_B(P_k^T M) := 0.5/(1-1/e)\max_{g_1,\ldots,g_k}\sum_{\ell=1}^{B}\log(1+\sigma^{-2}g_\ell\lambda_\ell)$, where λ's are the eigenvalues of the kernel matrix of the weighted rows $P_k^T M$, and σ is the variance of the noise in obtaining the payoff.

We are now ready to state and prove a convergence result for the zero-sum setting.

Proposition 4. *Consider the average of the attack distributions produced by Algorithm 1, $\hat{P}_K := \frac{1}{K}\sum_{k=1}^{K} P_k$. This distribution satisfies*

$$\max_{\boldsymbol{\alpha}} \hat{P}_K^T M\boldsymbol{\alpha} \leq \underbrace{\min_{P\in\Delta^m}\max_{\boldsymbol{\alpha}} P^T M\alpha}_{\textit{Value of the matrix game } M} + \frac{1}{K}\sum_{k=1}^{K}\epsilon_k + \delta(m,K),$$

with probability of at least $1 - K\upsilon$.

Proof. We start with

$$\begin{aligned}
\max_{\alpha} \hat{P}_K^T M\boldsymbol{\alpha} = \frac{1}{K}\max_{\boldsymbol{\alpha}}\sum_{k=1}^{K} P_k^T M\alpha &\leq \frac{1}{K}\sum_{k=1}^{K}\max_{\boldsymbol{\alpha}} P_k^T M\alpha \\
&\leq \frac{1}{K}\sum_{k=1}^{K}(P_k^T M e_{j_k} + \epsilon_k) \quad \text{Using Prop. 3 with prob. at least } 1 - K\upsilon, \\
&\leq \frac{1}{K}\sum_{k=1}^{K}(\bar{P}^T M e_{j_k} + \epsilon_k) + \delta(m,K) \qquad \text{using Prop. 2}, \\
&= \min_{\bar{P}\in\Delta^m} \bar{P}^T \frac{1}{K}\sum_{k=1}^{K} M e_{j_k} + \frac{1}{K}\sum_{k=1}^{K}\epsilon_k + \delta(m,K) \\
&\leq \max_{\boldsymbol{\alpha}} \bar{P}^T M\alpha + \frac{1}{K}\sum_{k=1}^{K}\epsilon_k + \delta(m,K).
\end{aligned}$$

Since this holds for any fixed distribution $\bar{P}$, one such particular choice is a saddle-point policy for the adversary. This completes the proof. □

Remark 1. Proposition 4 quantifies the proximity of the outcome of Algorithm 1 to the saddle-point value (i.e., the Nash equilibrium) of the matrix game M with high probability, under certain technical assumptions on the entries of the payoff matrix. Furthermore, the error in the outcome depends *logarithmically* on the number of rows m and columns n of the payoff matrix M. This means that one can use a large number of pure policies while incurring only a modest increase in the error bound.

3.5 Analytic Properties of the Non Zero-Sum Set-Up

In this subsection, we derive analytical properties of the non-zero-sum game under some assumptions. Consider a two-player stochastic game with a finite state space $s \in \mathcal{S}$, having finite action spaces $\pi(s)$ and $\boldsymbol{\alpha}$ for the adversary and defender, respectively, in each state s. We denote this game by

$$\Gamma = \{\mathcal{S}, \pi(s), \boldsymbol{\alpha}, \hat{r}, p\}, \tag{24}$$

where $\hat{r} := \{\hat{r}_1, \hat{r}_2\}$ is a vector-valued function for the defender and adversary, respectively, in the domain

$$\mathcal{Z} = \{(s, \pi(s), \boldsymbol{\alpha}); s \in \mathcal{S}, \pi(s) \in \Pi, \boldsymbol{\alpha} \in [\underline{\alpha}, 1]^{|\mathcal{E}|}\}.$$

In particular, $\hat{r} := \{\hat{r}_1 := c^d(s, \pi, \boldsymbol{\alpha}), \hat{r}_2 := r(s, \pi, \boldsymbol{\alpha})\}$ for the described problem Equations (9) and (6), respectively. Lastly, the state transition probability is given by

$$\mathbf{p} = \{p(z|s, \pi(s), \boldsymbol{\alpha}); z \in \mathcal{S}, (s, \pi(s), \boldsymbol{\alpha}) \in \mathcal{Z}\},$$

where $p(z|s, \pi(s), \boldsymbol{\alpha})$ denotes the probability that the state moves from state s to z when the actions $\pi(s)$ and $\boldsymbol{\alpha}$ are taken in the state s. The state transition probabilities satisfy the following properties,

$$p(z|s, \pi(s), \boldsymbol{\alpha}) \geq 0, \quad \text{and} \quad \sum_{z \in \mathcal{S}} p(z|s, \pi(s), \boldsymbol{\alpha}) = 1.$$

Definition 1 (Additive reward (AR) and additive transition (AT) game (ARAT game) [61]). The game Γ Equation (24) possesses an additive rewards property, if for all $(s, \pi(s), \boldsymbol{\alpha}) \in \mathcal{Z}$,

$$\begin{aligned} c^d(s, \pi, \boldsymbol{\alpha}) &= c_1^d(s, \boldsymbol{\alpha}) + c_2^d(s, \pi), \\ r(s, \pi, \boldsymbol{\alpha}) &= r_1(s, \boldsymbol{\alpha}) + r_2(s, \pi), \end{aligned}$$

for appropriate functions c_1^d, c_2^d, r_1 and r_2 on the domain. The game Γ Equation (24) simplifies to a *controlling game* if the states can be partitioned into two sets $\mathcal{S}_1$ and $\mathcal{S}_2$ such that

$$\begin{aligned} \forall s \in \mathcal{S}_1, &\quad p(z|s, \pi(s), \boldsymbol{\alpha}) = p_1(z|s, \boldsymbol{\alpha}) \\ \forall s \in \mathcal{S}_2, &\quad p(z|s, \pi(s), \boldsymbol{\alpha}) = p_2(z|s, \pi(s)). \end{aligned}$$

The partitioning of states enables the game Γ Equation (24) to possess additive transitions for all $(s, \pi(s), \boldsymbol{\alpha}) \in \mathcal{Z}$ of the form

$$p(z|s, \pi(s), \boldsymbol{\alpha}) = p_1(z|s, \boldsymbol{\alpha}) + p_2(z|s, \pi(s))$$

Assumption 6 ("Switching control graphs"). *The graph $\mathcal{G}$ satisfies the following properties:*

1. *There are no self loops,*
2. *the defense policy $\boldsymbol{\alpha}$ is such that for every cyber node with a single outgoing edge e, $\alpha_e \neq 1$,*
3. *for every other edge, $\alpha_e = 1$, and*
4. *the game is played over an infinite horizon in a discounted setting*

A line graph represents one such example. Figure 2 shows a non-trivial example of a switching control graph. Then, the following is a property of the game described in Sections 4.2 and 4.3.

Proposition 5 (ARAT Game with Switching Control Graphs). *Under Assumption 6, the stochastic game Γ defined by Equation (24) is an ARAT game.*

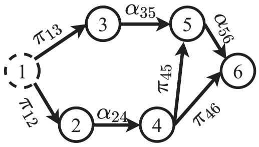


Fig. 2. Switching control graph with nodes 1, 4, and 6 representing adversary control, and nodes 2, 3, and 5 representing defender control.

Proof. We will verify that the cyber rewards Equation (25) and physical rewards Equation (27) satisfy the AR property, and the state transitions satisfy the AT properties. Under Assumption 6, the expected cyber rewards are partitioned as

$$r(s_t, \pi, \boldsymbol{\alpha}) = \begin{cases} r_1(s_t, \boldsymbol{\alpha}) := \alpha_e w_e - c, & s_t \in \mathcal{S}_1, \\ r_2(s_t, \pi) := \pi_e(s_t) w_e - c(\pi_e(s_t)), & s_t \in \mathcal{S}_2, \end{cases}$$

where c corresponds to the cyber cost for all the states belonging to the set $\mathcal{S}_1$, i.e., states under defender's control.

Note that when the adversary reaches the physical state $s_t = \{y_{\text{root}}, x_t\}$ the defender's action has no impact on the reward obtained by the adversary. The state transitions under Assumption 6 are of the form

$$p_1(z|s_t, \boldsymbol{\alpha}) = \alpha_e w_e, \forall s_t \in \mathcal{S}_1,$$
$$p_2(z|s_t, \boldsymbol{\alpha}) = \pi_e(s_t) w_e, \forall s_t \in \mathcal{S}_2.$$

Therefore, we satisfy both ARAT property for the stochastic game Γ Equation (24). □

Using Theorem 3.1 from Reference [61], we conclude that the ARAT game Γ Equation (24) admits a Nash equilibrium in stationary strategies which uses at most two pure actions for each player in each state. This result will allow us to significantly prune down the adversary edges of a large graph that satisfies Assumption 6.

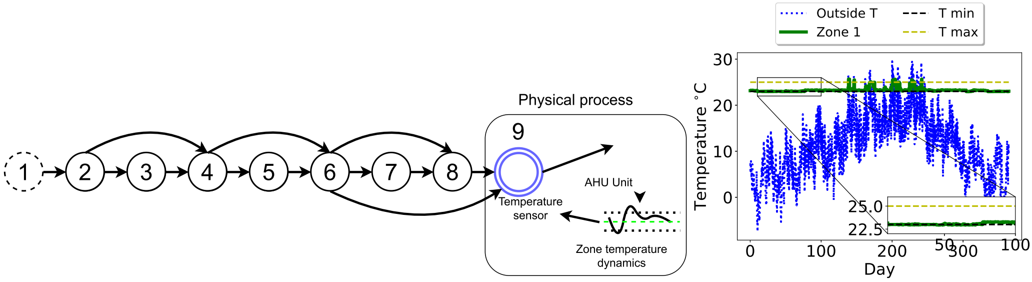
4 NUMERICAL EXPERIMENTS

We now demonstrate the effectiveness of our proposed network hardening algorithm on a smart building case-study with a cyber layer inspired by a ransomware attack graph and the physical layer obtained from a truncated model identified using real-world experiments.

4.1 Case-Study: Sensor Deception Attacks on Building

In this use case, the adversary aims to maximize the occupant discomfort of a single zone in the given building over a defined time horizon, while the defender seeks to minimize a combination of the discomfort and the hardening cost. The building's **air-handling unit** (**AHU**) performs standard operations by reconditioning ambient air and return air to a specific supply-air temperature and then supplying it to various building zones using a supply fan. The adversary aims to manipulate temperature measurements from various zone-level sensors to deceive the AHU control system and send poorly conditioned air into various zones, causing comfort-bound violations over time. However, to gain access to the temperature sensors at various zones, the adversary has to penetrate the sensor unit via a set of cyber exploits present on different components of a **Building Automation System** (**BAS**), such as IoT devices (e.g., IP cameras and smart thermostats), building-management workstations, and **programmable logic controllers** (**PLC**).

For the cyber layer, we use a pruned version of a ransomware attack graph [65] created using information flow. The original graph represents multiple stages of an attack progression: (a) a



(a) (b)

Fig. 3. (a) An HAG inspired from a ransomware attack graph [65]. The source node 1 is represented by the dashed circle and the physical node (sink node) 9 is represented by concentric circles. (b) Trajectories of Zone 1 temperature (Zone 1) along with the outside air temperature (Outside T) over a year with upper (T max) and lower temperature (T min) comfort bounds.

privilege escalation stage, (b) lateral movement over the cyber nodes, and (c) reaching the goal node. We use a HAG to represent these specific stages, as shown in Figure 3(a). Similar attack graphs for BAS were used in [25], where the attack paths involved executing a subset of tactics defined in popular attack frameworks, such as MITRE's ATT&CK [1].

The reward functions used for an adversary in the cyber and physical layer of a CPS is usually system-specific and depends on the system's overall security objective and specifications. For instance, the cyber reward at a certain node in a HAG can be set equal to the loss a defender or system administrator would incur in case an adversary were to successfully access the corresponding node. For this case study, we set the cyber reward to a positive value that incentivizes a resource- and/or time-constrained adversary to reach the physical node as quickly as possible. However, other cyber-layer reward specifications can be easily integrated in our framework. On the other hand, reward in the physical layer is generally associated with a metric that corresponds to loss in physical-system performance due to the adversary's actions. Examples include power, energy, efficiency or deviation of performance beyond a specified bound. It is also important to note that probability of transitions between different nodes in a HAG is usually determined from related attack-incident reports in the literature (see Reference [22] for more details). However, we use synthetic transition-probability values in the ransomware attack graph for demonstrative purposes only. Next, we elaborate the cyber and physical layer components of the proposed HAG using notation described in Section 2.

4.2 Cyber Layer

The HAG consists of eight cyber vertices with the associated cyber exploits also known as tactics from MITRE ATT&CK framework. The physical node is represented via concentric blue circles (node 9 in Figure 3(a)). Each vertex (tactic) and its corresponding edge (technique) are shown in Table 1. A user can generate such attack graphs and models using the framework in Reference [22]. The success probability of any of the cyber exploit is independently sampled from a uniform distribution, $\mathcal{U} \sim [0.5, 1)$. For an attack action $a_t \sim \pi(s_t, \boldsymbol{\alpha}; \psi)$ on the cyber layer, the adversary incurs a cost $c(a_t)$ of 0.1 and a nominal reward of 1 if an exploit is successful, while the reward for doing nothing is assigned a value of 0. The reward from the cyber layer to the adversary is given by

$$r(s_t, \pi, \boldsymbol{\alpha}) = \begin{cases} 1 - c(\pi_e(s_t)), & \text{with probability} \quad \alpha_e w_e \pi_e(s_t), \\ -c(\pi_e(s_t)), & \text{with probability} \quad 1 - \alpha_e w_e \pi_e(s_t), \end{cases} \tag{25}$$

Table 1. Cyber Exploits and their Corresponding Probability of Success

Node	Tactic	Edge	Transition Probability	Node	Tactic	Edge	Transition Probability	Node	Tactic	Edge	Transition Probability
1	Initial access	(1,2)	0.82 (Internet Accessible Device)	4	Evasion	(4,5)	0.56 (Utilize/Change operating module)	6	Lateral movement	(6,8)	0.87 (Remote File Copy)
2	Execution	(2,3)	0.63 (Execution through API)			(4,6)	0.94 (Rootkit)			(6,9)	0.78 (Program Organization Units)
		(2,4)	0.88 (Man in the middle)	5	Discovery	(5,6)	0.97 (Control Device Identification)	7	Inhibit response function	(7,8)	0.87 (Block serial COM)
3	Persistence	(3,4)	0.89 (Module Firmware)	6	Lateral movement	(6,7)	0.59 (External Remote Services)	8	Impair process control	(8,9)	0.50 (Change Program State)

Table 2. Description of the Variables in the Building Model

Variable	Description	Unit
x_t	Building envelope states	°C
y_t	Zone temperature measurements	°C
u_t	Amount of heating or cooling (control inputs)	$°\mathrm{C\ kg\ s^{-1}}$
w_t	Ambient temperature (disturbance)	°C

where $\pi_e(s_t)$ denotes the adversary's probability of choosing exploit e while in the state s_t. Then, the expected reward until the root (physical) node is not compromised is given by

$$\mathbb{E}[r(s_t, \pi, \boldsymbol{\alpha})] = \alpha_e w_e \pi_e(s_t) - c(\pi_e(s_t)),$$

subject to the dynamics in Equation (4). Note that each exploit has a positive expected net reward, which incentivizes the adversary to reach the root node as quickly as possible.

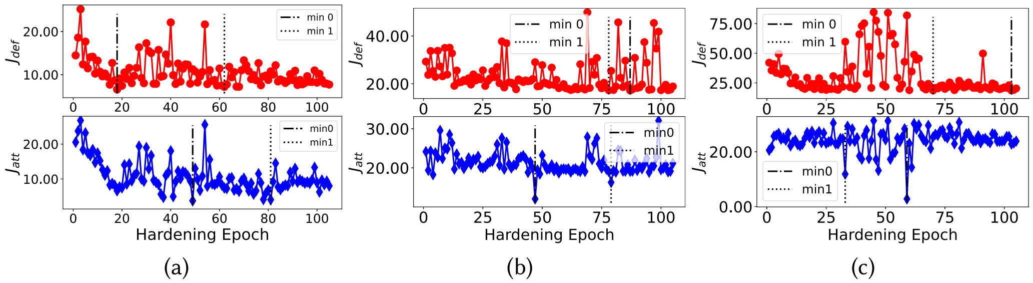
4.3 Physical Layer

We consider a multi-zone residential building with a single floor as our representative building, which is based on the setup described in Reference [57]. The building has six conditioning zones and a central AHU that sends thermally conditioned air to each zone using a supply-air fan. The AHU unit uses an absorption chiller for conventional cooling and a backup boiler for emergency heating during very low ambient temperatures. Conventional heating is provided by **Variable Air Volume** (**VAV**) terminal units with reheat coils that regulate the temperature and flow-rate of the air entering each zone.

To accurately model the building dynamics, a *linearized*, time-invariant, discrete-time, reduced-order SSM can be used, as discussed in Reference [57]. We use the *RenoLight* SSM as part of the **Python Systems Library** (**PSL**) [59] to simulate the dynamics of our representative building. The RenoLight model consists of 250 states (building envelope variables), 6 control inputs (amount of heating or cooling for each zone) and 6 observations (zone temperatures). The sampling frequency of the model is set to 15 minutes. Notation and description of the different components of the SSM are reported in Table 2. We use a rule-based controller to provide occupant thermal comfort by maintaining zone temperature in each zone within specified comfort bounds. Specifically, the amount of heating or cooling at time t in zone i was set according to

$$u_t^i = \begin{cases} -u_{\max} \min\left\{\frac{y_t^i - y_{\max} + \delta}{\tilde{\epsilon}}, 1\right\}, & \text{if } y_t^i > y_{\max} - o, \\ -u_{\max} \min\left\{\frac{-y_t^i + y_{\min} + \delta}{\tilde{\epsilon}}, 1\right\}, & \text{if } y_t^i \le y_{\min} + o, \\ 0, & \text{otherwise,} \end{cases}$$

where $y_{\min}$ and $y_{\max}$ are the prescribed lower and upper comfort bounds, o is hysteresis parameter, $\tilde{\epsilon}$ is proportional gain and $u_{\max}$ is the maximum heating or cooling capacity of the controller. For our experiments, we set $y_{\min} = 23\,°\mathrm{C}$ and $y_{\max} = 25\,°\mathrm{C}$, respectively. Figure 3(b) shows the nominal



(a) (b) (c)

Fig. 4. (a) Defender's (J_{def}) and Attacker's (J_{att}) objective with a hardening cost factor $d_e := 0.1$, where $\min(i)$ is defined as the i^{th} argument minimum of $J_{\text{def/att}}$. (b) Defender's (J_{def}) and Attacker's (J_{att}) objective with a hardening cost factor of $d_e := 0.5$. (c) Defender's (J_{def}) and Attacker's (J_{att}) objective with a hardening cost factor $d_e := 1$.

annual performance of the rule-based controller (under no attacks), which clearly shows that the zone temperatures stay within the comfort bounds with high probability.

On acquiring access to a zone temperature sensor, the adversary can perturb the sensor measurements to cause occupant discomfort in that zone. With a slight abuse of notation, let a_t be the adversarial temperature perturbation at time t. For demonstrative purposes, only the temperatures in zone 1 are allowed to be perturbed; henceforth, we drop the zone superscripts. The perturbed zone temperature measurement at time t changes to $y_t = x_t + a_t$. The adversary's reward for executing the action a_t when the physical state is $s_t = \{\gamma_t, x_t\} = \{\gamma_{root}, x_t\}$, denoted by $r(s_t, a_t)$, equals

$$r(s_t, a_t) = (y_{\min} - y_t)^+ + (y_t - y_{\max})^+ - ca_t^2. \tag{26}$$

For $u \in \mathbb{R}$, the first (resp. second) term is the thermal discomfort caused by temperature deviation from the lower (resp. upper) comfort bound. The cost for executing an action a_t is scaled by a proportional term c. Since a_t takes values in a discrete set, the expected reward is given by

$$\mathbb{E}[r(s_t, \pi_t)] = \sum_{a_t \in \mathcal{A}(s_t)} \pi_{a_t}(s_t)\Big((y_{\min} - y_t)^+ + (y_t - y_{\max})^+ - ca_t^2 \Big).$$

Note that based on the action and the state, the adversary will either observe the cyber or the physical reward.

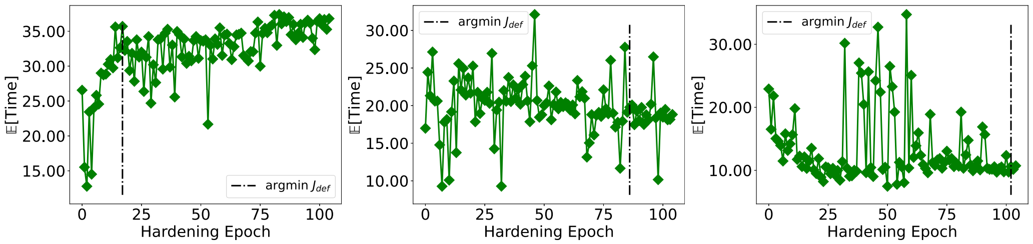
Once the root node is compromised, the defender can only measure the discomfort caused by the adversary's perturbation in any zone. The expected return incurred under a set of defenses and adversary policy in state s_t equals

$$c^d(s_t, \pi) = -\mathbb{E}\left[r(s_t, \pi)\right]. \tag{27}$$

Since the defender's actions are purely on the cyber layer, once an adversary reaches a root node, the return c^d is invariant of the defender policy $\boldsymbol{\alpha}$.

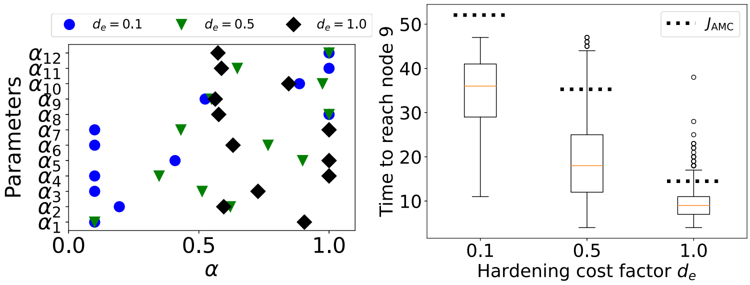
Network Hardening

We numerically demonstrate the outcome of Algorithm 1 with the following parameters, (a) time horizon $T = 48$, (b) hardening iteration $K = 100$, (c) AC episodes $N = 30,000$ and (d) lower bound for hardening $\underline{\alpha} = 0.1$. Figure 4 illustrates the defender's and adversary's objectives at the end of the hardening iteration for different values of the hardening cost factor $d_e = 0.1, 0.5$, and 1. As shown in Figures 4(a), 4(b), and 4(c), increasing values of d_e lead to higher objectives for both the adversary and defender. The adversary's and defender's objectives show diminishing marginal improvement with an increasing number of iterations of the approach, suggesting proximity to an approximate



(a) (b) (c)

Fig. 5. (a) Average time steps required to reach the physical node for the adversary for the hardening factor of $d_e := 0.1$. (b) Average time steps to reach the physical node with $d_e := 0.5$ (c) Average time steps to reach the physical node with $d_e := 1.0$.



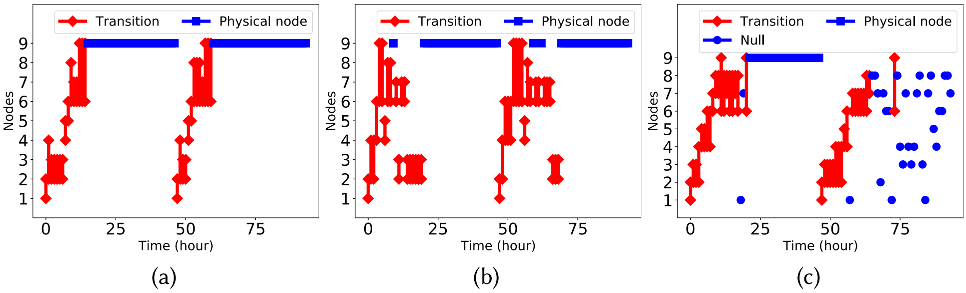
(a) (b)

Fig. 6. (a) Cyber exploits weights obtained from the result of Algorithm 1 with a cyber cost factor of $d_e = 0.1, 0.5$ and 1.0. (b) Time to reach the physical node 9 for varying hardening cost factor and compared with the expected time to reach (J_{AMC}) obtained from (11).

NE of the game. But since this is a non-zero-sum game, characterizing additional properties such as the price of anarchy and convergence to an NE will require additional assumptions on the structure of the players' objectives, and is a topic of future investigation.

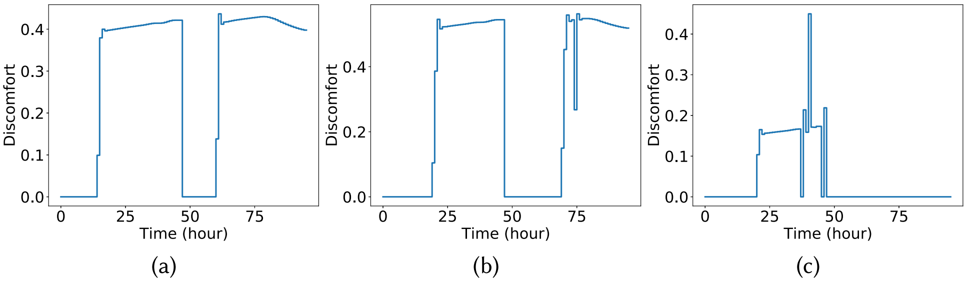
We quantify the effectiveness of Algorithm 1, by measuring the average time taken by the adversary to reach the physical node during the BO process for different values of d_e, as shown in Figures 5(a), 5(b) and 5(c). We observe that as d_e increases, the average time taken to reach the physical node decreases. Furthermore, we compared the distribution of the time required to reach the physical node for the corresponding values of d_e against the expected absorption time in Equation (11) as shown in Figure 6(b). We observe that the expected absorption time J_{AMC} is greater than the median value of the empirically determined time. This result justifies the use of the proposed approach over standard optimization methods for optimizing J_{AMC}

Next, we visualize the defender policy $\boldsymbol{\alpha}$ for the three values of d_e shown in Figure 6(a). We observe that a majority of the weights are hardened for smaller values of d_e, indicating the effectiveness of our approach in balancing between the cost of hardening and the cost of securing the CPS. We demonstrate a sample node trajectory for the corresponding values of d_e shown in Figures 7(a), 7(b) and 7(c). As expected, the adversary takes significantly longer to reach the physical node with $d_e = 0.1$ as compared to $d_e = 1.0$. Finally, as the defender can only observe the discomfort in HAG, we evaluated the same for the prior defined values of d_e using the obtained policies of $\{\pi^*, \boldsymbol{\alpha}^*\}$ shown in Figure 8(a), 8(b) and 8(c). The results show a decrease in discomfort for the lowest value of $d_e := 0.1$.



(a) (b) (c)

Fig. 7. Sample node trajectory obtained from an attack policy with a hardening cost factor of (a) $d_e = 1.0$, (b)$d_e = 0.5$, and (c) $d_e = 0.1$, where with null action corresponding to no action taken by the adversary.



(a) (b) (c)

Fig. 8. Discomfort corresponding under the optimal policy $\{\pi^*, \boldsymbol{\alpha}^*\}$ for the hardening cost factor (a) $d_e := 1.0$, (b) $d_e := 0.5$, and (c) $d_e := 0.1$.

Our approach to adversarial network hardening provides a principled defense planning solution in the presence of an adversary. Despite the defender's limited knowledge of the adversary's movements and only being able to measure physical attributes, our approach prevents the adversary from gaining privileges in the HAG. Our framework optimizes network hardening and adversary cost simultaneously, resulting in robust policies for both players, leading to an approximate best-response pair for the non-zero-sum game. This approach offers a promising defense mechanism against adversarial attacks.

5 CONCLUSION AND FUTURE DIRECTIONS

This article developed a domain-aware framework for automated adversarial defense planning, accounting for cross-layer interaction between the cyber and physical components of a CPS. Our approach leveraged an MDP with a hybrid state representing the cyber (discrete) and physical (continuous) state of the system to capture the adversary's progression over the HAG. We formulated the automated defense planning as a non-zero-sum game between an adversary and a defender. We used AC, a RL method, and BO to iteratively solve the adversary's and defender's problem, respectively. Finally, we demonstrated the effectiveness of our proposed framework on a ransomware inspired graph in conjunction with smart building dynamics. The obtained results show a hardened network for varying hardening costs along with diminishing marginal improvement for both players.

Future work will focus on studying the convergence properties of our proposed approach. Additionally, integrating an **Intrusion Detection System** (**IDS**) and an IRS on the cyber layer would enable a more informed and active defender. We also plan to extend the defender's policy from a static network hardening approach to an active network reconfiguration with one or multiple adversaries in the HAG. Exploring zero-day exploits and preemptive defense mechanisms within the framework is another area of interest. Finally, we will investigate the strategic use of backup systems and their interaction within the CPS. These backup systems could represent hidden parts of the HAG, and the defender may choose to activate them to improve the current system's performance.

REFERENCES

[1] 2021. MITRE ATT&CK. Retrieved from https://attack.mitre.org/.

[2] BoHyun Ahn, Taesic Kim, Jinchun Choi, Sung-won Park, Kuchan Park, and Dongjun Won. 2021. A cyber kill chain model for distributed energy resources (DER) aggregation systems. In *2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. 1–5. DOI:https://doi.org/10.1109/ISGT49243.2021.9372209

[3] Abdullah Al-Dujaili, Erik Hemberg, and Una-May O'Reilly. 2018. Approximating nash equilibria for black-box games: A bayesian optimization approach. *International Workshop on Optimization in Multiagent Systems*, AAMAS. https://web.ecs.syr.edu/~ffiorett/cfp/OPTMAS18/papers/paper_14.pdf.

[4] Rawan Al-Shaer, Jonathan M. Spring, and Eliana Christou. 2020. Learning the associations of MITRE ATT & CK adversarial techniques. In *2020 IEEE Conference on Communications and Network Security (CNS)*. 1–9. DOI: https://doi.org/10.1109/CNS48642.2020.9162207

[5] Otis Alexander, Misha Belisle, and Jacob Steele. 2020. MITRE ATT&CK® for industrial control systems: Design and philosophy. *The MITRE Corporation: Bedford, MA, USA* (2020), 29. https://attack.mitre.org/docs/ATTACK_for_ICS_Philosophy_March_2020.pdf.

[6] Paul Ammann, Duminda Wijesekera, and Saket Kaushik. 2002. Scalable, graph-based network vulnerability analysis. *(CCS'02)*. Association for Computing Machinery, New York, NY, 217–224. https://doi.org/10.1145/586110.586140

[7] Anup Aprem and Stephen Roberts. 2019. A bayesian optimization approach to compute nash equilibrium of potential games using bandit feedback. *Computer Journal* 64, 12 (Dec. 2019), 1801–1813. DOI: https://doi.org/10.1093/comjnl/bxz146

[8] Martin Arvidsson and Ida Gremyr. 2008. Principles of robust design methodology. *Quality and Reliability Engineering International* 24, 1 (2008), 23–35.

[9] Georgios Bakirtzis, Bryan T. Carter, Carl R. Elks, and Cody H. Fleming. 2018. A model-based approach to security analysis for cyber-physical systems. In *2018 Annual IEEE International Systems Conference (SysCon)*. 1–8. DOI: https://doi.org/10.1109/SYSCON.2018.8369518

[10] Tamer Başar and Geert Jan Olsder. 1998. *Dynamic Noncooperative Game Theory, 2nd Edition*. Society for Industrial and Applied Mathematics. DOI: https://doi.org/10.1137/1.9781611971132

[11] Dimitris Bertsimas and Omid Nohadani. 2010. Robust optimization with simulated annealing. *Journal of Global Optimization* 48, 2 (2010), 323–334.

[12] Dimitris Bertsimas, Omid Nohadani, and Kwong Meng Teo. 2010. Nonconvex robust optimization for problems with constraints. *INFORMS Journal on Computing* 22, 1 (2010), 44–58.

[13] Dimitris Bertsimas, Omid Nohadani, and Kwong Meng Teo. 2010. Robust optimization for unconstrained simulation-based problems. *Operations Research* 58, 1 (2010), 161–178.

[14] Arnab Bhattacharya, Thiagarajan Ramachandran, Sandeep Banik, Chase P Dowling, and Shaunak D Bopardikar. 2020. Automated adversary emulation for cyber-physical systems via reinforcement learning. In *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 1–6. DOI: https://doi.org/10.1109/ISI49825.2020.9280521

[15] Steven J. Bradtke. 1994. *Incremental Dynamic Programming for On-Line Adaptive Optimal Control*. Ph.D. Dissertation. Citeseer.

[16] Anna L. Buczak and Erhan Guven. 2015. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials* 18, 2 (2015), 1153–1176.

[17] Nicolo Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge University Press.

[18] Somali Chaterji, Parinaz Naghizadeh, Muhammad Ashraful Alam, Saurabh Bagchi, Mung Chiang, David Corman, Brian Henz, Suman Jana, Na Li, Shaoshuai Mou, Meeko Oishi, Chunyi Peng, Tiark Rompf, Ashutosh Sabharwal, Shreyas Sundaram, James Weimer, and Jennifer Weller. 2019. Resilient Cyberphysical Systems and their Application Drivers: A Technology Roadmap. arXiv:2001.00090. Retrieved from https://arxiv.org/abs/2001.00090.

[19] Thomas M. Chen, Juan Carlos Sanchez-Aarnoutse, and John Buford. 2011. Petri net modeling of cyber-physical attacks on smart grid. *IEEE Transactions on Smart Grid* 2, 4 (2011), 741–749.

[20] Ying Chen, Shaowei Huang, Feng Liu, Zhisheng Wang, and Xinwei Sun. 2018. Evaluation of reinforcement learning-based false data injection attack to automatic voltage control. *IEEE Transactions on Smart Grid* 10, 2 (2018), 2158–2169.

[21] Ye Chen, Yanda Li, Dongjin Xu, and Liang Xiao. 2018. DQN-based power control for IoT transmission against jamming. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, 1–5. DOI: https://doi.org/10.1109/VTCSpring.2018.8417695

[22] Seungoh Choi, Jeong-Han Yun, and Byung-Gil Min. 2021. Probabilistic attack sequence generation and execution based on MITRE ATT&CK for ICS datasets. In *Cyber Security Experimentation and Test Workshop* (Virtual, CA, USA) *(CSET'21)*. Association for Computing Machinery, New York, NY, 41–48. DOI: DOI: https://doi.org/10.1145/3474718.3474722

[23] Peter Dayan and Terrence J. Sejnowski. 1994. TD (λ) converges with probability 1. *Machine Learning* 14, 3 (1994), 295–301.

[24] Seyed Mehran Dibaji, Mohammad Pirani, David Bezalel Flamholz, Anuradha M. Annaswamy, Karl Henrik Johansson, and Aranya Chakrabortty. 2019. A systems and control perspective of CPS security. *Annual Reviews in Control* 47 (2019), 394–411. https://www.sciencedirect.com/science/article/pii/S1367578819300185.

[25] Daniel dos Santos, Clement Speybrouck, and Elisa Costante. 2019. *Cybersecurity in Building Automation Systems*. Technical Report. Forescout Technologies.

[26] Karel Durkota, Viliam Lisy, Branislav Bošansky, and Christopher Kiekintveld. 2015. Optimal network security hardening using attack graph games. In *Proceedings of the 24th International Conference on Artificial Intelligence* (Buenos Aires, Argentina) *(IJCAI'15)*. AAAI Press, 526–532.

[27] Stefano Ermon, Carla Gomes, Ashish Sabharwal, and Bart Selman. 2014. Designing fast absorbing markov chains. *Proceedings of the AAAI Conference on Artificial Intelligence* 28, 1 (Jun. 2014). DOI: https://doi.org/10.1609/aaai.v28i1.8843

[28] Yoav Freund and Robert E Schapire. 1999. Adaptive game playing using multiplicative weights. *Games and Economic Behavior* 29, 1-2 (1999), 79–103.

[29] TN Goh. 1993. Taguchi methods: Some technical, cultural and pedagogical perspectives. *Quality and Reliability Engineering International* 9, 3 (1993), 185–202.

[30] James Hannan. 1957. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games* 3, 2 (1957), 97–139.

[31] Peter J. Hawrylak, Michael Haney, Mauricio Papa, and John Hale. 2012. Using hybrid attack graphs to model cyber-physical attacks in the smart grid. In *2012 5th International Symposium on Resilient Control Systems*. IEEE, 161–164. DOI: https://doi.org/10.1109/ISRCS.2012.6309311

[32] Ashish R. Hota, Abraham A. Clements, Saurabh Bagchi, and Shreyas Sundaram. 2018. *A Game-Theoretic Framework for Securing Interdependent Assets in Networks*. Springer International Publishing, 157–184. DOI:https://doi.org/10.1007/978-3-319-75268-6_7

[33] Mariam Ibrahim and Ahmad Alsheikh. 2019. Automatic hybrid attack graph (AHAG) generation for complex engineering systems. *Processes* 7, 11 (2019). https://www.mdpi.com/2227-9717/7/11/787.

[34] Anna Jaśkiewicz and Andrzej S Nowak. 2015. On pure stationary almost markov nash equilibria in nonzero-sum ARAT stochastic games. *Mathematical Methods of Operations Research* 81, 2 (2015), 169–179.

[35] Donghwoon Kwon, Hyunjoo Kim, Jinoh Kim, Sang C Suh, Ikkyun Kim, and Kuinam J. Kim. 2019. A survey of deep learning-based network anomaly detection. *Cluster Computing* 22, 1 (2019), 949–961.

[36] Harjinder Singh Lallie, Kurt Debattista, and Jay Bal. 2020. A review of attack graph and attack tree visual syntax in cyber security. *Computer Science Review* 35 (2020), 100219. https://www.sciencedirect.com/science/article/pii/S1574013719300772.

[37] Ralph Langner. 2011. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy* 9, 3 (2011), 49–51.

[38] Robert M. Lee, Michael J. Assante, and Tim Conway. 2014. German steel mill cyber attack. *Industrial Control Systems* 30 (2014), 62.

[39] John Leyden. 2008. Polish teen derails tram after hacking train network. *The Register* 11 (2008).

[40] Chong Li and Meikang Qiu. 2019. *Reinforcement Learning for Cyber-Physical Systems: with Cybersecurity Case Studies*. CRC Press. DOI: https://doi.org/10.1201/9781351006620

[41] Jinliang Liu, Liang Xiao, Guolong Liu, and Yifeng Zhao. 2017. Active authentication with reinforcement learning based on ambient radio signals. *Multimedia Tools and Applications* 76, 3 (2017), 3979–3998.

[42] George Louthan, Phoebe Hardwicke, Peter Hawrylak, and John Hale. 2011. Toward hybrid attack dependency graphs. In *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research* (Oak Ridge, Tennessee, USA) *(CSIIRW'11)*. Association for Computing Machinery, New York, NY, Article 62, 1 pages.DOI: https://doi.org/10.1145/2179298.2179368

[43] Xiaozhen Lu, Dongjin Xu, Liang Xiao, Lei Wang, and Weihua Zhuang. 2017. Anti-jamming communication game for UAV-aided VANETs. In *GLOBECOM 2017—2017 IEEE Global Communications Conference*. 1–6. DOI: https://doi.org/10.1109/GLOCOM.2017.8253987

[44] Yuriy Zacchia Lun, Alessandro D'Innocenzo, Francesco Smarra, Ivano Malavolta, and Maria Domenica Di Benedetto. 2019. State of the art of cyber-physical systems security: An automatic control perspective. *Journal of Systems and Software* 149 (2019), 174–216. https://www.sciencedirect.com/science/article/pii/S0164121218302681.

[45] Mayra Macas and Wu Chunming. 2013. Enhanced cyber-physical security through deep learning techniques. In *2019 Proceedings of the Cyber-Physical Systems PhD Workshop*. 72–83. Retrieved from http://ceur-ws.org/Vol-2457/8.pdf.

[46] J. P. McDermott. 2001. Attack net penetration testing. In *Proceedings of the 2000 Workshop on New Security Paradigms* (Ballycotton, County Cork, Ireland) *(NSPW'00)*. Association for Computing Machinery, New York, NY, 15–21. DOI: https://doi.org/10.1145/366173.366183

[47] Fei Miao, Quanyan Zhu, Miroslav Pajic, and George J. Pappas. 2016. Coding schemes for securing cyber-physical systems against stealthy data injection attacks. *IEEE Transactions on Control of Network Systems* 4, 1 (2016), 106–117.

[48] Erik Miehling, Cedric Langbort, and Tamer Başar. 2020. Secure contingency prediction and response for cyber-physical systems. In *2020 IEEE Conference on Control Technology and Applications (CCTA)*. 998–1003. DOI: https://doi.org/10.1109/CCTA41146.2020.9206253

[49] Erik Miehling, Mohammad Rasouli, and Demosthenis Teneketzis. 2015. Optimal defense policies for partially observable spreading processes on bayesian attack graphs. In *Proceedings of the 2nd ACM Workshop on Moving Target Defense* (Denver, Colorado, USA) *(MTD'15)*. Association for Computing Machinery, New York, NY, 67–76.DOI: https://doi.org/10.1145/2808475.2808482

[50] Erik Miehling, Mohammad Rasouli, and Demosthenis Teneketzis. 2022. *Control-Theoretic Approaches to Cyber-Security*. Springer-Verlag, Berlin, 12–28. DOI: https://doi.org/10.1007/978-3-030-30719-6_2

[51] Luan Nguyen and Vijay Gupta. 2021. Towards a framework of enforcing resilient operation of cyber-physical systems with unknown dynamics. *IET Cyber-Physical Systems: Theory & Applications* 6, 3 (2021), 125–138. DOI: https://doi.org/10.1049/cps2.12009

[52] Thanh Thi Nguyen and Vijay Janapa Reddi. 2021. Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–17. DOI: https://doi.org/10.1109/TNNLS.2021.3121870

[53] Zhen Ni and Shuva Paul. 2019. A multistage game in smart grid security: A reinforcement learning solution. *IEEE Transactions on Neural Networks and Learning Systems* 30, 9 (2019), 2684–2695.

[54] Felix O. Olowononi, Danda B Rawat, and Chunmei Liu. 2021. Resilient machine learning for networked cyber physical systems: A survey for machine learning security to securing machine learning for CPS. *IEEE Communications Surveys & Tutorials* 23, 1 (2021), 524–552. DOI: https://doi.org/10.1109/COMST.2020.3036778

[55] Kyuchan Park, Bohyun Ahn, Jinsan Kim, Dongjun Won, Youngtae Noh, Jinchun Choi, and Taesic Kim. 2021. An advanced persistent threat (APT)-style cyberattack testbed for distributed energy resources (DER). In *2021 IEEE Design Methodologies Conference (DMC)*. 1–5. DOI: https://doi.org/10.1109/DMC51747.2021.9529953

[56] Martin Pelikan, David E. Goldberg, and Erick Cantú-Paz. 1999. BOA: The bayesian optimization algorithm. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation—Volume 1* (Orlando, Florida) *(GECCO'99)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 525–532.

[57] Damien Picard, Ján Drgoňa, Michal Kvasnica, and Lieve Helsen. 2017. Impact of the controller model complexity on model predictive control performance for buildings. *Energy and Buildings* 152 (2017), 739–751. https://www.sciencedirect.com/science/article/pii/S0378778817302190.

[58] Victor Picheny, Mickael Binois, and Abderrahmane Habbal. 2019. A bayesian optimization approach to find nash equilibria. *Journal of Global Optimization* 73, 1 (2019), 171–192.

[59] PNNL. 2019. Python Systems Library. Retrieved from https://github.com/pnnl/psl.

[60] Tereza Pultarova. 2016. Cyber security-Ukraine grid hack is wake-up call for network operators [news briefing]. *Engineering & Technology* 11, 1 (2016), 12–13.

[61] Tirukkannamangai E. S. Raghavan, S. H. Tijs, and O. J. Vrieze. 1985. On stochastic games with additive reward and transition structure. *Journal of Optimization Theory and Applications* 47, 4 (1985), 451–464.

[62] Sudip Saha, Anil Vullikanti, and Mahantesh Halappanavar. 2017. FlipNet: Modeling covert and persistent attacks on networked resources. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 2444–2451. DOI: https://doi.org/10.1109/ICDCS.2017.298

[63] Sudip Saha, Anil Kumar S. Vullikanti, Mahantesh Halappanavar, and Samrat Chatterjee. 2016. Identifying vulnerabilities and hardening attack graphs for networked systems. In *2016 IEEE Symposium on Technologies for Homeland Security (HST)*. 1–6. DOI: https://doi.org/10.1109/THS.2016.7568884

[64] Dinuka Sahabandu, Shana Moothedath, Joey Allen, Linda Bushnell, Wenke Lee, and Radha Poovendran. 2019. Stochastic dynamic information flow tracking game with reinforcement learning. In *Decision and Game Theory for Security*, Tansu Alpcan, Yevgeniy Vorobeychik, John S. Baras, and György Dán (Eds.). Springer International Publishing, Cham, 417–438.

[65] Dinuka Sahabandu, Shana Moothedath, Joey Allen, Linda Bushnell, Wenke Lee, and Radha Poovendran. 2021. A Reinforcement Learning Approach for Dynamic Information Flow Tracking Games for Detecting Advanced Persistent Threats. arXiv:2007.00076. Retrieved from https://arxiv.org/abs/2007.00076.

[66] Dinuka Sahabandu, Baicen Xiao, Andrew Clark, Sangho Lee, Wenke Lee, and Radha Poovendran. 2018. DIFT games: Dynamic information flow tracking games for advanced persistent threats. In *2018 IEEE Conference on Decision and Control (CDC)*. 1136–1143. DOI: https://doi.org/10.1109/CDC.2018.8619416

[67] Aaron Schlenker, Omkar Thakoor, Haifeng Xu, Fei Fang, Milind Tambe, Long Tran-Thanh, Phebe Vayanos, and Yevgeniy Vorobeychik. 2018. Deceiving cyber adversaries: A game theoretic approach *(AAMAS'18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 892–900.

[68] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. 2012. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory* 58, 5 (2012), 3250–3265.

[69] G. Edward Suh, Jae W. Lee, David Zhang, and Srinivas Devadas. 2004. Secure program execution via dynamic information flow tracking. *ACM Sigplan Notices* 39, 11 (2004), 85–96.

[70] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction.* MIT Press.

[71] Richard S. Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. 2009. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, Quebec, Canada) *(ICML'09)*. Association for Computing Machinery, New York, NY, 993–1000.DOI: https://doi.org/10.1145/1553374.1553501

[72] Huan Wang, Zhanfang Chen, Jianping Zhao, Xiaoqiang Di, and Dan Liu. 2018. A vulnerability assessment method in industrial internet of things based on attack graph and maximum flow. *IEEE Access* 6 (2018), 8599–8609. DOI: 10.1109/ACCESS.2018.2805690

[73] Chathurika S. Wickramasinghe, Daniel L. Marino, Kasun Amarasinghe, and Milos Manic. 2018. Generalization of deep learning for cyber-physical system security: A survey. In *IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society*. 745–751. DOI: https://doi.org/10.1109/IECON.2018.8591773

[74] Liang Xiao, Yan Li, Guoan Han, Guolong Liu, and Weihua Zhuang. 2016. PHY-layer spoofing detection with reinforcement learning in wireless networks. *IEEE Transactions on Vehicular Technology* 65, 12 (2016), 10037–10047.

[75] Xin Xu, Lei Zuo, and Zhenhua Huang. 2014. Reinforcement learning algorithms with function approximation: Recent advances and applications. *Information Sciences* 261 (2014), 1–31. https://www.sciencedirect.com/science/article/pii/S0020025513005975.

[76] Jun Yan, Haibo He, Xiangnan Zhong, and Yufei Tang. 2016. Q-learning-based vulnerability analysis of smart grid against sequential topology attacks. *IEEE Transactions on Information Forensics and Security* 12, 1 (2016), 200–210.

[77] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. *Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms*. Springer, Cham, 321–384.

[78] Yichi Zhang, Yingmeng Xiang, and Lingfeng Wang. 2016. Power system reliability assessment incorporating cyber attacks against wind farm energy management systems. *IEEE Transactions on Smart Grid* 8, 5 (2016), 2343–2357.

Received 8 July 2021; revised 1 December 2022; accepted 11 April 2023