IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 February 2024, accepted 3 April 2024, date of publication 18 April 2024, date of current version 10 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3390722

RESEARCH ARTICLE

Leveraging Deep Reinforcement Learning Technique for Intrusion Detection in SCADA Infrastructure

FRANTZY MESADIEU, DAMIANO TORRE, (Member, IEEE), AND ANITHA CHENNAMANENI, (Member, IEEE)
Subhani Department of Computer Information Systems, Texas A&M University Central Texas, Killeen, TX 76549, USA

Corresponding author: Anitha Chennamaneni (anitha.chennamaneni@tamuct.edu)

This work was supported in part by the Air Force Research Laboratory (AFRL) and the Department of Homeland Security (DHS) Science and Technology Directorate (S&T) under Award FA8750-19-C-0077.

ABSTRACT The prevalence of cyber-attacks perpetrated over the last two decades, including coordinated attempts to breach targeted organizations, has drastically and systematically exposed some of the more critical vulnerabilities existing in our cyber ecosystem. Particularly in Supervisory Control and Data Acquisition (SCADA) systems with targeted attacks aiming to bypass signature-based protocols, attempting to gain control over operational processes. In the past, researchers utilized deep learning and reinforcement learning algorithms to mitigate threats against industrial control systems (ICS). However, as technology evolves, these techniques become ineffective in monitoring and enhancing the cybersecurity defenses of those system against unwanted attacks. To address these concerns, we propose a deep reinforcement learning (DRL) framework for anomaly detection in the SCADA network. Our model utilizes a ''Q-network'', which allows it to achieve state-of-the-art performance in pattern recognition from complex tasks. We validated our solution on two publicly available datasets. The WUSTL-IIoT-2018 and the WUSTL-IIoT-2021, each comprised of twenty-five networking features representing benign and attack traffic. The results obtained shows that our model successfully achieved an accuracy of 99.36% in attack detection, highlighting DRL's potential to enhance the security of critical infrastructure and laying the foundation for future research in this domain.

INDEX TERMS Critical infrastructure, deep reinforcement learning, cybersecurity, SCADA.

I. INTRODUCTION

Cybersecurity threats and attacks against businesses are escalating and becoming increasingly sophisticated, making them extremely challenging to detect. While traditional antivirus software with pattern recognition algorithms is commonly used for early detection, however, the complexity and frequency of attacks have made it difficult for network administrators and standard applications to monitor. Moreover, the relentless number of attacks occurring each day has also added to the growing challenge [1]. Among the domains and software that are vulnerable and susceptible to cyber-attacks, is autonomous critical infrastructure or Supervisory Control and Data Acquisition (SCADA). In an industrial control system, such as ''critical infrastructure'' these systems enhance and automate equipment's performances [2]. For example, a SCADA-controlled smart grid infrastructure would leverage internet protocols to communicate with sensors to detect faults and isolate potential damages to power lines and mechanical field assets responsible of performing daily operations [3], [4].

With the advent of the Internet, coupled with lurking persistent cyber-threats and the proliferation of the Internet of Things (IoT), a new paradigm has shifted towards artificial intelligence (AI) and the dynamism of ''deep reinforcement learning'' (DRL). Based on prior knowledge, this technique is used to navigate through new obstacles to solve future problems. The methodological complexity and deep neural

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed Mohamed.

network (DNN) algorithmic policy-based that interconnects the model's neurons, gives it significant computational aptitude to analyze intricate network data and make decisions [5]. In terms of multi-tasking capabilities, DRL technique provides researchers with an array of advanced AI technical frameworks ready to be deployed in various domains of industrial importance. In areas such as smart grids for voltage regulation [2]; adaptive assembly lines [6]; in robotics to perform specific functions that optimized individual tasks [7]. For example, in cybersecurity [8], the DRL algorithm is applied to filter through network traffic and prevent intrusions [9], thus reducing cost and the unnecessary manpower needed to accomplish complex and time-consuming cyber-monitoring tasks. Conceptually, DRL technique is trained in a pre-defined environment where agents learn from low dimension of feature's inputs from metadata and perform ***meta-learning***[1] through trial and error [10], [11].

As a subset of machine learning, DRL's computational proficiency and intellectual effectiveness in the video gaming industry display exceptional results in both offensive attack strategies as well as defensive tactics [12]. DRL's neural network application uses a function estimator to observe data state or labels' input, as it operates in a set environment, and uses a greedy algorithmic policy to process unlimited network traffic's history for anomalous and malicious attacks [13]. In retrospect, these capabilities make it a valuable intrusion detection algorithm for protecting and defending against sophisticated cybersecurity attacks, including advanced persistent threats (APT) [14].

In the digital transformation era, various sectors such as business infrastructure, banking, IoT, Industrial IoT, and transportation, including SCADA infrastructure, are under constant threats of cybersecurity attacks and the risk of data breaches [14]. However, in recent years, machine learning techniques have shown great promise in enhancing the security posture of critical infrastructure by aiding in penetration testing, pattern detection, and real-time attack mapping. However, it should be noted that the complexity of code and the challenges posed by large datasets remain an obstacle for certain aspects of machine learning algorithms [15], [16], particularly in areas involving continuous complex calculations and large-scale decision-making.

In this paper, we conducted a qualitative assessment of cyber-attacks on critical infrastructure and their impact on business operations, drawing insights from studies, e.g., [17] and [18]. We performed a comprehensive review of prior research on the application of DRL methodology in combating SCADA network anomalies; our analysis revealed that while there have been numerous research efforts utilizing DRL approaches to enhance the security of SCADA network infrastructure, however, the focus has been primarily directed towards hardware and the monitoring aspects of (CI). Including the development of intrusion detection methods and autonomous controls using diverse datasets [19], [20], [21]. In fact, we did not find any relevant papers on DRL that used similar data at the time of our research. Considering these findings, we explored the effectiveness and robustness of the DRL framework [22] in the cybersecurity domain. We developed and implemented a model algorithm, testing it on two SCADA datasets: Wustl-IIoT-2018 and Wustl-IIoT-2021. Our research results demonstrated the successful application of DRL methodology in detecting cyber threats within SCADA's critical network [14].

[1]**Meta-learning** in ML is the process that refers to learning algorithms that continuously learn from other learning algorithms.

A. RESEARCH OBJECTIVES

The main goal of this work is to explore the capabilities of Deep Reinforcement Learning (DRL) algorithms to enhance the accuracy of anomaly detection.

In particular, this research explores the potential of implementing a DRL technique to protect and defend critical infrastructure from continuous cybersecurity attacks as we investigate whether the algorithm can effectively enhance the security and resilience of SCADA systems against ongoing threats from bad actors.

Similarly, we explore possible challenges associated with applying DRL in Smart Grid systems and discuss specific difficulties or limitations in implementing the technique in the context of critical systems' network anomaly detection, which encompasses protection from cybersecurity attacks.

We discuss optimization and deployment strategies for utilizing the technique to effectively secure SCADA systems and provide insights into specific approaches, methodologies, and necessary considerations for maximizing the efficacy of DRL to improve network security.

Lastly, we identify the potential advantages and disadvantages of deploying DRL as an Intrusion Detection System and Intrusion Prevention System in Industrial Control Systems [3], [23], [24]. This approach seeks to analyze how DRL can enhance the security and resilience of ICS to effectively detect and prevent cybersecurity threats, improve incident response, and mitigate risks to critical infrastructure.

B. CONTRIBUTIONS

Our proposed approach presents a multifold contribution to critical infrastructures' existing security measures.

A) It incorporates novel features such as actor-critic algorithm [23], designed for optimal policy update [24], to dynamically assist the model (actor) in maximizing its decision process. The architectural design of the actor and critic networks are mirrored with the same dynamic parameters to facilitate a reciprocal training environment. In this manner, our actor is trained to generate actions that maximize the expected rewards, as the critic evaluates the quality of those actions for the expected value of future rewards. This iterative process continues until the model converges.

To address issues stemming from correlated data and non-stationary distributions, we introduce a "ReplayMemory" function [3] designed to store experiences and sample transitions (*i.e.*, *state*−*action*−*reward*−*next*−*state*, *tuples*) that the agent encounters during its exploration of the environment. This memory buffer enables the agent to learn from past experiences by randomly sampling previous transitions from the memory during training. By incorporating this technique, the agent can leverage a wider range of experiences from its entire history rather than relying solely on its most recent encounters.

B) We present a detailed assessment and conduct a qualitative and quantitative evaluation using two real-world datasets to train our model. We evaluate our algorithm on the WUSTL-IIoT-2018 dataset [25], and the WUSTL-IIoT-2021 dataset [26], consisting of network traffic protocols. To validate our solution, we load the saved model with test data containing multiple binary classification inputs. The agent applies the methods described in algorithms (6) and (7) to read the labeled attacks and predict the targeted attack labels. The results obtained, demonstrates that our DRL model can effectively classifies threats in real time and provides detection and response [27].

C) Lastly, we offer valuable insights for future research directions regarding the use of DRL to detect cybersecurity attacks in the SCADA domain. Despite its successes, some of the challenges faced by DRL technique applications are: Deep neural network utilization, greedy policy implementation, and the need for sample or data efficiency. In addition to these challenges, there are several points of interest for future research on DRL implementation in the cybersecurity domain. These include: Investigating the capability of DRL framework to handle larger and more complex SCADA network datasets; evaluating the robustness and generalizability of the technique by testing it on diverse SCADA systems and network environments; and exploring the possibility of combining the DRL algorithm with other machine learning techniques, such as anomaly detection and ensemble methods, to further improve the accuracy and effectiveness of cyber threat detection and response. These research directions aim to address challenges in adopting DRL for cybersecurity and advance the development of active solutions in this field.

C. STRUCTURE

The remainder of the paper is structured as follows: Section II details a comprehensive background of DRL and highlights key topological concepts of SCADA. Section III reviews prior work that employs DRL methodology for anomaly detection in SCADA. Section IV describes the methodology used to build our DRL model and the application of the model to SCADA domain. Section C details the training process and the implementation of our DRL model. Section VI presents the results of our evaluation. Section VII outlines the threats to validity. Section VIII concludes the paper.

II. BACKGROUND

In this section, we provide a brief introduction of DRL framework. Following that, we summarize the necessary background information related to SCADA infrastructure.

A. DEEP REINFORCEMENT LEARNING

Deep reinforcement learning (DRL), a subset of machine learning, is a branch of artificial intelligence that combines deep learning algorithms and reinforcement learning techniques allowing an agent to interact with its environment and learn from trial and error. Using Python programming language, we defined the following: an agent is represented through functions, software, or a block of codes. The environment, on the other hand, is a function that simulates both virtual and physical training scenarios, containing parameters for an agent to exploit. In this conceptual framework (Fig.1), the agent explores a pre-defined environment and exploits its parameters [28] to maximize reward signals.

In training, for each time step, the environment sends a scalar reward signal to the reinforcement learning agent for each action taken. It is important to highlight that due to the insistent algorithmic nature of DRL framework, the sole objective of the agent is to learn from its implemented stochastic policy to maximize cumulative reward over time [29].

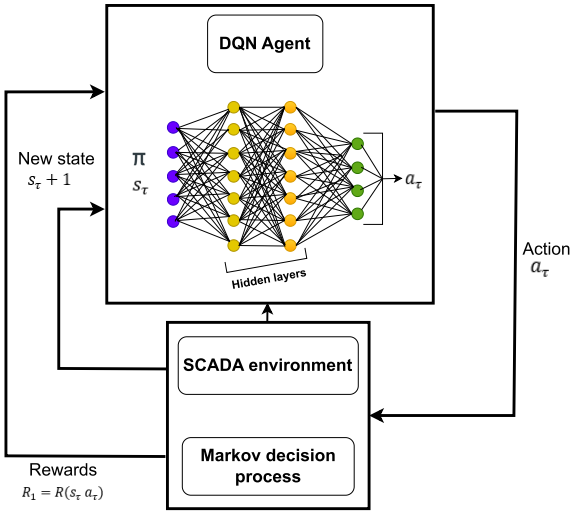

FIGURE 1. DQN Agent in SCADA Environment Using MDP.

A typical DRL framework has several key components. An essential element in the software design is the "***environment***", representing the task or challenges the agent is attempting to resolve [30], as depicted in *figure 1*. In that context, the environment facilitates the agent with observations, thus prompted an action (a), state (s) that geared towards a signal reward (r). This virtual cyber-space is usually a high-dimensional sensory inputs, with expected rewards, indicating the agent's performance based on the quantity of rewards obtained [15]. The second component in DRL framework is the "***agent***", this methodology employs

deep neural network, hence enabling the agent to learn and interact with the environment. During training and upon initialization, the agent uses data input as observations and its outputs as actions, while attempting to maximize reward signals, where $Q(s, a)$ represents the sum of all rewards, and $maxQ(s', a')$ for the maximum rewards an agent is able to achieve from its current state [31]. The third component is the "***training algorithm***", which updates the deep neural network's weights based on the reward function and the actions taken by the agent. In our proposed model, the state-action value function is called a Q function, and is defined as $Q(s, a)$. Referred to as Q-learning, this Q function is used for optimal rewards in each action as it updates the policy using this Bellman equation [24].

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma maxQ(s', a') - Q(s, a))$$

1) DEEP Q-NETWORK ALGORITHM

Deep Q-Network (DQN) is a deep reinforcement learning algorithm that uses deep neural networks or a Q-network to approximate optimal action-value functions in a Markov Decision Process (MDP) [32]. The technique is built upon the traditional Q-learning algorithm, which serves as a function approximator. Unlike reinforcement learning which maintains a ($Q - table\ to\ store\ Q - values\ for\ each\ state - action\ pair$), Q-network takes the raw observation state as input and directly outputs the estimated Q-values for all possible actions. Introduced by Google DeepMind in 2013 [33], the DQN technique combines deep neural networks and Q-learning to learn optimal policies in complex and high-dimensional environments [34].

2) IMPROVING DQN ACTION SELECTION USING TEMPORAL DIFFERENCE

Using MDP principle, we initialized the DQN ReplayMemory with S_t, A_t, where the model takes random action in the environment and uses experience replay samples to update Q-values. However, from a policy-based perspective, the algorithm is designed for optimal performance, from which an RL agent's goal is to select a policy that optimizes its expected return (V-value function):

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r + \gamma V_\pi(s')]$$

This aggressive approach may lead to (TD) or Temporal Difference learning which may create noise [35], [36]. To calculate TD for reward maximization, where $a_t = s_t$, we use $Q(s_{t+1}, a')$ for immediate reward $R = Q(s_{t+1}, a')$. By adding $Q(s_t, a_t)$, we were able to update the difference between TD-target, using a as the learning rate for each added $Q(s_t, a_t)$ [37].

B. SCADA

The Supervisory Control and Data Acquisition System (SCADA), is a modern computerized control system comprised of hardware components, software, and network data communications. Known as the Industrial Internet of Things or IIoT 4.0 [38], these peripherals enable remote automation and high-level supervision of critical infrastructures like power grids and water distribution plants. The system is an integral component of smart grids and facilitates real-time data analysis, command, and control processes of assets. Depicted in figure 2, a standard SCADA structural design can be outfitted with "Remote Terminal Units, Programmable Logic Controller, etc." These microprocessors communicate commands to field devices such as Pump units and valves, where the processes are then displayed in a "human-machine interface" HMI for visual confirmation [39].

1) SCADA OPTIMIZATION PROCESS

- Control processes
- Monitor and gather information in real-time
- Interact with various devices
- Record events in a back-end database

Based on figure 2, regardless of topological configuration and structural design, a SCADA system uses data from its connected nodes or (IIoT) devices to perform its operations. These nodes are a collection of sensors and other monitoring devices attached to the network via ethernet cables or wireless communication channels [40]. Depicted in 3, remote control access requires internet communication and uses the following mode for data exchange: "Local Area Network" (LAN) or a "Wide Area Network" (WAN). Protocols such as Modbus, DNP3, OPC, and others, defined the format and rules for exchanging data and commands between nodes within a system [40].

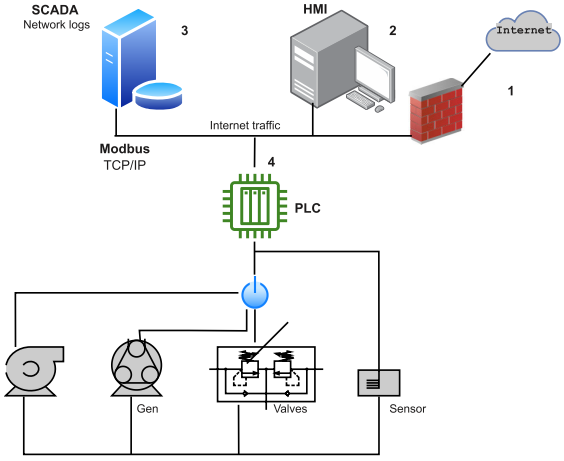

FIGURE 2. SCADA network topology.

2) THREAT ANALYSIS

This section outlines the application of threat analysis in the context of SCADA systems operational continuity and its relevance with the integration of Industry 4.0.

Threat analysis helps identify potential vulnerabilities and assess their impact on business operations. It is a quantitative and qualitative assessment of critical organizational resources

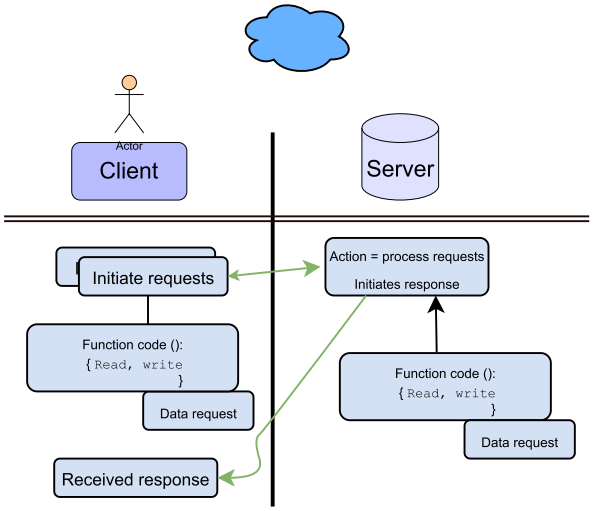

FIGURE 3. Modbus server.

to determine the business impact on Information Technology and Information Systems IT/IS [41], [42].

a: INDUSTRY 4.0 VULNERABILITY

IoT and IIoT devices integrated in SCADA infrastructure, enable advanced automation and cost effectiveness of operational processes. However, the risks associated with critical infrastructure are derived from the programmable logic controllers (PLCs) installed in components responsible for executing tasks upon commands [39]. SCADA's physical structure is comprised of PLCs, such as actuators, sensors, enterprise resource planning, and manufacturing execution system (MES) for asset tracking and management [43]. Together, vulnerabilities like "insecure software/firmware, insecure web interface, lack of transport encryption, insufficient authentication, can all be exploited as part of the system's flaws.

b: IDENTIFYING AND DETERMINING RISKS

SCADA communication links in relation to topological structure, functionality, and proximity, make the use of internet protocol (e.g., TCP/IP) essential to its operation; as a result, IoT devices connected to network traffic facing the internet are likely to become targets of malicious attacks [44], [43]. Meaning, both hardware and software components, such as PLCs,[2] RTUs,[3] sensors and other controllers performing data communication and acquisitions, can also be exposed to unwanted attacks when using internet resources (e.g., local area network (LAN) and wide area network or (WAN) to carry out critical functions). In lieu of these identifiable risks, a network attack such as Stuxnet[4] on IoT devices can disrupt an entire system and causes great financial loss in terms of the business impact to consumers and stakeholders [43], [45]. SCADA software and IoT devices can be subjected to various types of security attacks [41], especially when considering and evaluating communications' links with respect to software-related vulnerabilities. Although multiple AI techniques and frameworks have been developed in conjunction with cybersecurity taxonomies for intrusion detection and mitigation, unfortunately, these techniques are not designed to address software flaws.

III. RELATED WORK

In retrospect to threat prevention and detection, our main focus is to develop a cutting-edge DRL algorithm that incorporates network communication protocols to proactively detect network intrusion in SCADA's industrial control systems and field assets. To guide our search, we referred to the systematic mapping conducted in [46], which helps identify existing machine learning techniques that can be leveraged to analyze network traffic behaviors for threat anomalies. Surprisingly, most of the research conducted on DRL in this domain focused on anomaly detection, but utilized diverse datasets, including proprietary data, to address specific SCADA network security challenges. Table 1 provides a summary of the reviewed studies, including the related work, datasets employed, and their respective domains.

For example, Liu et al. [3] developed a DRL framework that monitors IIoT components in physical water plant to investigate the impact of a performance-based attack on IoT devices. The authors' proposed DQN experimented on proprietary data, collected from a simulated water distribution testbed. In [14], the deep RL-based APT defense scheme introduced by Ning and his associates, combined deep learning and policy-gradient based actor-critic to identify "advanced persistent threats" and determine the type of resources needed to manage both the speed and the interval at which attacks are launched against critical infrastructure. Tharewal et al. [15] presented a DRL-based intrusion detection system for the industrial Internet of Things, to detect network intrusions that are otherwise too complex for conventional machine learning techniques to identify. Introduced by Landen et al. in [17], DRAGON is a DRL model, structured to actively monitor potential cyber-attacks in smart grids through interactions, and data collection, from which the model learned from past experiences using a grid simulator, featuring IEEE 14-bus power transmission system as its environment. Wei et al. [18] developed a DRL framework that serves as a transmission retriever and optimizer of lost communications during a cyber-attack. The model's objective is to minimize reclosing time among affected communication lines to both prevent

[2]**PLC** programmable logic controller; a microprocessor that communicates commands to field devices.

[3]**RTU** Remote Terminal Unit, is a microprocessor that monitors & controls field devices.

[4]**Stuxnet** a self-replicating malware that takes advantage of auto-execution vulnerabilities.

TABLE 1. Related work comparison.

Refs.	Year	Dataset	Domain/System	DRL Main goal
Liu et al. [3]	2021	Private (IIoT)	RTU water distribution	On security for Industrial Internet of Things
Ning and Liang [14]	2021	Private	Smart Grids	Defense Against Advanced Persistent Threats in Smart Grids: A Reinforcement Learning Approach
Tharewal et al. [15]	2019	IIoT	Network interface	IDS for Industrial Internet of Things Based on DRL
Landen et al. [17]	2022	IEEE 14-bus	Network interface	DRAGON: DRL for Autonomous Grid Operation and Attack Detection
Wei et al. [18]	2020	IEEE 9-bus	Smart Grids	Cyber-Attack Recovery Strategy for Smart Grid Based on DRL
Wang et al. [21]	2020	IEEE 30-bus	Smart Grids	Coordinated Topology Attacks in Smart Grid Using DRL
Moradi et al. [29]	2022	W&W 6-bus, IEEE 30-bus	SCADA	Defending Smart Electrical Power Grids against Cyberattacks with Deep Q-Learning
Wang et al. [44]	2021	Gas pipeline	ICS/RTU	Abnormal flow detection in industrial control network based on DRL

and recover from network transmission attacks. The proposed methodology presented by Wang et al. [21] is a DRL based Q-learning attack strategy, simulated on the IEEE 30-bus system for SCADA load management uncertainty; a vulnerability that a hacker may use to trip critical transmission lines. Their model detects and prevents smart grid-coordinated topological attacks by identifying false electronic communication. The actor-critic approach by Moradi et al. [29] is a high-level algorithm, structured to strategically simulate attacks in a smart environment. The technique can simultaneously learn policies and detect network anomalies from smart electrical communication systems. The evaluation shows positive results against the Wood and Wollenberg 6-bus and the IEEE 30-bus systems respectively. The abnormal flow detection in industrial control network presented by Wang et al. [44] is a DRL model which the authors deploy to monitor ''abnormal flow'', a form of intrusion in ICS systems. The model helps prevent bad actors from taking over command of industrial control systems or natural gas pipeline operations. To the best of our knowledge, our study is the first to implement a DQN in SCADA ''Industrial Internet of Things'' for intrusion detection. In contrast, the gap between our proposed algorithm and the related work comparison presented in table 5, lies in the sophistication of our model's architecture, the complexity of our datasets and the performance achieved, highlighting our algorithm's advancements in handling complex tasks, and preventing continuous cyberterrorism attacks against SCADA communication's infrastructure.

IV. METHODOLOGY

As follows, we present the research methodology employed in constructing our DRL model, which is based on our investigative discoveries [38].

Our research aims to explore the potential of DRL techniques in the cybersecurity domain, specifically focusing on optimizing and deploying these techniques to enhance the security of industrial control systems. By leveraging DRL, we aim to strengthen the resilience of SCADA systems against various cyber threats. Here, we introduce the four-step methodology:

1. **Building the DRL Model** (Section IV-A). We developed a DQN-based approach that effectively addresses the challenges in protecting and defending critical infrastructure against cybersecurity attacks using complex IIoT datasets.
2. **Applying DRL to SCADA** (Section IV-B). We describe how the DRL model can be applied to SCADA domain.
3. **Optimizing DRL to SCADA** (Section IV-C). We detail the steps to take in order to optimize our DRL model to SCADA domain.
4. **Monitoring SCADA with DRL** (Section IV-D). We elaborate on the monitoring techniques that enable our model to identify potential attacks on SCADA systems.

Following this, we offer a detailed breakdown of the four steps.

A. BUILDING THE DRL MODEL

In this paper, we utilize datasets within the IoT and IIoT domain.

As follows, we outline some of the primary protocols and software deficiencies affecting *IIoT* with respect to SCADA, highlighting how these discrepancies continue to jeopardize the security of smart grid systems. In reference to the framework of IoT vulnerabilities outlined in [40] and [47], we categorize security threats in IIoT devices to include the following vulnerabilities:

- *Outdated software*: Inadequate access control, which may allow unauthorized access.

Algorithm 1 DRL Agent Action Algorithm for Unexpected Queries

Input: Query q
Output: Action a
if q *contains restricted content* **then**
  if *anomaly detected* **then**
    initiate countermeasures to neutralize request;
    a = neutralize;
  end
  else
    raise alert;
    a = raise-alert;
  end
else
  allow request;
  a = allow;
end

- *Poor network segmentation*: The absence of encryption causes security vulnerabilities within the network of devices and equipment used in industrial settings.

These limitations pose the most significant security concerns in SCADA operations. Therefore, implementing DRL as an intrusion detection system (IDS) and as a network intrusion prevention system (NIPS) can potentially enhance cybersecurity defenses against continuous attacks in critical infrastructure [18]. Our approach, can analyze past attacks and simulate threat scenarios to learn and adapt to new threats in real-time. Utilizing a concept called Deep Q-Learning as contrast in Figure 4, the proposed model intelligently uses meta-leaning[1] for continuous decision-making to maximize rewards [17].

Because unlike *Q-learning* [37], which uses a Q-table (Fig. 4) to store expected rewards for each state-action pair [32], "Deep Q-Learning" architecture, however, uses a dual neural network and weights differences for its learning processes [48]; the added neural networks allow a trained DRL agent to identify patterns of malicious behaviors and take appropriate actions to mitigate threats, such as detecting anomalies and unusual network traffic in the system's data. For example, if a DRL agent detects any unexpected queries made to a SCADA client's server, attempting to access restricted content [49], as shown in Figure 3, the agent can take deliberate actions, such as initiating countermeasures as seen in (Algorithm 1) to neutralize the request if an anomaly is detected [14]. These measures may include blocking network traffic from suspicious requests to isolating affected parts of the system [3].

For our proposed scheme, the "Deep Q-learning" technique is implemented, because it combines "Convolutional Neural Network" with the classical Q-learning to approximate rewards signal [29], [32]. In this way, training our agent to interact with the SCADA system and learn from its actions in a complex environment has proven to be fully

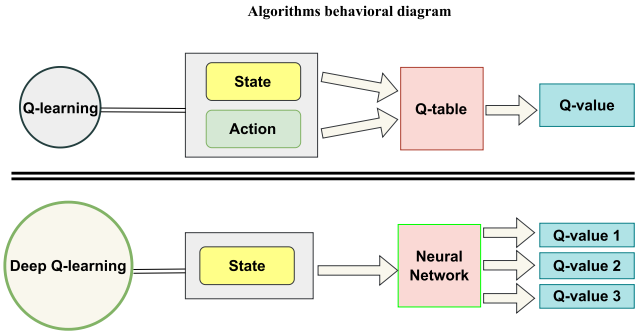

FIGURE 4. Q-learning vs Deep Q-learning.

capable of detecting and mitigating cybersecurity threats in real time. Based on our proposed approach, it is evident that with the Deep Q-learning algorithm, when paired with our (ϵ) epsilon greedy policy, the result greatly enhances our model's performance and in turn strengthens the security of SCADA infrastructure [48].

1) PROPOSED MODEL ALGORITHM STRUCTURE

In the following manner, we present a detailed description of our DRL-based model:

1) The structural design of our QNetwork, leverages the Bellman equations [50], [51] to compute and express actions' value taking in a given state, with respect to the sum of the immediate reward and the discounted value of the next state:
   $\gg$ $Q(s,a) = r(s,a) + \gamma \cdot max'_a(Q(s',a'))$ where:
   - Q(s,a) is the expected value of action's taken a in state s
   - $r(s,a)$ is the immediate reward received for action's taken a in state s
   - γ is the discount factor, which determines how much the agent values future rewards over immediate rewards
   - s' is the next state that the agent transitions to after taking action a in state s
   - $max'_a(Q(s',a'))$ is the maximum expected value over all possible actions a' in the next state s'
2) Policy Gradient: Our model uses this policy in conjunction with an epsilon greedy policy to optimize the objective function directly [52] by computing the gradient of the expected reward with respect to the policy parameters:
   $\gg$ $\nabla_\theta J(\theta) = E[\sum_t = 0^T \nabla_\theta log\pi_\theta(a_t|s_t) \cdot A_t]$ where:
   - $J(\theta)$ is the expected reward (also known as the objective function) for a given policy parameterized by θ
   - $\pi_\theta(a_t|s_t)$ is the probability of action's taken a_t in state s_t under the policy parameterized by θ
   - A_t is the advantage function, which measures how much better the agent performed in state s_t compared to the expected reward in that state.

3) Actor-Critic: In our model, we implemented an actor-critic algorithm (2) [23], combined with our policy gradient which updates the training agent's (actor) expected reward for a given state-action pair:
$\gg \delta_t = r_t + \gamma \cdot V(s_{t+1}) - V(s_t)\theta < -\theta + \alpha_\theta \cdot \delta_t \cdot \nabla_\theta log\pi_\theta(a_t|s_t)V(s_t) < -V(s_t) + \alpha_V \cdot \delta_t$
where:
- δ_t is the temporal difference (TD) error, which measures the difference between the predicted value of the current state-action pair and the actual reward received.
- $V(s_t)$ is the estimated value of state s_t
- α_θ and α_V are the learning rates for the policy and value function, respectively.

Algorithm 2 Actor critic algorithm implementation

Input: Initialize policy, two soft Q and two target soft Q-DNNs; Initialize experience replay buffer with attack samples; **Result**: Optimised actor and critic DNNs **for** *each episode* **do**
  for *each step* **do**
    sample actions from the policy; sample transition from the environment; store the transition in the replay buffer;
  end
  for *each gradient update step* **do**
    update the soft DQNetwork weights; update the policy DQNetwork weights; adjust the entropy temperature; update the target DQNetwork weights;
  end
end

2) ON-POLICY VS OFF-POLICY

Actor-critic algorithms can be on-policy or off-policy. These methods are used in DRLs when determining whether the data collected during training is used for updating the policy or remains neutral.

The main objective of this paper is to develop a DRL framework that improves network security with a robust algorithm that optimizes attack's detection. In that, we tested our actor-critic algorithm using the two policy methods; however, because ***on-policy*** uses the same policy (SARSA), which it seeks to improve experience collection [49], as a result, that approach could not fully optimize our actor network for our intended solution.

Based on this insight, we conducted our experiment using the ***off-policy*** approach, which revealed that the method instead follows the tailored policy gradient, such as "$\in$-greedy" for experience replay. In doing so, our agent learned from a parameterized policy [53] that differed from the one it was supposed to follow (on-policy), thus allowing it to explore the state space, using a stochastic algorithmic policy.[5]

[5]**Stochastic**: A probability distribution that allows an agent to explore the state space without always taking the same action.

In adopting this algorithm, the agent updates the action-values $Q(S, A)$, representing the sum of rewards to $Q(s, a) = maxQ(s', a')$ relative to optimal rewards at $s_(t + 1)$ based on the maximum action-value of the next state, even though the exact action-value for the next state might not be fully known, due to the random actions [35], [6]. Nonetheless, this tabular algorithm ensured the computation and convergence of the action-values $Qt + 1(St, At) = Qt(St, At) + \alpha_t$ for each state-action pair, leading to the exploration of infinite optimal action-values as states are explored. In algorithm 3, we present the pseudo code implementation of our off-policy model as described by [54] Sutton and Barton.

Algorithm 3 Q-learning (off-policy approach) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $(epsilon)\epsilon > 0$;
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+$, $a \in \mathcal{A}(s)$, arbitrarily except that $Q(\text{terminal}, \cdot) = 0$;
for *each episode* **do**
  Initialize S;
  for *each step of episode* **do**
    Choose A from S using policy derived from Q $(\epsilon - greedy)$;
    Take action A, observe R, S';
    $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$;
    $S \leftarrow S'$;
  end
end

In addition to off-policy, we improved our approach in its exploratory state of the environment, by adding this decay function $\sigma(N) = \sigma_0 e^{kN}$. The algorithm represents the relationship between the model's input and output as the rate of decay. By defining the values of σ_0 and k, we parameterized the rate at which the decay value changes during iterations. The method helps estimate future values and rewards, based on observed trends [55].

3) SOLVING EXPLORATORY NOISE

In our algorithm, exploration noise arose from deliberate exploration strategies, environmental stochasticity, random initialization, and the use of experience replay. These features enable our agent to discover optimal policies in complex environments [56] and is also added to encourage the agent to explore new actions.

To enhance our approach during the exploratory phase of the environment, we utilize a decay function. Gradually, the method reduces the level of exploration thus making the agent more deterministic as it accumulates experiences in the environment. By incorporating a decay function within the algorithm as a policy, we helped guide the agent's actions towards increased determinism based on its experience level, using this equation:

$$\pi(a|s, N) = (1 - \sigma(N))\pi_{\text{target}}(a|s) + \frac{\sigma(N)}{|\mathcal{A}|}$$

where $\pi_{\text{target}}(a|s)$ is the target policy, $|\mathcal{A}|$ is the number of actions in the action space, and $\sigma(N) = \sigma_0 e^{kN}$ is the decay function for the exploration noise [55].

Alternatively, when incorporating the decay function into the exploration noise as a policy, our agent learned how to effectively balance both the exploration and exploitation time, thereby allowing it to achieve better performance during each episode over the course of the training [57].

In testing our model's robustness, we train and evaluate our DRL algorithm with network traffic collected from a SCADA test bed. We inspect the data for typos, and remove all duplicates and irrelevant content, which streamlines the data preparation. Next, we label the attacks to be iterated, so that when introduced to the environment, the model can learn from the patterns [35]. We set the decay parameter, and epsilon values for each step, initialize both our ''loss function and the replaymemory'' with a bach-size of 32, and then train the model for 250 episodes Fig. 10. The successful results obtained from our experiment shows that DRL has the potential to improve the detection and prevention of persistent attacks against SCADA systems.

Summary: DRL has the capability to learn patterns effectively, using *convolutional neuro network* while leveraging *Deep Q − Learning* (Fig. 4) to efficiently classify intrusions from SCADA's network traffic protocols.

B. APPLYING DRL TO SCADA

DRL is a powerful technique, ideal for monitoring and securing SCADA systems. However, from an algorithm design perspective, many challenges in method applicability must be considered when choosing the appropriate framework best suited for SCADA security; particularly when employing the Deep Q-learning algorithm and epsilon-greedy policy [29]. During the software development, training, and testing of our DRL algorithm, overcoming the following challenges were crucial to the success of our experiment:

Exploration vs Exploitation Trade-Off: This framework represents the fundamental concept in reinforcement learning, including DRL. It outlines the dilemma faced by an agent when deciding between exploring new actions or exploiting its current knowledge to maximize long-term rewards.

In the explorative state, training our actor in a complex environment with large state and action spaces is very time-consuming as the agent network samples each set of actions to obtain optimal rewards. This risky strategy can lead to negative results and ultimately affects our model's detection outcome. In contrast, our model exploitation state experienced ''local optima'' and over-fitting as it exploits prior knowledge through experience in favor of long-term rewards [58]. Though exploiting existing knowledge can lead to more efficient and effective decision-making, however, the random selection of actions with a probability of (ϵ) epsilon, can also lead to sub-optimal performance [59].

To address this trade-off, we improved the robustness of our DRL-based model by incorporating a deep Q-learning algorithm, along with an off-policy approach and a decay function. This optimization aims to overcome the limitations and enhance the overall performance of the model [14].

Summary: In developing our DQN model, the explorative and exploitative concepts within the DRL methodology present significant ***challenge*** in our design, particularly in large state and action spaces. To overcome this trade-off and achieve optimal performance, we carefully managed and balanced the technique by incorporating a ReplayMemory function with a mini-batch training; a decay function in conjunction with an off-policy approach.

C. OPTIMIZING DRL FOR SCADA ENVIRONMENT

From our investigative analysis of SCADA, particularly from the perspective of the dynamic configuration (i.e., proprietary software, hardware, and topological configuration), the information collected and expertise gained from the analysis led to the structural design of our proposed approach.

As outlined in subsection IV-C, para IV, the difference between exploration & exploitation can be balanced to help optimize DRL as they merge in unified facets, in which the model decides ***"when"*** to explore versus ***"what"*** to exploit as a strategy [48]. It is worth noting that several explorative strategies have been developed to address this trade-off, including ''Thompson sampling, Upper Confidence Bound (UCB), epsilon-greedy, and more. By incorporating a level of randomness, these strategies help balance this trade-off.

To overcome all performance challenges associated with explorative and exploitative states, we designed a DRL framework that balances explorative strategies IV-B using epsilon-greedy as presented in (Algorithm 3), which encourages the agent network to explore, trying out new actions, while actively taking advantage of experienced knowledge through exploitation. Our design incorporates regularization, in which ''decay weight'' is adjusted, with an ensemble method comprised of an actor-critic (Algorithm 2); the technique helps balance the model's efficiency and prevents generalization to new environments.

We adopted this approach because SCADA datasets are subject to data bias, due to their dynamic and non-stationary properties [21], and also because of privacy and security concerns, as their structural designs differ based on regions, software, etc.

To avoid model specialization in retrospect to DRL challenges, we create a separate data class with specific functions and methods, tailored to accommodate individual dataset's format [60]. This allows us to carefully tune and

adjust our hyperparameters [48], [59], as shown in Table 3, in order to achieve the best possible performance.

> **Summary**: To optimize DRL for SCADA network security, we addressed the challenges caused by model generalization and specialization. They are due in part to data formatting; we create a separate data class with functions and methods tailored to SCADA designs to resolve the problem.

D. MONITORING SCADA USING DRL ALGORITHM

In our investigative findings, we identified various domains in which DRL algorithm implementation has shown remarkable success. For example, in the gaming industry, DRL deployment in video game applications developed learning strategies in complex and large action spaces for reward maximization; in assembly line operations, the technique also demonstrated substantial results in process optimization and labor reduction. However, in retrospect to these domains, when considering similar methodology for cybersecurity threat detection, particularly in SCADA application, we identified several functions in DRL structural design that affect the algorithm performance [3], [61], [62].

For example, implementing a *Greedy Policy Limitation* function confines the agent to rely solely on current knowledge to exploit actions that maximize immediate rewards. The *Exploration and Exploitation* function within the algorithm designs would direct the agent to either explore the environment to gain knowledge, discover new threat patterns about protocols' vulnerabilities to maximize rewards or exploit the environment by taking uncertain and potentially risky actions that may reduce immediate rewards [63]. The cascading effects of these features can make it challenging for a DRL agent to operate effectively in stochastic, unpredictably large, and complex environments [53].

Unlike video game structural environments and assembly lines which are strictly designed with constants such as known rules and constraints, with a deterministic configuration that makes it relatively easy for the DRL technique to optimize rewards, SCADA systems, on the other hand, often operate in dynamic and unpredictable real-world environments, with complex interconnected systems comprising of water treatment plants, power grids, or manufacturing facilities. The interdependencies of nodes and network communication settings of the individual system are also proven to be very challenging for a DRL agent to optimize rewards effectively [52], [64]. As follows, we provide the explanations of the different method implementations:

- To actively monitor SCADA systems, we address the challenges stemming from the DRL framework by developing a DQN model with defined security objectives that allow the algorithm to detect network anomalies and classify them as network intrusions based on our label parameters [58].
- To *improve scalability*, we implemented a replay memory and utilized a mini-batch function that allows the agent to sample data from its prior experiences to optimize learning. We employ an off-policy along with a decay function to balance the trade-off between exploration and exploitation for rewards optimization while reducing unnecessary exploitation time [65].
- We pre-processed and formatted our data using a multi-class binary classification, which *reduces false positives*, and then trained our model on historical data containing examples of both normal network behavior and various attack scenarios.
- We continuously monitor our DQN performance by testing it on additional datasets to ensure its efficacy of real-time threat detection and response, thus simulating the *automation* of regular updates of new attack patterns and emerging security risks.

Using this approach, as the model adapts to changing network traffic, it generates appropriate actions or alerts to address security threats. This proactive monitoring technique allows our model to identify potential attacks of SCADA communication systems in real-time and take preventive measures to ensure the integrity and reliability of the grid [66].

> **Summary**: To train and deploy a DRL model to actively monitor smart infrastructure, we simulated the following: We selected two well-known datasets to simulate the data collection and formatting process; we define the security objectives using binary classification; we developed and trained a DQN model using a SCADA dataset; we then reloaded the saved model and tested it on a second set of data, thus simulating "real-time deployment." The result of this experiment shows that our model continuously analyzes the incoming data, leveraging its training to recognize patterns of normal behavior and anomalous activity.

V. IMPLEMENTATION AND EVALUATION

For our implementation, we utilize the SCADA use-case presented in Fig. 5. This use-case scenario describes a network fully responsible for remotely supervising, monitoring, and controlling critical infrastructure, such as electrical grids and stations, water treatment facilities, transportation systems, or oil refineries. In this complex operational cyber-space, the threat model involves an unauthorized attacker gaining access and compromising SCADA network traffic protocols, attempting to disrupt the system's operations.

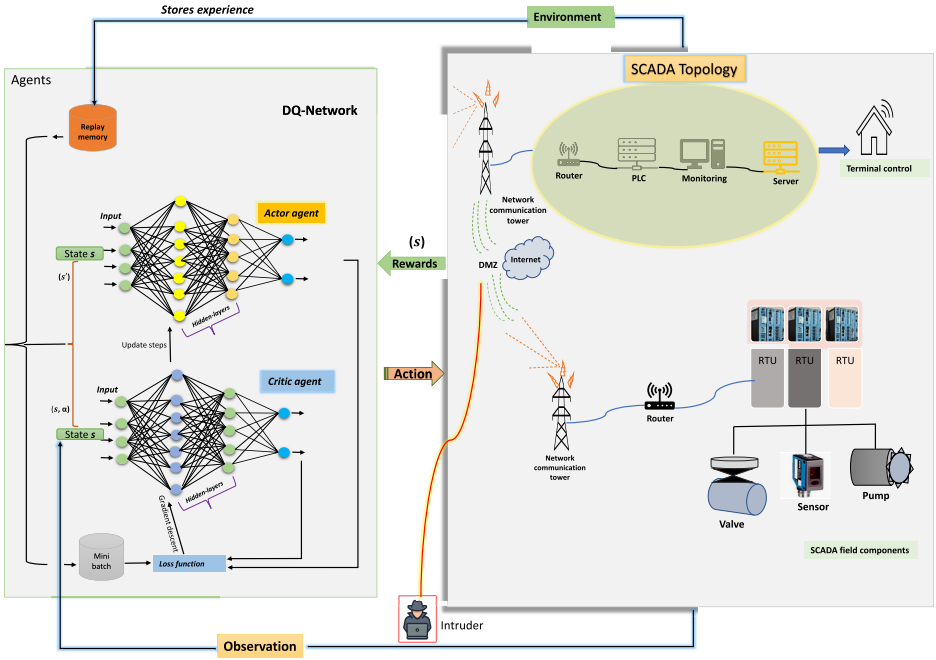

FIGURE 5. DRL deployment.

A. SYSTEM THREAT MODEL

1. ***Initial Compromise***: Through vulnerability exploitation, either by means of brute force and or social engineering, an attacker gains unauthorized access to the system's network and begins to sabotage network traffic protocols and disrupts industrial processes.
2. ***Traffic Protocol Manipulation and Alteration***: Upon establishing administrative command, the attacker cancels and encrypts control commands, injecting false sensor data to disrupt communications between IoT and IIoT devices. Such compromise may lead to catastrophic loss and physical damage to consumers.
3. ***Detection Scheme and Response***: Deploy a trained DRL as an intrusion detection response system to mitigate and assess abnormal network behaviors and to identify possible threats caused by compromised traffic protocols.

Critical infrastructure Model: Because modern critical infrastructure systems are autonomous, they use IoT and IIoT devices, and internet protocols (i.e., TCP /IP) to monitor and control critical functions in water treatment plants, power grids, transportation networks, etc. Although these devices provide cutting-edge multi-functionality to data collection and automation processes, their designs pose the greatest security vulnerability to the system. So, deploying a trained DRL as an NIDS [35], [67], [68] can authenticate communication and filter TCP/IP traffic of attached nodes in a synchronous and asynchronous [66] manner to help detect lurking threats.

Applying DRL to Critical Infrastructure: In this section, we describe our **Deep QNetwork** approach to ensure the security of SCADA's network-attached nodes (Fig. 2). These interconnected IoT and IIoT devices lack the necessary security features, which create a larger attack surface within the system, thus making it more susceptible to cyber threats. However, applying DRL as an intrusion detection "avant-guard" addresses this challenge by using deep q-learning [58] algorithms to analyze network traffic patterns, identify anomalous behaviors for potential intrusions, while reducing downtime using the deployment approach in (Fig. 5) and the detection strategy in (Algorithm 4).

B. EXPERIMENTAL ENVIRONMENT

All of our experiments were conducted on a computer with the following specifications:

- Intel(R) Core^{TM} I7-1700 CPU @ 2.90GHz; 16GB RAM
- Intel(R) UHD Graphics 630 GPU
- And a dedicated AMD Radeon RX 640 GPU
- Windows 10 Pro operating system
- Python 3.9.12, gym 0.21.0, Tensorflow 2.9.1
- Keras 2.9.0, SciKit-Learn 1.0.2, matplotlib, seaborn

C. EVALUATION

In this section, we briefly outline key tenants of our algorithm research objective and provide a brief description of the datasets used in our proposed approach.

Our experimental objective aimed to develop and evaluate a DRL framework to assess its robustness and effectiveness and to determine whether the algorithm can be successfully leveraged to detect cybersecurity attacks in *large and complex environments* [29]; specifically, to investigate the impact of using DRL's technique on new and unseen data.

Algorithm 4 Deploying DRL as an IDS

_**Use-case**: we consider a SCADA (Fig. 2) network fully responsible for remotely supervising, monitoring and controlling critical infrastructure, such as electrical grids and stations, water treatment facilities, transportation system, or oil refineries;
_**Threat**: unauthorized access and compromised network traffic protocols;
1. Initial Compromise:
Attacker exploits vulnerabilities through brute force or social engineering;
Gain unauthorized access to the network;
Sabotage network traffic protocols and disrupt industrial processes;
2. Network Protocol Manipulation and Alteration:
The attacker establishes administrative command;
Cancel and encrypt control commands;
Inject false sensor data;
Disrupt communication between IoT and IIoT devices;
Potential catastrophic loss and physical damages;
3. Detection Scheme and Response:
Deploy a trained DRL as an intrusion detection response system;
while *not done* **do**
  Observe the network state;
  Select an action using an epsilon-greedy policy;
  Perform the selected action in the network;
  Observe the next network state, reward, and done flag;
  Store the experience in the replay memory;
  Sample a minibatch of experiences from the replay memory;
  Update the Q-network using the actor-critic algorithm;
  Update the actor-network using TD error;
  Update the critic network using the TD target;
  Periodically update the target networks;
  Check if the episode is done or the maximum number of steps is reached;
  if *yes* **then**
    Go to the next episode;
  end
  Evaluate the trained DRL agent's performance;
  Publish attacks: Types, Names and graph results
end

To validate these objectives, we developed a DNQ model consisting of two hidden layers with an actor-critic algorithm and a replay-memory function.

As illustrated in algorithm 5, to balance our model explorative and exploitative behavior in large, complex state and action spaces, we implemented an off-policy along with a decay function that allows the agent to sample mini-

Algorithm 5 DQN cross-validation

Input: Dataset D
Discount factor γ
Exploration rate ϵ
Minimum exploration rate ϵ_{min}
Exploration rate decay factor ϵ_{decay}
Replay memory size N
Batch size B
Number of episodes E
Target network update frequency C
Learning rate α
Hidden layer size H
Number of hidden layers L
Output:
Trained DQN model
Procedure:
Initialize main Q-network Q with random weights
Initialize target Q-network Q' with same weights as Q
Initialize the replay memory R with size N
for *episode in range(E)* **do**
  Initialize state s
  Initialize *done* to False **while** *not done* **do**
    if *With probability* ϵ, *select a random action* a **then**
    end
    else
      Select $a = \arg\max_a Q(s, a)$
    end
    Execute action a and observe reward r and next state s'
    Add transition $(s, a, r, s', done)$ to replay memory R
    Sample random minibatch of B transitions $(s_i, a_i, r_i, s'_i, done_i)$ from R
    foreach *transition in the minibatch* **do**
      if $done_i$, *set target* $= r_i$ **then**
      end
      else
        $target = r_i + \gamma \max_{a'} Q'(s'_i, a')$
      end
      Calculate loss $L = (Q(s_i, a_i) - target)^2$
      Update weights of Q using gradient descent with learning rate α to minimize L
    end
    for *Every C steps* **do**
      Copy weights of Q to Q'
    end
    ϵ-greedy exploration rate is decayed linearly from ϵ to ϵ_{min} over time.
  end
end
Return trained DQN model

bach from replay-memory containing past experiences, while reducing exploration rate and improving exploitation of prior knowledge [67].

We test the model on two publically available IIoT SCADA datasets, wustl-iiot-2018[6] and the wustl-iiot-2021[7] dataset to gain insights into the technique's performance in these aspects.

1) DATASETS

The *WUSTL-IIoT-2018* dataset is a dataset focused on Industrial Internet of Things (IIoT) systems, capturing features related to network traffic, device interactions, and protocols used. It contains a significant number of

[6]wustl-iiot-2018 https://www.cse.wustl.edu/ jain/iiot/index.html
[7]wustl-iiot-2021 https://www.cse.wustl.edu/ jain/iiot2/index.html

attacks, including DDoS, injection, and command execution attacks [25].

The *WUSTL-IIoT-2021* dataset is a recent dataset focusing on IIoT devices and systems. It includes updated information on IIoT behavior, network interactions, and potential cybersecurity threats. The dataset contains a variety of features specific to IIoT, with a substantial number of attacks, including unauthorized access, data exfiltration, and device manipulation [69].

Datasets Characteristics: During our preliminary analysis and pre-processing of the datasets (Fig. 6), we identified four types of attacks commonly simulated to test SCADA's network security defenses: ***Port Scanner***; ***Address Scan Attack***; ***Device Identification Attack*** and ***Exploit*** [16], [70], [71], [72]. These ''scans'' represent a precursor or the initial stage for more severe attacks. These frontal assaults help collect sensitive data and subsequently expose an organization's vulnerability [14]. Their success can lead to data breaches and possibly affect control decisions, causing equipment damage or triggering unsafe use of infrastructure assets. Based on those threat patterns, we mapped and labeled those attacks as our target detections (Algorithm 6).

Algorithm 6 Label mapping

```
Input: Initialize dataset
Process: Initialize an empty dictionary to store labels
Output: DICTIONARY = Attack_labels & Attack_map

for each label in mapping dict do
    if the label is "normal" then
        add to dictionary with a value of 0
        if the label is "Probe" then
            add to dictionary with a value of 1
            if the label is "R2L" then
                add to dictionary with a value of 2
                if the label is "DoS" then
                    add to dictionary with a value of 3
                end
            end
        end
    end

    labels
end
```

Algorithm 7 Reading Labels

```
Input: Initialize dictionary
Output: Attack_labels

forall correct label do
    labels ← Agent (by)
    for each iteration do
        s_t , a_t = []
        for attack label in attack_map do
            n=length(attack_map)
        end
        return labels

        labels
    end
end
```

In meeting our set goals, we designed a double-layer Deep QNetwork algorithm [58], we pre-processed two unique datasets, and extracted features of importance for detecting cyber anomalies. We then label each selected features and store them in a dictionary to fit our model's input as shown in (Algorithm 6). Upon initialization, the actor-network loops through the dictionary and matches the corresponding labels with each attack category as presented in (Algorithm 7).

Next, we fine-tuned the algorithm using the values in (Table 3). These methods allow our actor to recognize patterns from normal and anomalous network traffic behavior.

2) PERFORMANCE METRICS

For validation, we used the standard measurement metrics to assess the model's predicting ''Accuracy, F-1 score, False Positive, and False Negative rates.'' We also use the formulas TP, TN, FP, and FN to calculate our model's performance.

Accuracy: It measures the ratio of correctly predicted labels using this equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision: Measures the ratio of the accuracy of positive predictions:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall: Quantifies the number of correct positive predictions made out of all positive predictions that could have been made by the classifier:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F1-score: This metric measures the harmonic mean of precision and recall using this formula:

$$f1 - score = \frac{2 \cdot P \cdot R}{P + R} \tag{4}$$

Lastly, we trained and evaluated our model using the selected datasets: WUSTL-IIoT-2018 [25] and WUSTL-IIoT-2021 [26], comprised of network traffic protocols. After successfully training the model, our DRL learns to classify threats in real-time and provides detection and response [27]. In retrospect to validation V-C2, we loaded the saved model with our test data, comprised of multiple binary classification inputs. Using both methods from algorithms 6 and 7, the agent-network read the selected **labeled attacks** and predicts our targeted attack labels.

VI. RESULTS

The results presented from our evaluation is an illustration of the training characteristic of our DRL framework. Table 4 shows the evaluation results achieved on the two datasets considered in this work: WUSTL-IIoT-2018 and WUSTL-IIoT-2021. Likewise, in Table 5, we present the performance metrics of our model, which were compared with two other research works that also utilized a DRL technique. (i.e., [15] and [44] on the same dataset. We considered only the

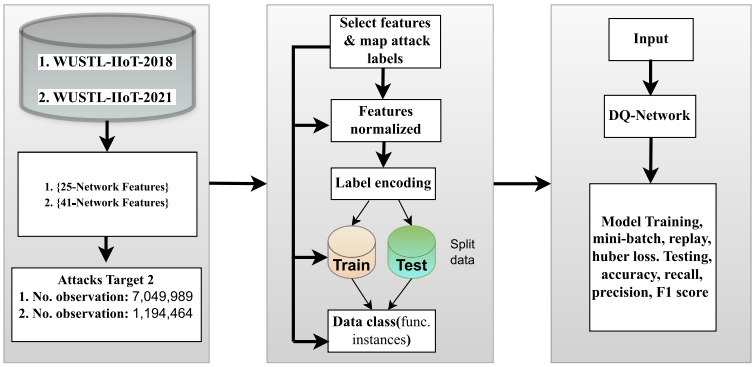

FIGURE 6. Data processing.

TABLE 2. Common attacks found on each dataset.

Attack type	Description	Dataset	No. Samples
Command Injection	Manipulates a system or application to execute unauthorized commands.	WUSTL-IIoT-2018 WUSTL-IIoT-2021	N/A 370284
R2L	Targets services such as SSH, Telnet, or FTP, allowing remote login.	WUSTL-IIoT-2018 WUSTL-IIoT-2021	52785 470
U2R	User gains access through vulnerability and later access to resources.	WUSTL-IIoT-2018 WUSTL-IIoT-2021	3470359 470
Probe	A type of scan that identifies open ports and network vulnerability.	WUSTL-IIoT-2018 WUSTL-IIoT-2021	2111 550
Backdoor	Used as a mean of maintaining unauthorized access to a system.	WUSTL-IIoT-2018 WUSTL-IIoT-2021	704 298616
DOS	This attack floods the target with traffic that triggers a crash.	WUSTL-IIoT-2018 WUSTL-IIoT-2021	4272056 1074779
Reconnaissance	Unauthorized attempt to gather information about a target network.	WUSTL-IIoT-2018 WUSTL-IIoT-2021	796137 112996
Normal	Represent routine network traffic w/no anomalies.	WUSTL-IIoT-2018 WUSTL-IIoT-2021	6622055 1107448

TABLE 3. Environment hyper-parameter.

Hyperparameter	Value
γ	0.001
ϵ	1
$\epsilon_{\min}$	0.01
r_d	0.995
K	100
Minibatch_size	128
Learning_rate	0.01
Optimizer	Adam optimizer

research works by Tharewal et al. [15] and Wang et al. [44], because they were the only two presenting a complete evaluation of their solution with the appropriate evaluation metrics. In this experiment, we trained our model using an actor-critic algorithm that learns to make decisions. From the conceptual environment as demonstrated in Figure 10, the critic evaluates the actor's performance by providing feedback; the actor then updates the policy distribution based on the information received, thus helping to improve the model's learning capabilities.

Our DQN is trained with optimized parameters as recorded in table 3 and algorithms 6 and 7 to accurately classify both normal and anomalous behavior from the input data. In this synchronous process, we set our critic parameter to equal those of our actor-network (i.e., identically mirroring γ, ϵ, $\epsilon_{\min}$, and r_d) as target value.

We adjust our DQN minibatch_size after each training phase to minimize the loss function, aiming to improve the alignment between the predicted output of the model and the desired output, as labeled in algorithm 6. Overall the training results demonstrate that our model successfully captures and accurately interprets the assigned attack labels, achieving convergence with an improved loss of less than 0.4 over 250 training episodes (Fig. 10), while accumulating the highest number of rewards (Fig. 7).

During each learning step, our model updates both the Actor-Critic's parameter using ''policy gradients'' and ''advantage value'' by minimizing the mean squared error using the Bellman's equation as detailed and implemented in IV-A1.

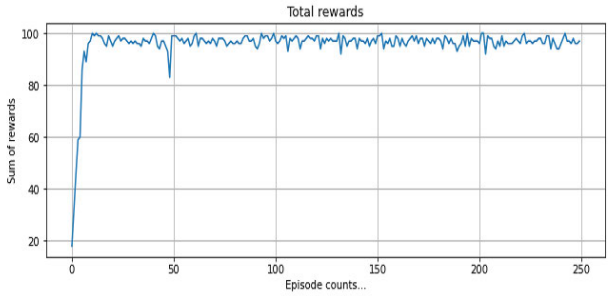

FIGURE 7. DQN agent cumulative rewards.

After convergence, the model total loss per episode drops from 0.5 and remains stable at under 0.3 for the duration of the training as presented in Figure 8.

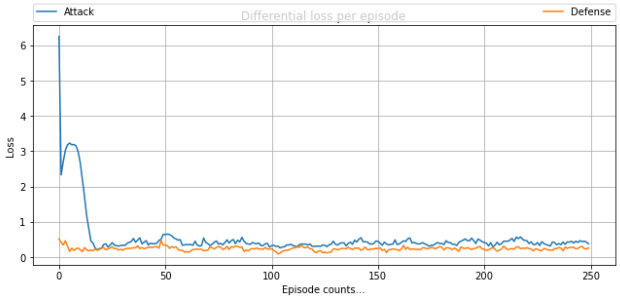

FIGURE 8. Converging loss.

Figure 9 illustrates our DQN successful training results, depicting the accurate labeling and classification of network attacks.

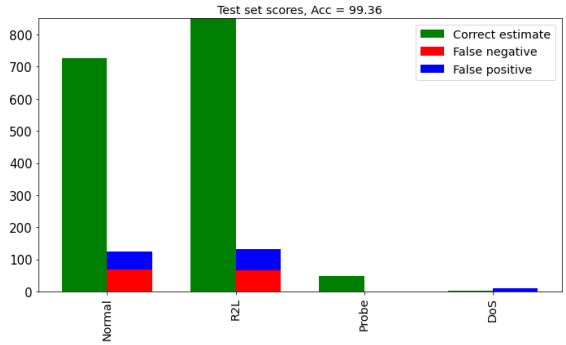

FIGURE 9. Training results.

VII. THREATS TO THE VALIDITY

In this section, we explore the conceivable threats to the validity of our study and discuss criteria that could potentially impact the integrity and applicability of our findings. We acknowledge and address the following threats:

SCADA technology plays a significant role in the autonomous operations of smart infrastructure. From a functionality and security perspective, the diverse structural designs of these systems, change the dynamics of data collection, in terms of hardware, network protocols, and data uniformity.

TABLE 4. Model performance results.

	Proposed DQN model			
Datasets	Accuracy	Precision	F1-Score	Recall
WUSTL-IIoT-2018	99.36%	99.40%	99.36%	99.38%
WUSTL-IIoT-2021	98.62%	99.23%	98.90%	98.62%

TABLE 5. Comparison table WUSTL-IIoT-2018.

	Results			
Related technique	Accuracy	Precision	F1-Score	Recall
Tharewal et al. [16]	99.09 %	97.12%	95.05%	96.45%
Wang et al. [46]	99.04%	98.48%	96.81%	97.65%
Our approach *	**99.36%**	**99.40%**	**99.36%**	**99.38%**

A. SAMPLING BIAS AND CULTURAL BIAS

a.) Our selected WUSTL-IIOT datasets may not be representative of all categories of SCADA communication infrastructure; this is due in part to structural designs (e.g., topology, hardware, and software) which may produce significant variance in data samples

B. NON-STATIONARY DATA AND LIMITED GENERALIZABILITY

b.) SCADA communications network could vary by states, regions, and country-to-country, such that it changes over time. In that, when considering the WUSTL-IIOT dataset, there is a likelihood that these changes may not be reflected The presence of these variances could significantly affect the performance of the model on unseen data; meaning, If the sample is not representative of the target population, the generalizability of the results may be limited, and it may be inappropriate to extrapolate the findings to broader populations or contexts

C. MITIGATION STRATEGIES:

To alleviate these threats, we employed rigorous methodological procedures, including:

- randomized control trial selection selection
- we carefully conducted sample and robust statistical analyses

Nevertheless, it is important to acknowledge that some degree of uncertainty may still exist. Additionally, we have discussed and addressed several of these concerns in the limitations' section.

D. LIMITATIONS

Despite our proposed DRL promising outcomes and effective results in identifying patterns and detecting cyber-attacks in

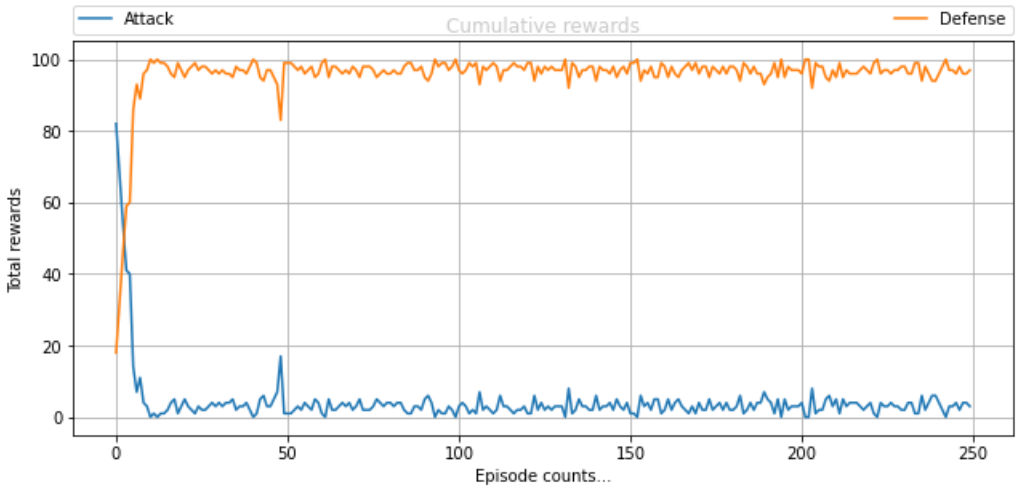

FIGURE 10. Cumulative training results.

SCADA infrastructure, we have identified three limitations that should be acknowledged. We detailed these techniques and outlined the measures taken to mitigate these constraints:

1. **Limited Dataset**: one of the limitations of our study is the availability of a limited dataset specific to SCADA infrastructure cyber-attacks. To mitigate this limitation, we utilized the WUSTL-IIoT-2018 and WUSTL-IIoT-2021 datasets, which include 25 networking features representing both benign and attack traffic. While these datasets provide valuable insights, they may not cover the entire spectrum of potential cyber-attacks. In particular, the selected datasets WUSTL-IIOT-2018 and the WUSTL-IIOT-2021 may not represent the broader population of SCADA infrastructure, which could lead to sampling bias. Such inconsistency may affect a model's accuracy and cause poor performance on unseen data [73]. To address this, we ensured rigorous preprocessing to maximize the utility of the available data.
2. **Generalization**: our proposed model may face challenges in detecting novel or previously unseen cyber-attack types that are not present in the training dataset. Although the Deep Q-learning algorithm and the Q-network as function approximators are designed to capture complex patterns, the model's performance may be affected when encountering unknown attack patterns. In other words, our DRL model trained on two specific datasets, as explained in "non-stationary data" may not generalize well to other datasets or when applied to different scenarios. This could limit the applicability of the model and may require substantial retraining when attempting to use the saved model in a new context. To address this, we emphasize the need for continuous monitoring and updating of the model with new data to adapt to emerging threats and to ensure ongoing effectiveness in real-world scenarios.
3. **Scalability**: the proposed framework utilizes a fully connected neural network architecture with two hidden layers consisting of 64 and 32 fully connected neurons, respectively. While this architecture has demonstrated satisfactory performance in our experiments, scalability may become a concern when dealing with larger SCADA systems or more complex environments. To address this, future research should focus on investigating advanced network architectures and exploring techniques such as convolutional or recurrent neural networks to enhance the scalability of the model without compromising its performance.

E. FUTURE RESEARCH DIRECTIONS

Subsequently, we present four future research directions that arise from our study:

1. **Focus on Adversarial Attacks**: as the sophistication of cyber-attacks continues to evolve, future research should explore the vulnerability of the proposed DRL framework to adversarial attacks. Investigating methods to enhance the model's robustness against adversarial manipulations and exploring adversarial training techniques could significantly improve its real-world applicability and effectiveness.
2. **Use Real-time Data**: real-time detection and response are crucial in protecting SCADA infrastructure. Future research should focus on reducing the inference time of the proposed model to ensure timely identification and prevention of cyber-attacks. Techniques such as model compression, quantization, and hardware acceleration can be explored to achieve low-latency and efficient deployment of the framework in real-world scenarios.
3. **Apply Transfer Learning and Data Augmentation**: to address the limited dataset issue, future research can explore transfer learning techniques to leverage pre-trained models on larger and more diverse datasets in

related domains. Additionally, data augmentation techniques can be employed to generate synthetic samples that represent a wider range of attack scenarios, further enhancing the model's generalization capabilities.

4. **Improve Explainability and Interpretability**: DL models, including the proposed DRL framework, often lack interpretability, making it challenging to understand the reasoning behind their decisions. Because DRL models are difficult to interpret, it may be challenging to fix errors, diagnose, or even improve a model's overall performance, all due in part to structural complexity. However, through proper design choices and documentation, including the model's assumptions and hyperparameters implementation, researchers can have a clear understanding of a model's behavior and even a broader insight into its decision-making process. Future research should focus on developing techniques to explain and interpret the model's behavior, providing insights into the features and patterns that contribute to cyber-attack detection. Doing so would not only help enhance the model's trustworthiness but also enable cybersecurity analysts to gain valuable insights for further investigations.

By addressing these limitations and exploring future research directions, we can continue to advance the capabilities of DRL in detecting and preventing cyber-attacks in SCADA infrastructure, ultimately enhancing critical infrastructure security and protecting against emerging threats.

VIII. CONCLUSION

In this work, we aimed to address the challenges of the DRL framework and its applicability in cybersecurity to advance the development of effective solutions in this field. To achieve this, we design a double-layered DQN model that utilizes a Deep Q-Learning algorithm that enhances learning abilities in complex environments while adapting to new and unseen data. To balance our model's explorative and exploitative behavior in complex state and action spaces, we implemented an off-policy with a decay function, allowing the agent to sample a mini-batch from replay memory containing past experiences. To evaluate the model's efficacy, we explored the security challenge of critical infrastructure against continuous cyber-security attacks. We conducted an investigative threat analysis of SCADA systems to assess their network dependency and potential vulnerabilities. Based on our findings, we selected the SCADA domain as our target objective and evaluated our DQN using two publicly available datasets from the SCADA testbed: WUSTL-IIoT-2018 and WUSTL-IIoT-2021. The results from our trained model show that our DQN can learn to classify threats at 99% accuracy and provide detection and response in real-time. In retrospect, it is important to note that detecting certain types of attacks, such as spoofed traffic, destination packets, and total byte attacks, can be challenging and may require collaborating with other techniques and tools. Overall, the use of DRL for detecting cybersecurity attacks in SCADA infrastructure is a promising approach with the potential to significantly improve the security of these systems.

REFERENCES

[1] D. P. Joseph and J. Norman, "An analysis of digital forensics in cyber security," in *Proc. 1st Int. Conf. Artif. Intell. Cogn. Comput. (AICC)*. Singapore: Springer, 2018, pp. 701–708.

[2] R. Diao, Z. Wang, D. Shi, Q. Chang, J. Duan, and X. Zhang, "Autonomous voltage control for grid operation using deep reinforcement learning," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Aug. 2019, pp. 1–5.

[3] X. Liu, W. Yu, F. Liang, D. Griffith, and N. Golmie, "On deep reinforcement learning security for industrial Internet of Things," *Comput. Commun.*, vol. 168, pp. 20–32, Feb. 2021.

[4] J. Duan, D. Shi, R. Diao, H. Li, Z. Wang, B. Zhang, D. Bian, and Z. Yi, "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 814–817, Jan. 2020.

[5] B. B. Zad, J.-F. Toubeau, O. Acclassato, O. Durieux, and F. Vallée, "An innovative centralized voltage control method for MV distribution systems based on deep reinforcement learning: Application on a real test case in Benin," in *Proc. CIRED 26th Int. Conf. Exhib. Electr. Distrib.* Edison, NJ, USA: IET, 2021, pp. 1577–1581.

[6] Y. Liu, H. Xu, D. Liu, and L. Wang, "A digital twin-based sim-to-real transfer for deep reinforcement learning-enabled industrial robot grasping," *Robot. Comput.-Integr. Manuf.*, vol. 78, Aug. 2022, Art. no. 102365.

[7] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, Sep. 2013.

[8] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–17, 2021.

[9] OpenAI. (2020). *Robogym*. [Online]. Available: https://github.com/openai/robogym

[10] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, Sep. 2022.

[11] S. Thrun and L. Pratt, *Learning to Learn: Introduction and Overview*. Boston, MA, USA: Springer, 1998, pp. 3–17, doi: 10.1007/978-1-4615-5529-2_1.

[12] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *J. Artif. Intell. Res.*, vol. 47, pp. 253–279, Jun. 2013.

[13] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.

[14] B. Ning and L. Xiao, "Defense against advanced persistent threats in smart grids: A reinforcement learning approach," in *Proc. 40th Chin. Control Conf. (CCC)*, Jul. 2021, pp. 8598–8603.

[15] S. Tharewal, M. W. Ashfaque, S. S. Banu, P. Uma, S. M. Hassen, and M. Shabaz, "Intrusion detection system for industrial Internet of Things based on deep reinforcement learning," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–8, Mar. 2022.

[16] O. Yousuf and R. N. Mir, "DDoS attack detection in Internet of Things using recurrent neural network," *Comput. Electr. Eng.*, vol. 101, Jul. 2022, Art. no. 108034. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S004579062200297X

[17] M. Landen, K. Chung, M. Ike, S. Mackay, J.-P. Watson, and W. Lee, "DRAGON: Deep reinforcement learning for autonomous grid operation and attack detection," in *Proc. 38th Annu. Comput. Secur. Appl. Conf.*, 2022, pp. 13–27.

[18] F. Wei, Z. Wan, and H. He, "Cyber-attack recovery strategy for smart grid based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2476–2486, May 2020.

[19] D. Silver, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[20] M. Alauthman, N. Aslam, M. Al-kasassbeh, S. Khan, A. Al-Qerem, and K.-K. Raymond Choo, "An efficient reinforcement learning-based botnet detection approach," *J. Netw. Comput. Appl.*, vol. 150, Jan. 2020, Art. no. 102479.

[21] Z. Wang, H. He, Z. Wan, and Y. Sun, "Coordinated topology attacks in smart grid using deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1407–1415, Feb. 2021.
[22] M. H. Ling, K.-L.-A. Yau, J. Qadir, G. S. Poh, and Q. Ni, "Application of reinforcement learning for security enhancement in cognitive radio networks," *Appl. Soft Comput.*, vol. 37, pp. 809–829, Dec. 2015.
[23] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.
[24] A. Kathirgamanathan, E. Mangina, and D. P. Finn, "Development of a soft actor critic deep reinforcement learning approach for harnessing energy flexibility in a large office building," *Energy AI*, vol. 5, Sep. 2021, Art. no. 100101. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666546821000537
[25] M. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, and M. Samaka, "SCADA system testbed for cybersecurity research using machine learning approach," *Future Internet*, vol. 10, no. 8, p. 76, Aug. 2018.
[26] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.
[27] J. Li, T. Yu, H. Zhu, F. Li, D. Lin, and Z. Li, "Multi-agent deep reinforcement learning for sectional AGC dispatch," *IEEE Access*, vol. 8, pp. 158067–158081, 2020.
[28] M. Botvinick, J. X. Wang, W. Dabney, K. J. Miller, and Z. Kurth-Nelson, "Deep reinforcement learning and its neuroscientific implications," *Neuron*, vol. 107, no. 4, pp. 603–616, Aug. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0896627320304682
[29] M. Moradi, Y. Weng, and Y.-C. Lai, "Defending smart electrical power grids against cyberattacks with deep q-learning," *PRX Energy*, vol. 1, no. 3, Nov. 2022, Art. no. 033005.
[30] K. Arulkumaran, M. Peter Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," 2017, *arXiv:1708.05866*.
[31] D.-J. Shin and J.-J. Kim, "Deep reinforcement learning-based network routing technology for data recovery in exa-scale cloud distributed clustering systems," *Appl. Sci.*, vol. 11, no. 18, p. 8727, Sep. 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/18/8727
[32] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.
[33] E. Gibney, "Deepmind algorithm beats people at classic video games," *Nature*, vol. 518, no. 7540, pp. 465–466, 2015.
[34] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 2, pp. 257–265, Jun. 2018.
[35] Z. Gao, Y. Gao, Y. Hu, Z. Jiang, and J. Su, "Application of deep Q-network in portfolio management," in *Proc. 5th IEEE Int. Conf. Big Data Analytics (ICBDA)*, May 2020, pp. 268–275.
[36] B. Berry, "Do you know these key SCADA concepts SCADA tutorial: A quick, easy, comprehensive guide (white paper)," DPS Telecom, Fresno, CA, USA, Tech. Rep., 2011.
[37] J. Clifton and E. Laber, "Q-learning: Theory and applications," *Annu. Rev. Statist. Appl.*, vol. 7, pp. 279–301, Mar. 2020.
[38] S. R. Chhetri, S. Faezi, N. Rashid, and M. A. Al Faruque, "Manufacturing supply chain and product lifecycle security in the era of industry 4.0," *J. Hardw. Syst. Secur.*, vol. 2, no. 1, pp. 51–68, Mar. 2018.
[39] R. Antrobus, B. Green, S. Frey, and A. Rashid, "The forgotten I in IIoT: A vulnerability scanner for industrial Internet of Things," Living Internet Things (IoT), London, U.K., Tech. Rep., May 2019.
[40] X. Liu, C. Qian, W. G. Hatcher, H. Xu, W. Liao, and W. Yu, "Secure Internet of Things (IoT)-based smart-world critical infrastructures: Survey, case study and research opportunities," *IEEE Access*, vol. 7, pp. 79523–79544, 2019.
[41] G. Stoneburner, A. Goguen, and A. Feringa, "Risk management guide for information technology systems," *Nist Special Publication*, vol. 800, no. 30, pp. 30–800, 2002.
[42] R. Huang, Y. Li, and X. Wang, "Attention-aware deep reinforcement learning for detecting false data injection attacks in smart grids," *Int. J. Electr. Power Energy Syst.*, vol. 147, May 2023, Art. no. 108815. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061522008110
[43] D. Wu, A. Ren, W. Zhang, F. Fan, P. Liu, X. Fu, and J. Terpenny, "Cybersecurity for digital manufacturing," *J. Manuf. Syst.*, vol. 48, pp. 3–12, Jul. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612518300396
[44] W. Wang, J. Guo, Z. Wang, H. Wang, J. Cheng, C. Wang, M. Yuan, J. Kurths, X. Luo, and Y. Gao, "Abnormal flow detection in industrial control network based on deep reinforcement learning," *Appl. Math. Comput.*, vol. 409, Nov. 2021, Art. no. 126379.
[45] P. ÄOisar and S. M. ÄOisar, "General vulnerability aspects of Internet of Things," in *Proc. 16th IEEE Int. Symp. Comput. Intell. Informat. (CINTI)*, Nov. 2015, pp. 117–121.
[46] D. Torre, F. Mesadieu, and A. Chennamaneni, "Deep learning techniques to detect cybersecurity attacks: A systematic mapping study," *Empirical Softw. Eng.*, vol. 28, no. 3, p. 44, May 2023.
[47] R. Antrobus, B. Green, S. Frey, and A. Rashid, "The forgotten I in IIoT: A vulnerability scanner for industrial Internet of Things," in *Proc. Living Internet Things (IoT)*, 2019, pp. 1–8.
[48] B. Jang, M. Kim, G. Harerimana, and J. W. Kim, "Q-learning algorithms: A comprehensive classification and applications," *IEEE Access*, vol. 7, pp. 133653–133667, 2019.
[49] J. Suh and T. Tanaka, "Sarsa(0) reinforcement learning over fully homomorphic encryption," in *Proc. SICE Int. Symp. Control Syst. (SICE ISCS)*, Mar. 2021, pp. 1–7.
[50] I. C. Dolcetta and H. Ishii, "Approximate solutions of the Bellman equation of deterministic control theory," *Appl. Math. Optim.*, vol. 11, no. 1, pp. 161–181, Feb. 1984.
[51] R. W. Beard, G. N. Saridis, and J. T. Wen, "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, vol. 33, no. 12, pp. 2159–2177, Dec. 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109897001283
[52] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Netw.*, vol. 21, no. 4, pp. 682–697, May 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608008000701
[53] A. Akagic and I. Džafic, "Deep reinforcement learning in smart grid: Progress and prospects," in *Proc. XXVIII Int. Conf. Inf., Commun. Autom. Technol. (ICAT)*, Jun. 2022, pp. 1–6.
[54] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[55] P. E. Protter, *Stochastic Differential Equations*. Berlin, Germany: Springer, 2005.
[56] S. Mohamed, J. Dong, A. R. Junejo, and D. C. Zuo, "Model-based: End-to-end molecular communication system through deep reinforcement learning auto encoder," *IEEE Access*, vol. 7, pp. 70279–70286, 2019.
[57] B. Øksendal, *Stochastic Differential Equations*. Berlin, Germany: Springer, 2003, pp. 65–84, doi: 10.1007/978-3-642-14394-6_5.
[58] G. Fragkos, J. Johnson, and E. E. Tsiropoulou, "Dynamic role-based access control policy for smart grid applications: An offline deep reinforcement learning approach," *IEEE Trans. Hum.-Mach. Syst.*, vol. 52, no. 4, pp. 761–773, Aug. 2022.
[59] J. Khoury and M. Nassar, "A hybrid game theory and reinforcement learning approach for cyber-physical systems security," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2020, pp. 1–9.
[60] Z. Wang and X. Chu, "Operating condition identification of complete wind turbine using DBN and improved DDPG-SOM," in *Proc. IEEE 11th Data Driven Control Learn. Syst. Conf. (DDCLS)*, Aug. 2022, pp. 94–101.
[61] D. Zhao, D. Liu, F. L. Lewis, J. C. Principe, and S. Squartini, "Special issue on deep reinforcement learning and adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2038–2041, Jun. 2018.
[62] J. Xu, H. Wang, J. Rao, and J. Wang, "Zone scheduling optimization of pumps in water distribution networks with deep reinforcement learning and knowledge-assisted learning," *Soft Comput.*, vol. 25, no. 23, pp. 14757–14767, Dec. 2021.
[63] J. Stranahan, T. Soni, and V. Heydari, "Supervisory control and data acquisition testbed vulnerabilities and attacks," in *Proc. SoutheastCon*, Apr. 2019, pp. 1–5.
[64] D. Hamouda, M. A. Ferrag, N. Benhamida, and H. Seridi, "Intrusion detection systems for industrial Internet of Things: A survey," in *Proc. Int. Conf. Theor. Applicative Aspects Comput. Sci. (ICTAACS)*, Dec. 2021, pp. 1–8.
[65] S. Wang, R. Diao, C. Xu, D. Shi, and Z. Wang, "On multi-event co-calibration of dynamic model parameters using soft actor-critic," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 521–524, Jan. 2021.
[66] X. Zhao, S. Ding, Y. An, and W. Jia, "Applications of asynchronous deep reinforcement learning based on dynamic updating weights," *Appl. Intell.*, vol. 49, pp. 581–591, 2019.

[67] S. Baek, J. Kim, H. Yu, G. Yang, I. Sohn, Y. Cho, and C. Park, "Intelligent feature selection for ECG-based personal authentication using deep reinforcement learning," *Sensors*, vol. 23, no. 3, p. 1230, Jan. 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/3/1230
[68] M. Zheng, I. Zada, S. Shahzad, J. Iqbal, M. Shafiq, M. Zeeshan, and A. Ali, "Key performance indicators for the integration of the service-oriented architecture and scrum process model for IoT," *Scientific Program.*, vol. 2021, pp. 1–11, Feb. 2021.
[69] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of industrial Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6822–6834, Aug. 2019.
[70] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for SCADA networks," in *Proc. SCADA Security Sci. Symp.*, vol. 46, 2007, pp. 1–12.
[71] W. Alsabbagh, S. Amogbonjaye, D. Urrego, and P. Langendörfer, "A stealthy false command injection attack on modbus based SCADA systems," in *Proc. IEEE 20th Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2023, pp. 1–9.
[72] I. Ortega-Fernandez and F. Liberati, "A review of denial of service attack and mitigation in the smart grid using reinforcement learning," *Energies*, vol. 16, no. 2, p. 635, Jan. 2023. [Online]. Available: https://www.mdpi.com/1996-1073/16/2/635
[73] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *J. Mach. Learn. Res.*, vol. 11, pp. 2079–2107, Jul. 2010.

FRANTZY MESADIEU received the Associate of Art degree from Central Texas College, USA, in 2018, and the B.S. degree in cybersecurity and the M.S. degree in computer information system from Texas A&M University Central Texas, USA, in 2020 and 2023, respectively.

From 2021 to 2022, he was a Graduate Research Assistant with the Centre for Cybersecurity Innovation, Texas A&M University Central Texas. He is currently a Research Associate and an Adjunct Faculty Member with the Subhani Department of Computer Information Systems, Texas A&M University Central Texas. His research interests include cybersecurity, more specifically in artificial intelligence, experimenting with reinforcement learning, and deep reinforcement learning framework.

DAMIANO TORRE (Member, IEEE) received the B.Sc. degree from the University of Bari, Italy, in 2009, the M.Sc. degree from the University of Castilla-La Mancha, Spain, in 2011, and the Ph.D. degree from Carleton University, Canada, in 2018.

From 2020 to 2023, he was an Associate Research Scientist with the Centre for Cybersecurity Innovation, Texas A&M University Central Texas, USA. Prior to coming to the USA, he was a Research Associate with the University of Luxembourg, from 2018 to 2021. His research interests include computer science, and more specifically on software engineering, cybersecurity, artificial intelligence, model-driven engineering, and empirical software engineering. He regularly serves on the organizing/program committees of ISSRE and QRS; and satellite events of ESEM, ICSE, and ASE.

ANITHA CHENNAMANENI (Member, IEEE) received the Ph.D. degree in business administration (with a major in information systems and a minor in computer science) from The University of Texas at Arlington. She is currently the Chair and a Professor with the Subhani Department of Computer Information Systems, Texas A&M University Central Texas, and the Director of the Centre for Cybersecurity Innovation. Her work has been published in many peer-reviewed journals and conferences. Her research interests include cybersecurity, artificial intelligence, deep learning, IS security and privacy, the Internet of Things, digital forensics, and knowledge management.