Detecting State of Charge False Reporting Attacks via Reinforcement Learning Approach

Mhd Ali Alomrani, Mosaddek Hossain Kamal Tushar, *Member, IEEE*, and Deepa Kundur, *Fellow, IEEE*

***Abstract*— The increased push for green transportation has been apparent to address the alarming increase in atmospheric CO_2 levels, especially in the last five years. The success and popularity of Electric Vehicles (EVs) have led many carmakers to shift to developing clean cars in the next decade. Moreover, many countries around the globe have set aggressive EV target adoption numbers, with some even aiming to ban gasoline cars by 2050. Unlike their gasoline-based counterparts, EVs comprise many sensors, communication channels, and decision-making components vulnerable to cyberattacks. Hence, the unprecedented demand for EVs requires developing robust defenses against these increasingly sophisticated attacks. In particular, recently proposed cyberattacks demonstrate how malicious owners may mislead EV charging networks by sending false data to unlawfully receive higher charging priorities, congest charging schedules, and steal power. This paper proposes a learning-based detection model that can identify deceptive electric vehicles. The model is trained on an original dataset using real driving traces and a malicious dataset generated from a reinforcement learning agent. The Reinforcement Learning (RL) agent is trained to create intelligent and stealthy attacks that can evade simple detection rules while also giving a malicious EV high charging priority. We evaluate the effectiveness of the generated attacks compared to handcrafted attacks. Moreover, our detection model trained with RL-generated attacks displays greater robustness to intelligent and stealthy attacks.**

***Index Terms*— Cybersecurity, deep learning, reinforcement learning, EV charging.**

I. Introduction

IN THE recent decade, the EV revolution has shown a promising solution to the climate change threat by alleviating air pollution in densely populated cities while diversifying energy resources [1]. To facilitate the unprecedented adoption of EVs in the coming decades, interconnected charging infrastructures are being built to manage the charging and discharging of millions of EVs [2]. An EV owner can drive into a charging station and seamlessly plug their car to start charging vehicle batteries. EVs can also contribute to the grid during peak demand by discharging to the network or even powering up appliances in a house during blackouts. Such a bi-directional flow of energy benefits a variety of grid stakeholders [3], [4].

Several protocols have been designed to enable the two-way flow of information and energy between EVs and the energy grid [5]. Such protocols enable the integration of electric vehicles into the smart grid, allow for a consumer-centric way for owners to charge their vehicles, and facilitate the grid's demand for stable and energy-efficient operation. For example, the ISO 15118 protocol governs all charging request communication between an EV and a charging station, such as EV ID, payment info, etc. The Open Charge Point Protocol (OCPP), introduced by Open Charge Alliances, enables public EV charging stations to seamlessly coordinate power flow and information between the EV and the grid [5].

Nonetheless, the inherent complexity of the charging system and the enhanced integration of communication systems within EVs increases the cyber-attack surface. Moreover, the exchange of sensitive and personal information between the EV and charging infrastructure attracts a host of threats. For example, attackers may target user privacy and integrity by accessing location information, EV ID, State of Charge (SoC), and payment info through Man-in-the-Middle attacks [6]. The leaked user data can then be used for unauthorized transactions or generating fake traffic to disrupt the grid [7]. With the lack of vital security measures, EVs can serve as compromised IoT devices that initiate a DDoS attack on charging stations by flooding the network with fake requests [7], [8]. Such attacks may overload the charging schedules and prevent other vehicles from using the grid. Moreover, compromised EVs may modify the "charging profile" to increase demand on the grid during peak hours. The smart grid has difficulty serving the connected load in such events, possibly preventing legitimate consumers from receiving power.

Previous work [7] has shown the original OCCP standard to be particularly vulnerable to such attacks. This is mainly due to communication between parties happening in clear text, thus, allowing attackers to sniff private information easily. Though OCPP developers can deploy the TLS protocol to provide encryption over links, manufacturers often leave these protocols to avoid overhead and additional costs. Nonetheless, even in the presence of TLS, Alcaraz et al. [9] found OCPP vulnerable to various distortion, disruption, and disclosure attacks. For instance, a Man-in-the-Middle attack on the charging station can destabilize the grid. An attacker may perturb

Manuscript received 8 December 2021; revised 23 May 2022, 4 October 2022, and 26 February 2023; accepted 8 May 2023. Date of publication 12 June 2023; date of current version 4 October 2023. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Nous remercions le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) de son soutien and in part by the Univerisity of Toronto for Postdoctoral Fellowship. The Associate Editor for this article was S. Garg. *(Corresponding author: Mosaddek Hossain Kamal Tushar.)*

Mhd Ali Alomrani is with Huawei Noah's Ark lab, Toronto, ON L3R5Y1, Canada (e-mail: mohammad.alomrani@mail.utoronto.ca).

Mosaddek Hossain Kamal Tushar was with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S, Canada. He is now with the Department of Computer Science and Engineering, University of Dhaka, Dhaka 1000, Bangladesh (e-mail: mosaddek.tushar@utoronto.ca).

Deepa Kundur is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S, Canada (e-mail: dkundur@ece.utoronto.ca).

Digital Object Identifier 10.1109/TITS.2023.3281476

power usage or reverse power flow during off-peak times to cause unanticipated loads, potentially initiating blackouts.

Bao et al. [10] analyze the various scenarios of ISO/IEC 15118 protocol where the charging service availability and integrity are compromised. The authors show that the adversary can exploit non-binding certificate authorities to perform masquerade and DoS attacks on charging stations. Another study [6] on ISO 15118 demonstrates the impersonate attack by copying transactions in the RFID chip and consequently disguising itself as another vehicle by replacing its ID with the victim's ID. The charging station receives the ID and can wrongfully write the billing information to the victim. More importantly, ISO 15118 is also susceptible to other attacks that fabricate metering data and battery level (SoC) to give the malicious EV smaller bills and higher charging priority.

The rise of such attack vectors calls for developing intrusion detection systems (IDS) that can effectively detect sophisticated attacks while helping aggregators and owners take appropriate mitigation strategies to isolate malicious actors and EVs. The IDS is vital for SoC tampering attacks wherein an attacker exploits the protocol vulnerabilities discussed above to report false SoC data to the charging infrastructure [11]. A malicious EV can send a charging request with a smaller SoC value to gain higher charging priority. This exploit can scale up an army of compromised or malicious EVs that gain higher priority to preventing benign users from charging, hence, causing what is known as a denial-of-charge attack. An ideal IDS should immediately detect and isolate such malicious charging requests, which may otherwise be difficult to discover by a human expert.

Nevertheless, widely deployed state-of-art IDS, such as signature-based approaches, require knowledge of previous attack signatures and must be updated regularly to detect new attack patterns [12]. This calls for more intelligent IDS that can generalize to novel attack strategies. Several prior works explore deep neural networks (DNNs) to flag suspicious behaviors in the smart grid infrastructure. Recently, DNN-based IDS displayed promising performance in detecting DoS attacks on charging stations [13] and deceptive EVs that report false SoC data [14]. However, while DNNs are known for their powerful generalization capabilities, ability to learn complex tasks and speed [15], they require training on real-world attack data, which is scarce in practice. To alleviate this issue, [14], [16] resorts to augmenting datasets with handcrafted attacks. Synthetic attacks are effective at maliciously gaining higher power allocation and charging priority; however, they may not mimic real-world attack strategies. Consequently, any detection model trained on such datasets may not generalize well to more intelligent attacks encountered in practice.

To solve this problem, we develop a reinforcement learning framework to generate SoC values that illegally fool scheduling mechanisms to favor malign EVs, cause a denial-of-charge to benign EVs, and disrupt the load on the grid. In contrast to previous works, we use the proposed framework to create intelligent (stealthy) attacks that are more effective than handcrafted ones. We evaluate the robustness of a detection model trained on the generated attacks and a dataset of benign EVs to classify malicious and honest EVs. Moreover, we show that using reinforcement learning to create potential attacks gives rise to more novel attack schemes from which the detection model can learn. Lastly, we demonstrate our model's robustness compared to the one trained on handcrafted attacks.

Our contributions can be summarized as follows:

- We develop a novel reinforcement learning framework to generate intelligent and stealthy attacks to falsify the SoC.
- We show that RL-generated attacks are considerably more effective than handcrafted ones.
- We train a more robust detection model on a combination of real-world user behavior and generated intelligent attacks.

The remainder of this paper is organized as follows: In the next section (Section II), we provide an overview of the charging infrastructure and relevant parties, related work on deep learning for EV cybersecurity, and the problem formulation and threat model considered. Section III outlines the RL attack methodology. Section IV discusses benign and malicious data generation. Section V proposes the detection model, followed by experimental details in Section VI. Finally, we analyze our findings in Section VII, and the conclusion is drawn in Section VIII.

II. Background and Problem Formulation

A. EV Infrastructure and Protocols

A typical EV charging infrastructure [2] is comprised of the following key stakeholders:

- EV owner and EV
- Electric vehicle supply equipment (**EVSE**): The device that connects the EV to the grid.
- Charging station (**CS**): A station equipped with many EVSEs.
- Aggregator: Responsible for collecting EV charging requests and sending them to the CC.
- Control Centre: A central management system that manages the power grid and supervises the energy requests by charging stations and EVs. The allocation of energy requests happens through an internal charging coordinator (**CC**).

The data shared among stakeholders is sent through several protocols. We highlight the critical data transmitted across multiple communication links and the corresponding protocols that attackers can exploit:

- The EV interacts with the CS (or CC) to reserve an EVSE and exchange charging parameters, including EV ID, location, SoC, payment info, etc. IEC/ISO 15118-1/2/3 protocols expedite such communication.
- The CS and control center exchange incoming EV data to ensure the availability of EVSE.
- The control center negotiates power usage, scheduling, and pricing with the power distributor. The aggregator forwards the charging schedules back to the EVs.
- The EV and EVSE employ a physical power line to exchange electricity and charging parameters (such as SoC). The flow of electricity is bi-directional to allow both charging and discharging.

Communication between the CS and EVSE happens through the J1772 standard, while OCPP governs all communication among EVSE, control center, and grid.

B. Deep Learning for EV Cybersecurity

In this section, we explore several works which utilize deep neural networks (DNNs) to flag suspicious behaviors in the smart charging infrastructure.

Kuadey et al. [17] devise a Long-Short-Term-Memory (LSTM) based DeepSecure framework to detect DDoS attacks on user equipment network traffic in the fifth-generation cellular network. The proposed method assigns an appropriate slice to a legitimate user equipment request, learns the network traffic features, and distinguishes malicious traffic from legitimate traffic. Their experiments show that the proposed LSTM method outperforms the previous deep-learning-based detection methods detecting DDoS attacks.

Zhang et al. [18] provide a survey and analyze the efficiency of deep learning-based attack detection models for the cyber-physical system. The authors demonstrate that with excellent accuracy and precision, the Recurrent Neural Network, LSTM, and Convolutional Neural Network models outperform baseline methods such as Support Vector Machines.

Furthermore, Basnet and Ali [13] introduce a novel deep learning-based IDS for detecting DoS attacks on EVSE servers. In such attacks, an attacker exploits the EVSE server to launch any SYN floods, buffer overflow, or teardrop attacks to compromise the availability of the grid's resources. The authors extract critical features from the attack data to train a feed-forward neural network (FNN) and LSTM network [19] that implicitly learn the digital fingerprints of DoS/DDoS attacks resulting in 99% detection accuracy.

Nonetheless, though the authors assume that such datasets are ideal candidates for learning, they do not include attacks on EV charging infrastructures that may differ in nature. For example, an attacker in the EV setting may intelligently overload charging schedules with fake requests rather than aim to increase the CPU and memory consumption of servers. Moreover, their models do not leverage EV charging information such as EV ID and previous charging history, which can be tremendously helpful in identifying benign and malicious charging behavior.

Bhusal et al. [20] propose a deep learning-based multi-label classification approach to detect coordinated data falsification attacks on distributed generators' power output. The derived solution can identify coordinated, additive, deductive, and combined attacks. The authors demonstrate the developed method on several systems, including a 240-node distribution system with 99% attack detection accuracy, and compare the proposed method with FNN, Convolutional Neural networks, and Recurrent Neural networks.

Shafee et al. [14] devise a machine-learning model to identify malicious EVs which report false SoC data to the charging coordinator (CC). The CC schedules EV charging based on its SoC and assigns higher priority to cars with lower SoC. Therefore, such attacks may congest charging schedules and overload the grid when coordinated with other malicious EVs. Consequently, the authors propose using the real-world charging behavior of plug-in hybrid electric vehicles combined with synthesized false reporting attacks to train a DNN with the gated recurrent unit (GRU) architecture. The GRU model can identify EVs that deviate from their benign behavior throughout the day and flag them as malicious. Their GRU model can accurately detect deceptive vehicles and demonstrates good generalization abilities in detecting new attacks. However, the synthetic attacks used in [14] and [16] do not explicitly utilize additional information to predict more intelligent attack strategies. An attacker who knows other incoming charging requests may adaptively report false SoC data while remaining undetected. Hence, detection models trained on hand-crafted attacks are not robust against more intelligent or stealthy attackers that may appear in practice.

Rahman et al. [11] study false data injection attacks on the EV battery management system. They show that attacks that tamper with energy requests lead to out-of-service vehicles, power grid destabilization, and battery pack damage via overcharging. Their proposed DNN model estimates current battery SoC based on low-level vital metrics such as temperature, open-circuit voltage, capacity, and power. After training the SoC estimator, the authors detect false reporting attacks by measuring the mean absolute percentage error between the estimated actual SoC and spoofed SoC values.

In conclusion, previous works in the literature have either used handcrafted attacks or general attack data on non-EV infrastructures to train detection models due to the lack of real-world attack data. While such attacks can work in practice, a more intelligent attacker with reconnaissance capabilities can learn to game the EV charging infrastructure with novel and adaptive attack schemes. Hence, the proposed detection models are not trustworthy enough to be deployed in practice.

C. Problem Formulation

The primary goal of this work is to design a robust DNN-based IDS that can identify attackers that report fake SoC values by observing their historical charging requests. The attacker represents a malicious EV that aims to gain higher charging priority by reporting a fake SoC value when sending a charging request. Our attack model assumes the attacker possesses reconnaissance capabilities and can sniff and edit messages between communicating parties, particularly the EVs and charging stations. The attacker knows all incoming charging requests to the charging coordinator but can only distort the malicious EV's SoC value. These abilities can be attained by exploiting the ISO 15118 and OCCP protocols using the strategies discussed in Section I.

We consider a charging system that comprises of a charging coordinator (CC), aggregator, and EV community, as described in Section II-A. Time across the day is divided into T time slots of equal length. At the beginning of each time slot, EVs that need charging send charging requests to the aggregator. The aggregator, in turn, forwards the charging request to the CC for scheduling. The requests contain essential information for scheduling, such as the battery SoC and time-to-complete-charge (TCC). Once the CC receives all charging requests from

TABLE I
SoC FALSE REPORTING ATTACKS [14]. t_a AND t_b ARE RANDOMLY GENERATED INTERVAL BOUNDS

Attacks	Attack Scheme
Attack 1	$\psi_i(d,t) = \alpha S_i(d,t)$
Attack 2	$\psi_i(d,t) = \beta_i(d,t) S_i(d,t)$
Attack 3	$\psi_i(d,t) = \begin{cases} 0 & t_a \le t \le t_b \\ S_i(d,t) & \text{otherwise} \end{cases}$
Attack 4	$\psi_i(d,t) = \begin{cases} 0 & t_a \le t \le t_b \\ \beta_i(d,t) S_i(d,t) & \text{otherwise} \end{cases}$

a particular area, the charging coordination mechanism [21] prioritizes a subset of EVs such that the total allocated power does not exceed the energy capacity C. More concretely, for each charging request j, the CC receives SoC_j, TCC_j, and the energy demand η_j to construct a priority index p_j for an EV (j):

$$p_j = \epsilon f_1(SoC_j) + (1-\epsilon) f_2(TCC_j) \tag{1}$$

where $0 \le \epsilon \le 1$, f_1, and f_2 are functions that map SoC and TCC to values between 0 and 1. The CC then divides the priority index of each vehicle p_j by its energy demand (η_j) and selects the EVs with the highest ratios for charging such that the maximum charging capacity (C) is not exceeded as,

$$\sum_{j \in o} \eta_j \le C; \quad \text{where} \quad o \subseteq n \tag{2}$$

where n is the set of EVs that send requests for charging, and o is the subset of n with higher priority that satisfy the constraint given in eq. (2). Consequently, for any EV j, the CC allows it to charge with energy demand η_j or defer the request for a future time slot. The energy demand η_j is defined in terms of the SoC: $\eta_j = (1 - SoC_j)B$ where B is the total battery capacity of a vehicle.

While such efficient CC mechanisms help maintain the grid's stability and preserve the users' privacy, they naively assume that EVs report correct charging data, such as SoC and TCC. *With the lack of intelligent detection mechanisms, a malicious EV may report false SoC data to obtain higher priority and energy allocation*. Consequently, malicious EV with basic false reporting capabilities can steal power, congest charging schedules, and destabilize the grid.

Various attack strategies have been proposed in [14] and [16], where an EV reports a false SoC value at day d and time t for EV i, denoted by $\psi_i(d,t)$, as shown in Table I. The existing research shows that such attacks disrupt the scheduling scheme; however, they can strongly deviate from normal charging behavior, making them easy to detect, and do not utilize the current state of the charging schedules to their advantage, e.g., reporting SoC value of 0 may not be wise at some times of the day. Therefore, their model is not robust against more intelligent false reporting attacks. An attacker with knowledge of incoming charging requests of other EVs can generate fake SoC values to gain higher priority and power while also evading being detected.

Our paper's main contribution is that we employ reinforcement learning methods to generate realistic and stealthy attacks that can evade basic detection mechanisms and provide higher priority to the attacker. We combine these intelligent attacks with the handcrafted attacks in Table I and a normal behavior dataset to train a detection model to classify an EV as honest or malicious accurately.

In Table I, $S_i(d,t)$ denotes the real SoC value of EV i at day d and time t. α is a constant less than 1. $\beta_i(d,t)$ is a hand-picked time-dependent function between 0 and 1. In principle, the attacks randomly report a lower SoC value as charging coordinators generally give cars with low batteries higher priority to charge. Attack 1 scales the real SoC by a constant $\alpha < 1$. Attack 2 scales the SoC by a time-dependent function $\beta_i(d,t)$ between 0 and 1. Attacks 3 and 4 reports an SoC value of 0 on random intervals of the time horizon.

III. RL ATTACK METHODOLOGY

We propose using simple feed-forward neural networks to detect malicious behavior against CCs by leveraging historical data such as SoC and TCC. Our approach is similar to the one presented in [14] and [16], except that we decompose training into two stages. First, we train (see Fig. 2) an RL agent to generate charging requests with fake SoC values that maximize the energy allocation and priority of a malicious EV. Second, we train a feed-forward neural network on a dataset composed of the generated attacks by the trained RL agent and regular charging requests by benign EVs to detect malicious behavior.

The advantages of such decomposition are three-fold. First, the RL agent can adapt its attack strategy to any scheduling mechanism instead of general handcrafted attacks, thus making the detection model more robust to various deceptive strategies. Second, the RL agent can be trained to evade a detection mechanism by giving it a low reward signal if it deviates from the actual behavior. Lastly, deep reinforcement learning methods learn through interactions with the environment, requiring little data to generate attacks. In contrast, generative models, such as GANs [22], need a dataset of attacks and can only create samples that are as good as the dataset.

To train the RL-based SoC attack method within an RL framework, we must formulate the SoC attack problem as a Markov decision process:

- **State**: At timestep, t, a state s_t consists of the incoming charging requests of all EVs, represented by a vector of the current SoC values of all incoming requests.
- **Action**: At each timestep t, the agent must choose to perturb the actual current SoC value by a continuous normalized amount $a_t \in [-1, 1]$. The agent only perturbs the SoC value $S_i(t)$ of the malicious vehicle i:

$$\psi_i(t) = S_i(t) + a_t \tag{3}$$

  $\psi_i(t)$ represents the perturbed SoC value of malicious EV i at time t.
- **Reward function**: The terminal reward ω is the sum of the power allocated to the malicious EV i by the charging scheduler across all timesteps. To encourage the agent to be stealthy, we subtract the term γa_t where the γ parameter controls the significance of this term. In RL terms, the total power is an *extrinsic* reward signal while

γa_t is an intrinsic reward signal. Therefore, the ultimate measure of performance we care about improving is the value of the extrinsic reward achieved by the agent; the intrinsic reward serves only to motivate the agent to be stealthy:

$$\omega = \sum_{t=1}^{T} (k_i(t) - \gamma a_t) \tag{4}$$

where $k_i(t)$ is the power allocated to the malicious EV i at timestep t, and T is the total number of timesteps in an episode.

- **Policy**: We define a solution as a set of actions $\pi = \{a_1, \ldots, a_T\}$, representing the perturbations of the SoC values reported at each timestep. The policy network defines a stochastic policy $p(\pi|S)$ for selecting a solution π given the sequence of actual SoC values of all EVs, denoted by S. It is parameterized by θ and can be factorized as:

$$p_\theta(\pi|S) = \prod_{t}^{T} p_\theta(a_t|s_t) \tag{5}$$

When $\gamma = 0$, the RL agent only aims to maximize the power gained, regardless of the amount of perturbation. However, large perturbations are easy to detect. Therefore, we subtract the term γa_t, which decreases the reward in case of large perturbations and, thus, discourages the agent from deviating too much from normal behavior. The RL agent must balance being stealthy and gaining more power. The γ hyperparameter lets us control how much we care about stealthiness.

A. Adversarial RL Agent

To enhance training data for a more accurate FNN cyberattack detection model, we make use of an adversarial RL agent to generate synthetic attacks that are presented in this section.

1) Adversarial RL Agent Simulation Strategy: Consider a set of n EVs and a set of time slots of equal length across a day (episode). At each time-step t, any EV that wants to charge sends a request to the CC with its current SoC. Once the CC receives all charging requests at time-step t, it sends back the power allocations for each charging demand, with some recommendations receiving no power.

Algorithm 1 outlines the RL agent training in the charging environment. Let i be the index of a malicious EV, and $CC(.) : \mathbb{R}^{c+1} \rightarrow \mathbb{R}^{c+1}$ be the charging coordinator, N be the training epochs, λ be the arriving rate, D be the benign training set and the α be the learning rate. The number of arriving charging requests (other than malicious EV) is c, determined by the Poisson distribution in line 8. The number of vehicles is n. At the beginning of each training episode, we sample a benign SoC sequence S_i from the real-world training set. This represents the real SoC values of i throughout one day. Let $S_i(t) \in \mathbb{R}$ be the real SoC value of malicious vehicle i at time t. To simulate the benign incoming charging requests, we also sample c, the number of other arriving charging requests at the current timestep t, from the Poisson distribution to mimic real-world charging request arrivals. Then, the c benign SoC values are sampled from the uniform distribution. $S_b(t) \in \mathbb{R}^c$ represents the reported SoC values of all benign vehicles. Before sending the charging requests to the CC, the RL agent perturbs $S_i(t)$ by $a_t \in [-1, 1]$. Finally, the CC returns $k(t)$, the power allocation for each charging request. $k_i(t)$ represents the power allocation of i.

In our simulation, all vehicles have a battery capacity of B of 200 kWh. The total power available at the charging station is 1500 kWh. We assume the number of arriving charging requests at times-step t follows the Poisson distribution with arrival rate λ. This is observed in real-world charging stations where incoming EVs expect to be charged as soon as possible [23], [24]. We simulate 48-time steps, equivalent to the training set's length of the SoC sequences.

Algorithm 1 RL Agent Training in Charging Simulation

Input: p_θ, CC, N, D, λ, α, B.
Initialize $T \leftarrow 48$.
Initialize $n \leftarrow 30$
for $i \leftarrow 0$ to N **do**
  Initialize $\omega \leftarrow 0$ ▷ Reward for malicious vehicle
  $S_i \leftarrow \{D\}$ ▷ Real SoC sequence for malicious vehicle
  for $t \leftarrow 1$ **to** T **do**
    $c \sim \rho(\lambda)$ ▷ Number of arriving charging request sample
    $S_b(t) \sim U[0, 1]$ ▷ SoC value for benign EV on sample c
    $\psi(t) \leftarrow S_b(t) \cup S_i(t)$
    $(\mu, \sigma) \leftarrow p_\theta(\psi(t))$
    $a_t \sim N(\mu, \sigma)$ ▷ Action sampled from Normal distribution
    $\psi(t) \leftarrow S_b(t) \cup (S_i(t) + a_t)$ ▷ Perturbed SoC value for malicious EV
    $\eta(t) \leftarrow (1 - \psi(t))B$ ▷ Power demand
    $k(t) \leftarrow CC(\eta(t))$ ▷ Power allocation at t for each charging request
    $\omega \leftarrow \omega + (k_i(t) - \gamma a_t)$ ▷ Update reward for malicious EV
  end for
  $\theta \leftarrow \theta - \alpha \nabla L(\theta|S)$
end for

B. Architecture and Training

Our policy network $p_\theta(a_t|s_t)$ utilizes the feed-forward architecture. The forward feed model inputs a list of all vehicles' actual SoC values, including the malicious EV at the current timestep. The input is followed by 3 hidden layers of size 236 neurons. Each hidden layer is followed by a ReLU activation unit [25] to introduce non-linearity. The final output layer consists of 2 neurons representing the mean μ and standard deviation σ of a normal distribution $\mathcal{N}(a_t|\mu, \sigma)$. The continuous action a_t, which represents the value by which the malicious EV's SoC value is perturbed, is sampled from the normal distribution. This is a classic trick to allow RL agents to operate on continuous action spaces [26]. The policy network is trained separately and uses a different FNN than the detection model (See Section V).

The action policy is trained using policy gradient reinforcement learning [27] for its effectiveness and simplicity. That is, we learn the policy parameter θ by optimizing the loss $L(\theta|S)$ using gradient descent:

$$\nabla L(\theta|S) = \mathbb{E}_{p_\theta(\pi|S)}[(L(\pi) - b(\pi))\nabla \log \ p_\theta(\pi|S)]$$

where $L(\pi) = -r$ with reward $r = \omega$, and S is sequence of real SoC values of all EVs. In principle, the policy gradient algorithm reinforces actions that maximize the expected outcome and discourages actions that give a low reward.

TABLE II
ATTACK TYPES THAT EACH MODEL IS TRAINED ON

Model	Attack Types	
Model 1	Attacks 1-4	
Model 2	$2 \times \gamma = 0.6$	
Model 3	$2 \times \gamma = 0.6$	Attacks 1-4

To reduce gradient variance and noise, we add a baseline $b(s)$ which is the exponential moving average [28], $b(\pi) = M$, where M is the loss $L(\pi)$ in the first training iteration. The update step is $b(\pi) = \beta M + (1 - \beta)L(\pi)$ with *decay* β. In principle, our formulation can support any RL algorithm that operates in a Markov Decision Process [26]. We choose the policy gradient algorithm combined with an effective baseline as this setup has been shown to work on several real-world problems [29], [30] and has minimal hyper-parameters.

It is important to note that the proposed adversarial RL framework is not confined to EV charging attacks but can be applied to many attack formulations against the grid. For example, the RL agent could report fraudulent home power usage to disrupt the smart grid or steal power. We leave such attacks for future work.

IV. DATASET

The generation of training data is of paramount importance to the success of detecting malicious data. In this section, we outline our approach for dataset generation for both benign and malicious samples.

A. Benign Dataset

We use a dataset of 536 Plugin Hybrid Electric Vehicles (taxis) [31] that reported their locations (latitude and longitude) every minute and charging times for 24 days. We also assume that the data represents the Kia Soul EV [32] and use Kia's charging rates and battery capacity to estimate the minute-by-minute SoC values from the driving traces. During charging or driving, the SoC value is updated using the following respective equations:

$$SoC = SoC + \frac{\text{Charging rate} \times \text{duration}}{\text{Battery Capacity}} \tag{6}$$

$$SoC = SoC - \frac{\text{Consumption rate} \times \text{duration}}{\text{Battery Capacity}} \tag{7}$$

To create a data sample, we sample the SoC value every 30 minutes to create a sequence of 48 SoC values for one day. In total, we have 536 taxis $\times$ 24 days $=$ 12864 data samples. Fig. 1 shows the SoC distribution of 2 taxis reported over 23 days. It can be seen that each EV has a unique behavior. Therefore, we must employ a complex model to learn the temporal behavior of the taxis and detect any malicious deviation.

B. Malicious Dataset

In addition to the handcrafted attacks outlined in Table I, we use the trained RL agent to generate intelligent, stealthy attacks from each data sample. We deploy the RL agent in the charging simulation to perturb the malicious EV's actual SoC values for each data sample in the benign dataset. The resulting perturbed sequence is labeled as a malicious data sample. We generate different malicious samples for each benign sample to ensure that we have rich attack data. To get new attack samples, we deploy the RL agent in the charging simulation multiple times but with different random seeds each time. The ADASYN [33] method is then used as a data augmentation technique to balance the ratio of benign samples to malicious samples. The entire framework can be seen in Figure 2.

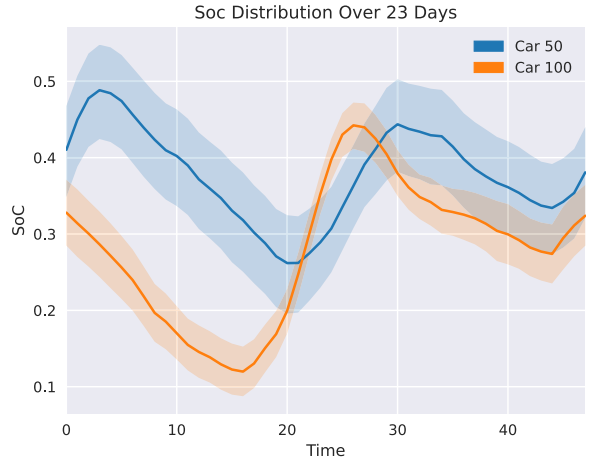

Fig. 1. SoC Distribution per hour of two taxis over 23 days.

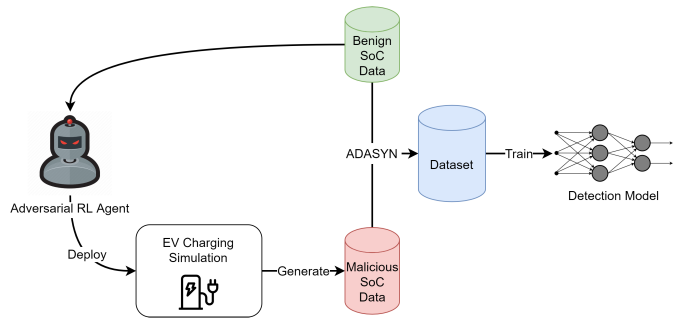

Fig. 2. The full training pipeline for the detection model. Note that the adversarial RL agent used here is already trained.

V. MODEL FOR CYBERATTACK DETECTION

This section outlines the architecture and training of the detection model, which utilizes the feed-forward architecture to classify benign and malicious behavior.

Feed-forward neural networks (FNN) have displayed phenomenal results over the past few years for their generalization abilities, and powerful learning capacity [15]. FNNs are multi-layer perceptrons where each node takes input from all nodes in the previous layer and passes it on to all nodes in the next layer. Each node applies weights to the inputs before applying a non-linearity such as a ReLU [25] or Sigmoid. The successive layers of non-linearities give the FNN powerful learning capacities. In our experiments, we use a similar detection model as in [14]. Our FNN comprises six hidden layers of size 768 neurons with the ReLU

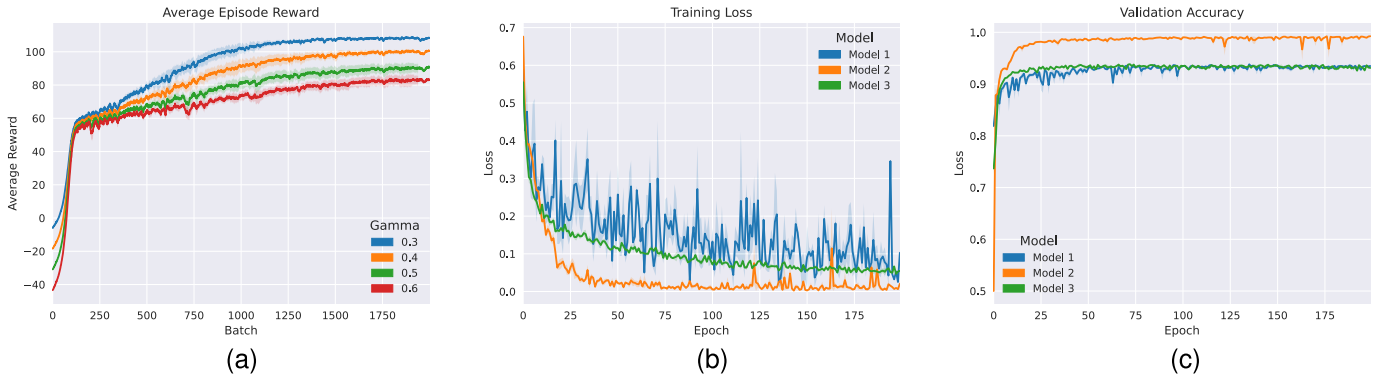

Fig. 3. Training plots for RL agent and detection models. (a) Average Episode Reward throughout training for 4 RL agents with different random seeds. (b) Average training loss from 4 runs with different random seeds for all models. (c) Validation accuracy throughout training.

non-linearity. The input layer receives the SoC sequence for one day $\psi(d, *)$, and the output layer consists of 2 neurons followed by a softmax layer which outputs the probability of the sequence being malicious. Since over-fitting can cause a severe problem in our large model, we add a dropout [34] layer after each hidden layer. Dropout is a technique that addresses over-fitting by randomly dropping out some nodes in the network during training with probability p.[1] This prevents the nodes in an extensive network from co-adapting too much and thus performs better generalization.

VI. Experimental Details

The charging simulation has been implemented in Python 3.7. We use the Pytorch library [35] for the training and evaluation of all deep learning models. All models are trained using the Adam optimizer [36]. Hyper-parameters such as the learning rate and exponential decay are tuned on the validation set using the grid search method. See repository[2] for details.

A. RL Agent Training

Figure 3 shows the average reward per episode as training progresses for 120 epochs. The models are fairly robust across different random runs and converge quickly. We train 4 agents with $\gamma \in [0.3, 0.4, 0.5, 0.6]$. One can see that the higher the γ the lower the reward that the agents converge to. This is due to the trade-off between stealthiness and gaining more power and priority in the charging schedules. A higher γ forces the agent to make smaller perturbations while also gaining more power than benign EVs.

B. Detection Model Training

To experiment with the effectiveness of the intelligent attacks, we train the detection model on three datasets with different combinations of malicious samples. Table II outlines the malicious attacks that are generated per benign sample using attack models described in Section III. Models 1, 2, and 3 are the same, except that each is trained on a dataset with different combinations of malicious samples. Attacks 1-4 represents the hand-crafted samples described in Table I.

[1] A tuned hyper-parameter.

[2] https://github.com/alomrani/ev-charging-rl-attacks.git

TABLE III

Dataset Split for Each Model

Model	Training	Validation	Test
Model 1	95k	4000	4000
Model 2	43k	4000	4000
Model 3	145k	4000	4000

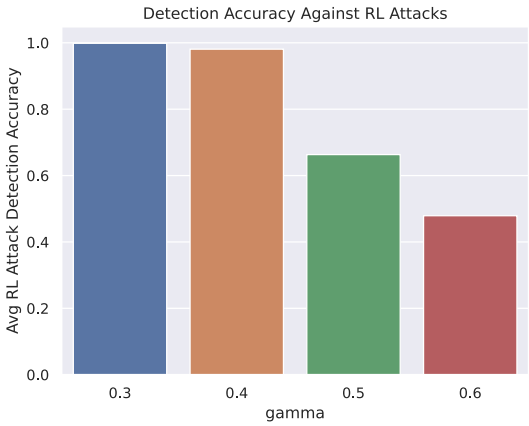

Fig. 4. Model 1 avg Detection Accuracy on datasets of random malicious samples generated by RL Agents with different γ. Results are averaged over 4 runs with different seeds.

γ attacks represent malicious samples generated by an RL agent trained with γ—the dataset size after ADASYN augmentation is shown in Table III.

We train all detection models for 200 epochs. Figure 3c shows the training results of all models. Notably, the addition of dropout layers allows the model to learn for a long period without over-fitting too much of the training data, as evident in the first plot.

VII. Numerical Evaluation

A. Effect of γ

In Figure 5, we visualize the SoC perturbed sequence reported by some malicious EV versus the real SoC sequence under different γ settings. For $\gamma = 0.3$, the RL agent mostly cares about maximizing the amount of power received across all time steps; Thus, the perturbed sequence is stochastically compared to the actual sequence representing a real-world EV's behavior. As γ increases, the reported SoC values

TABLE IV

EFFECTIVENESS OF RL ATTACKS ON CHARGING SIMULATION AND DETECTION MODELS. EACH TEST ATTACK DATASET WAS GENERATED 4 TIMES WITH DIFFERENT RANDOM SEEDS. WE REPORT (MEAN ± STD) FOR ALL RESULTS. ALL MODELS' BEST ACCURACY IS HIGHLIGHTED IN **BOLD**

Attack Type	Avg. (KWh) Malicious Vehicle	Avg. (KWh) Benign Vehicle	Model 1 Accuracy	Model 2 Accuracy	Model 3 Accuracy
$\gamma = 0.3$	126.32 ± 0.26	44.52 ± 0.004	0.99 ± 0.001	$\mathbf{1.0 \pm 0}$	$\mathbf{1.0 \pm 0}$
$\gamma = 0.4$	117.8 ± 0.35	44.78 ± 0.0001	0.93 ± 0.005	$\mathbf{1.0 \pm 0}$	$\mathbf{1.0 \pm 0}$
$\gamma = 0.5$	97.8 ± 0.29	45.38 ± 0.01	0.5 ± 0.004	$\mathbf{1.0 \pm 0}$	$\mathbf{1.0 \pm 0}$
$\gamma = 0.6$	88.65 ± 0.08	45.65 ± 0.006	0.33 ± 0.005	$\mathbf{0.99 \pm 0.001}$	$\mathbf{0.99 \pm 0.007}$
Attack 1	59.32 ± 0.1	46.78 ± 0.007	$\mathbf{0.9 \pm 0.005}$	0.003 ± 0.001	0.88 ± 0.01
Attack 2	59.6 ± 0.9	46.7 ± 0.03	$\mathbf{0.98 \pm 0.002}$	0.42 ± 0.004	$\mathbf{0.98 \pm 0.003}$
Attack 3	35.8 ± 0.6	47.6 ± 0.02	0.98 ± 0.002	0.002 ± 0.001	$\mathbf{0.99 \pm 0.003}$
Attack 4	34.8 ± 0.67	47.6 ± 0.03	$\mathbf{0.99 \pm 0.001}$	0.02 ± 0.001	$\mathbf{0.99 \pm 0.001}$
No Attacks	47	46.6	0.91	**0.98**	0.9

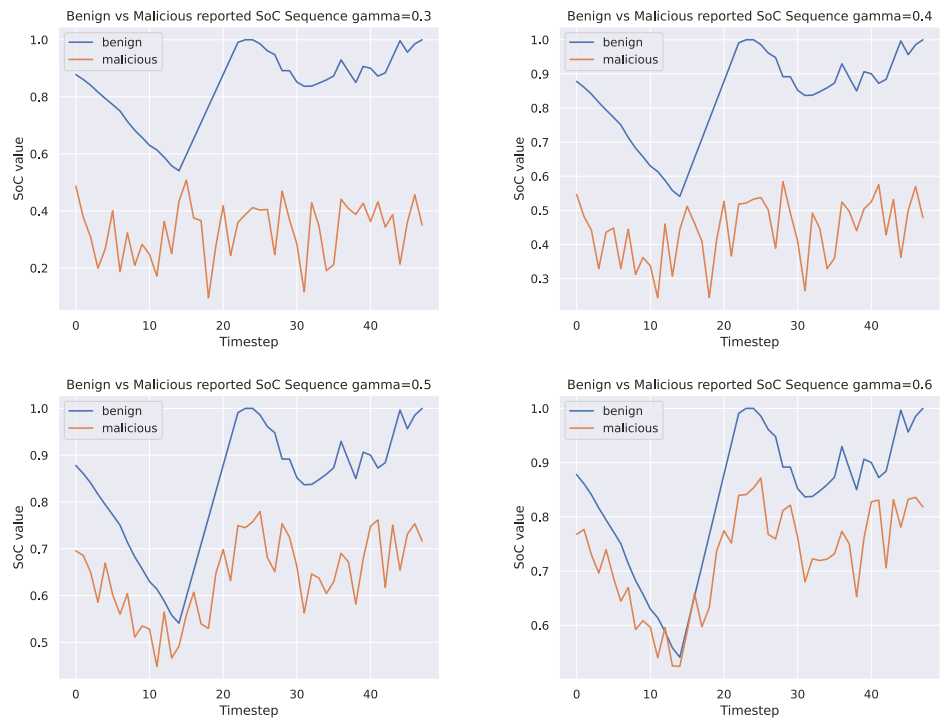

Fig. 5. Malicious vs. Benign SoC sequence generated by the RL agents on a random EV.

become closer to the valid values, and therefore the SoC sequence across the day becomes more realistic.

Table IV shows the average power allocated to the malicious EV per timestep by the charging coordinator. For $\gamma = 0.3$, the malicious EV is allocated 3 times more power than a benign EV. However, as seen in Fig. 5, the RL agent deviates significantly from real SoC values, making this abnormal charging pattern easily detectable. For $\gamma = 0.6$, the RL agent is encouraged to make small perturbations only and, therefore, receives less power than with $\gamma = 0.3$. However, the malicious EV is still given $\geq 90\%$ more power than the average benign vehicle while staying close to the true SoC values (See Fig. 5 for an example).

These results further show that the role of γ is necessary to make the RL attacks hard to detect while remaining effective at stealing more power than benign EVs. Therefore, the RL agent can learn novel attack strategies that exploit the charging coordination mechanism while remaining stealthy. Notably, based on the knapsack algorithm, our charging coordinator is designed to be fair and efficient to all requests [21]. However, without a detection mechanism in place, it is evident that one can "game" the charging schedules by spying on other requests to gain more priority.

Moreover, the intrinsic motivation controlled by γ encourages the agent to learn stealthy attack strategies that can mislead a naive detection model. Fig. 4 plots the detection accuracy of Model 1, proposed by [14] and trained on the handcrafted attacks only against the RL attacks. For $\gamma = 0.3, 0.4, 0.5$, Model 2-3 can detect these attacks with full accuracy since they are too stochastic and deviate significantly from normal charging behavior (See Fig. 5 for an example). However, the accuracy steadily decreases with higher γ, eventually approaching a value of approximately 0.5, which means the model cannot distinguish between benign charging behavior and intelligent RL attacks.

We note that Attacks 1 and 2 provide the malicious EV with approximately 10% more power than the average EV but are less effective than the RL attacks. Attacks 3 and 4 give the malicious EV less power than a benign EV, although they follow similar strategies to Attacks 1 and 2. We believe this is

due to reporting an SoC value of 0 on some intervals, giving the EV a low priority in the charging schedules and, hence, less power on average.

B. Remarks on Detection Models

To investigate the generalization ability of the detection models, we include the detection accuracies of Models 2 and 3, which were trained on different attack types. For Model 2, trained on RL attacks only, the model performs well on all γ attacks, although it was only trained on $\gamma = 0.6$ attacks. However, the model cannot generalize well to all the handcrafted attacks because they are entirely different attack strategies. Therefore, for Model 3, we include both RL attacks and handcrafted attacks. As a result, Model 3 can detect both handcrafted and intelligent RL attacks and correctly identify normal charging behavior with reasonable accuracy. This further motivates combining RL attacks and well-handcrafted attacks to get a robust model.

VIII. Conclusion

We developed a machine learning-based method to detect intelligent attacks against the EV charging system. A novel RL approach was utilized to generate attacks aiming to gain higher priority and power in charging coordination systems. The adversarial RL policy was trained in a charging simulation using policy gradient to effectively perturb the real battery level before being sent to the CC. We add intrinsic motivation to the reward signal to encourage the agent to be stealthy. We show that RL-generated attacks are more effective than handcrafted attacks in gaining more power while also being undetectable by models trained on handcrafted attacks. Finally, we train an FNN on a combination of handcrafted and RL-generated samples to obtain a more robust detection model against various attack strategies. We outline a few future directions worthy of exploration:

- Multi-Agent Attacks: Our results show the effectiveness of only one malicious agent attacking the charging environment. However, a collaborative attack between multiple agents should yield far more damage on the grid and is more challenging to detect as multiple parties are involved.
- Extension to More Attack Settings: Our framework is not restricted to EV charging environments. It can be extended to other settings where real-world attack data is not readily available such as homeowners who report false power usage.

References

[1] I. E. Agency. (2009). *Transport Energy and CO_2: Moving towards Sustainability*. [Online]. Available: https://www.oecd-ilibrary.org/content/publication/9789264073173-en

[2] Q. Wang, X. Liu, J. Du, and F. Kong, "Smart charging for electric vehicles: A survey from the algorithmic perspective," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1500–1517, 2nd Quart., 2016.

[3] M. Mültin, "ISO 15118 as the enabler of vehicle-to-grid applications," in *Proc. Int. Conf. Electr. Electron. Technol. Automot.*, Jul. 2018, pp. 1–6.

[4] C. Liu, K. T. Chau, D. Wu, and S. Gao, "Opportunities and challenges of vehicle-to-home, vehicle-to-vehicle, and vehicle-to-grid technologies," *Proc. IEEE*, vol. 101, no. 11, pp. 2409–2427, Nov. 2013.

[5] M. Parchomiuk, A. Moradewicz, and H. Gawinski, "An overview of electric vehicles fast charging infrastructure," in *Proc. Prog. Appl. Electr. Eng. (PAEE)*, Jun. 2019, pp. 1–5.

[6] S. Lee, Y. Park, H. Lim, and T. Shon, "Study on analysis of security vulnerabilities and countermeasures in ISO/IEC 15118 based electric vehicle charging technology," in *Proc. Int. Conf. IT Converg. Secur. (ICITCS)*, Oct. 2014, pp. 1–4.

[7] J. Antoun, M. E. Kabir, B. Moussa, R. Atallah, and C. Assi, "A detailed security assessment of the EV charging ecosystem," *IEEE Netw.*, vol. 34, no. 3, pp. 200–207, May 2020.

[8] S. Mousavian, M. Erol-Kantarci, L. Wu, and T. Ortmeyer, "A risk-based optimization model for electric vehicle infrastructure response to cyber attacks," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6160–6169, Nov. 2018.

[9] C. Alcaraz, J. Lopez, and S. Wolthusen, "OCPP protocol: Security threats and challenges," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2452–2459, Sep. 2017.

[10] K. Bao, H. Valev, M. Wagner, and H. Schmeck, "A threat analysis of the vehicle-to-grid charging protocol ISO 15118," *Comput. Sci. Res. Develop.*, vol. 33, nos. 1–2, pp. 3–12, Feb. 2018.

[11] S. Rahman, H. Aburub, Y. Mekonnen, and A. I. Sarwat, "A study of EV BMS cyber security based on neural network SOC prediction," in *Proc. IEEE/PES Transmiss. Distrib. Conf. Expo. (T&D)*, Apr. 2018, pp. 1–5.

[12] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, Dec. 2019.

[13] M. Basnet and M. H. Ali, "Deep learning-based intrusion detection system for electric vehicle charging station," in *Proc. 2nd Int. Conf. Smart Power Internet Energy Syst. (SPIES)*, Sep. 2020, pp. 408–413.

[14] A. A. Shafee, M. M. Fouda, M. M. E. A. Mahmoud, A. J. Aljohani, W. Alasmary, and F. Amsaad, "Detection of lying electrical vehicles in charging coordination using deep learning," *IEEE Access*, vol. 8, pp. 179400–179414, 2020.

[15] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in deep learning," in *Mathematical Aspects of Deep Learning*. Cambridge Univ. Press, Dec. 2022, pp. 112–148, doi: 10.1017/9781009025096.003.

[16] M. Nabil, M. Ismail, M. M. E. A. Mahmoud, W. Alasmary, and E. Serpedin, "PPETD: Privacy-preserving electricity theft detection scheme with load monitoring and billing for AMI networks," *IEEE Access*, vol. 7, pp. 96334–96348, 2019.

[17] N. A. E. Kuadey, G. T. Maale, T. Kwantwi, G. Sun, and G. Liu, "DeepSecure: Detection of distributed denial of service attacks on 5G network slicing—Deep learning approach," *IEEE Wireless Commun. Lett.*, vol. 11, no. 3, pp. 488–492, Mar. 2022.

[18] J. Zhang, L. Pan, Q. Han, C. Chen, S. Wen, and Y. Xiang, "Deep learning based attack detection for cyber-physical system cybersecurity: A survey," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 3, pp. 377–391, Mar. 2022.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[20] N. Bhusal, M. Gautam, R. M. Shukla, M. Benidris, and S. Sengupta, "Coordinated data falsification attack detection in the domain of distributed generation using deep learning," *Int. J. Electr. Power Energy Syst.*, vol. 134, Jan. 2022, Art. no. 107345. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061521005846

[21] M. Baza et al., "Privacy-preserving and collusion-resistant charging coordination schemes for smart grids," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 4, pp. 2226–2243, Jul. 2022.

[22] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, Eds., vol. 27. Red Hook, NY, USA: Curran Associates, 2014, pp. 2672–2680. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[23] M. Alizadeh, A. Scaglione, J. Davies, and K. S. Kurani, "A scalable stochastic model for the electricity demand of electric and plug-in hybrid vehicles," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 848–860, Mar. 2014.

[24] Z. Wei, J. He, and L. Cai, "Admission control and scheduling for EV charging station considering time-of-use pricing," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–5.

[25] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, J. Fürnkranz and T. Joachims, Eds., 2010, pp. 807–814.

[26] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[27] R. S. Sutton et al., "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12, S. Solla, T. Leen, and K. Müller, Eds. Cambridge, MA, USA: MIT Press, 2000, pp. 1057–1063. [Online]. Available: https://proceedings.neurips.cc/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf

[28] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, May 1992, doi: 10.1007/BF00992696.

[29] W. Kool, H. van Hoof, and M. Welling, "Attention, learn to solve routing problems!" in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–25. [Online]. Available: https://openreview.net/forum?id=ByxBFsRqYm

[30] W. Kong et al., "A new dog learns old tricks: RL finds classic optimization algorithms," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–25. [Online]. Available: https://openreview.net/forum?id=rkluJ2R9KQ

[31] H. Akhavan-Hejazi, H. Mohsenian-Rad, and A. Nejat, "Developing a test data set for electric vehicle applications in smart grid research," in *Proc. IEEE 80th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2014, pp. 1–6.

[32] *Kiasoul Features*. Accessed: Mar. 1, 2020. [Online]. Available: https://evdatabase.uk/car/1154/Kia-Soul-EV-64-kWh

[33] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1322–1328.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[35] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, Eds. Red Hook, NY, USA: Curran Associates, 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Mhd Ali Alomrani received the M.Eng. degree in electrical and computer engineering and the H.B.Sc. degree in computer science from the University of Toronto. He is currently a Machine Learning Researcher with Huawei Noah's Ark Lab, Toronto, Canada. His research interests include deep learning, graph neural networks, and applications to real-world problems.

Mosaddek Hossain Kamal Tushar (Member, IEEE) received the B.Sc. degree in applied physics and electronics and the M.Sc. degree in computer science from the University of Dhaka, Dhaka, Bangladesh, in 1993 and 1995, respectively, the master's degree in information technology from the University of New South Wales, Sydney, NSW, Australia, in 2006, and the Ph.D. degree in electrical and computer engineering from Concordia University, Montreal, QC, Canada, in 2017. From May 2017 to January 2018, he was a Post-Doctoral Fellow with the Concordia Institute for Information System Engineering, Concordia University. He is an NSERC-Awarded Post-Doctoral Fellow (PDF) with the Department of Electrical and Computer Engineering, University of Toronto, from January 2020 to June 2022. Currently, he is working as a Faculty Member with the Department of Computer Science and Engineering, University of Dhaka. His research interests include smart grid cybersecurity, electric vehicles, the IoT, energy management, network design, game theory, deep learning, and optimization. He was awarded the Prestigious ENCS Doctoral Prize for Best Research Contribution in his Ph.D. study in 2017.

Deepa Kundur (Fellow, IEEE) received the B.A.Sc., M.A.Sc., and Ph.D. degrees in electrical and computer engineering from the University of Toronto in 1993, 1995, and 1999, respectively. She is currently a Professor and the Chair of The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto. She is the author of more than 200 journals and conference papers and is a recognized authority on cybersecurity issues. Her research interests include the interface of cybersecurity, signal processing, and complex dynamical networks. She is a fellow of the Canadian Academy of Engineering and a Senior Fellow of the Massey College. She was a recipient of teaching awards at both the University of Toronto and Texas A&M University. She has served as the Honorary Chair of the 2021 IEEE Electric Power and Energy Conference and has served in numerous conference executive organization roles, including the Publicity Chair for the 2021 ICASSP, the Track Chair for the 2020 IEEE International Conference on Autonomous Systems, the General Chair for the 2018 GlobalSIP Symposium on Information Processing, Learning and Optimization for Smart Energy Infrastructures, and the TPC Co-Chair for the 2018 IEEE SmartGridComm.