Detecting State of Charge False Reporting Attacks Via Reinforcement Learning Approach

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

***Abstract*—The increased push for the adoption of green transportation has been apparent, especially in the last five years, to address the alarming increase in atmospheric CO_2 levels. The success and popularity of EVs have led many carmakers to shift to the development of clean cars in the next decade. Moreover, many countries around the globe have set EV target adoption numbers, some even aiming to ban gasoline cars by 2050. Unlike their gasoline-based counterparts, EVs comprise a myriad of sensors, communication channels, and decision-making components vulnerable to cyberattacks. Hence, the unprecedented demand for EVs calls for the development of robust defenses against these increasingly sophisticated attacks. In particular, recently proposed cyberattacks demonstrate how consumers may mislead by sending false data to unlawfully receive higher charging priorities, congest charging schedules, and steal power. In this paper, we devise 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 RL (Reinforcement Learning) 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, we show that our detection model trained with the RL-generated attacks displays greater robustness to intelligent attacks.**

***Index Terms*—Cybersecurity, Deep Learning, Reinforcement Learning, EV Charging**

I. Introduction

As the climate change catastrophe looms day by day, the electrification of the automobile market along with low carbon energy generation are seen as key approaches to reduce air pollution in densely populated cities while diversifying energy resources [1]. Such strategies have been major drivers behind recent government policies to accelerate the advancement and adoption of sustainable energy, with some guidelines aiming to phase out combustion engine vehicles by 2050 [2].

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 [3]. An EV owner can drive into a charging station and seamlessly plug their car to start charging their 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. As such bi-directional flow of energy benefits a variety of grid stakeholders [4], [5]. Several protocols have been designed to enable the two-way flow of information and energy between EVs and the energy grid [6]. 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 at all times.

Nonetheless, the enhanced integration of information and communcation systems within EVs increases the associated cyberattack 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. The leaked user data can then be used for unauthorized transactions or generating fake traffic to disrupt the grid [7]. Furthermore, existing studies have confirmed that malicious consumers can gain higher priority and steal power by reporting false data to the smart grid [8], [9], [10].

II. Literature Review of EV Security

A. Vulnerabilities in ISO 15118 and OCPP

The inherent complexity of the smart charging infrastructure gives rise to a multitude of security concerns providing the adversary with a various attack vectors through the EV, electric vehicle supply equipment (EVSE), and charging station (CS), as shown in Fig. 1. We outline some key attack methods in the ISO 15118 and OCPP (Open Charge Point Protocol) protocols to gain an advantage over benign users. We assume the attacker possesses reconnaissance capabilities and can sniff messages between communicating parties such as the EV, EVSE, and CS.

Lee et al. [11] have analyzed various vulnerabilities in the ISO 15118 protocol that exploit the EV and EVSE. For instance, user identification data stored in the RFID chip can be copied or fabricated for unauthorized charging transactions. Consequently, a malicious EV could disguise itself as another vehicle by replacing its ID with the victim's ID. The EVSE, which receives the ID with no additional authentication, can wrongfully write the billing information to the victim. Moreover, ISO 15118 is susceptible to other message modification attacks that fabricate metering data and SoC to give the malicious EV smaller bills and higher charging priority.

With the lack of vital security measures, EVs can also serve as compromised IoT devices that can initiate a DDoS attack on the CS or charging coordinator (CC) by flooding the

Mhd Ali Alomrani is a Postgraduate Engineering student at the University of Toronto. Email: mohammad.alomrani@mail.utoronto.ca

Dr. Mossadek Hossain Kamal Tushar is a Post Doctoral Fellow at ECE of the University of Toronto. Email: mhktushar@ieee.org; mosaddek.tushar@utoronto.ca

Dr. Deepa Kundur is a Professor and Chair at ECE of University of Toronto. Email: dkundur@ece.utoronto.ca

network with fake charging requests [12], [13]. Such attacks may overload the charging schedules and prevent other owners from using the grid. Moreover, compromised EVs may modify the ”charging profile” parameters to increase demand on the grid during peak load times. In such events, the smart grid will have difficulty in serving the corresponding load possibly preventing legitimate consumers from receiving power.

OCPP primarily enables public EV charging to coordinate power flow and information between the EV, CS, and the grid. In the past five years, Open Charge Alliances (OCA) introduced several OCPP versions that offer new functions such as automatic power control and monitoring. The original OCPP standard communicates all data in clear text, allowing attackers to sniff private information easily. Though OCPP developers can deploy the TLS protocol to provide encryption over links, manufacturers often choose not to include the these protocols to avoid overhead and additional costs [7]. Nonetheless, even in the presence of TLS, Alcaraz et al [14] found OCPP to be vulnerable to various distortion, disruption, and disclosure attacks. For instance, a MitM (Man-in-the-Middle) attack on the CS may affect the power grid stability. An attacker may perturb power usage or reverse power flow during off-peak times to cause unanticipated loads, potentially initiating blackouts.

B. Deep Learning for EV Security

Timely detection of such attacks via intrusion detection systems (IDS) is vital to help aggregators and owners take appropriate mitigation strategies such as shutting down malicious EVSEs or EVs. However, widely deployed state-of-art IDS such as signature-based approaches require knowledge of previous attack signatures and must be updated manually to detect new attack patterns [15]. This calls for more intelligent IDS that can generalize well to novel attack strategies. In this section, we explore the use of deep neural networks (DNNs) to flag suspicious behaviors in the smart charging infrastructure. DNNs are known for their powerful generalization capabilities, ability to learn complex tasks, and speed [16].

1) Mitigating attacks using DL: Basnet et al. [17] introduced the first novel deep learning-based IDS for detecting DoS attacks on EVSE servers. In such attacks, an attacker exploits the EVSE server to launch any combination of SYN floods, buffer overflow, or teardrop attacks to compromise the availability of the grid’s resources. By accessing the network packets throughout such attacks, the authors extract key features from the data to train feed-forward neural network (FNN) and LSTM [18] models that implicitly learn digital fingerprints of any given DoS/DDoS attack. Their results show that both the FNN and LSTM based IDS achieved more than 99% detection accuracy. Moreover, the LSTM method proved to be superior to the FNN method in metrics such as accuracy, precision, recall, and measure.

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 choose to intelligently overload charging schedules at charging stations with fake requests rather than aim to increase CPU and memory consumption of servers. Moreover, their models do not leverage EV charging information such as EV ID and previous charging history that can be tremendously helpful in identifying benign and malicious charging behavior.

Nabil et al. [10] proposed a privacy-preserving electricity theft detection scheme utilizing convolutional neural networks. They train their model on a dataset of energy consumption readings from smart meters in homes. They consider electricity theft attacks that attempt to report lower meter readings to get than actually consumed. Due to the lack of such real world attacks, the authors generate a combination of synthetic partial reduction and time filtering attacks. Using privacy-preserving methods, the authors train the CNN on masked consumption data to attain reasonable detection accuracy with a minor drop in performance.

Shafee et al. [8] devise a machine learning model to identify malicious EVs which report false SoC data to the charging coordinator. The CC schedules EV charging based on its SoC and assigns higher priority to cars with lower SoC. Therefore, such attacks, when coordinated with other malicious EVs, may congest charging schedules and overload the grid. Consequently, the authors proposed using real-world charging behavior of plug-in hybrid electric vehicles (PHEVs) combined with synthesized false reporting attacks to train a DNN with the gated recurrent unit (GRU) architecture. The GRU model can identify EVs which deviate from their benign behavior throughout a day and flag them as malicious. Their GRU model can detect deceptive vehicles with high accuracy and demonstrate good generalization abilities in detecting new attacks. However, the synthetic attacks used in [8] and [10] do not explicitly utilize additional information to predict more intelligent attack strategies. Hence, detection models trained on such attacks are not provably robust against more intelligent or stealthy attackers in practice.

Rahman et al. [9] study attacks on the plug-in electric vehicle (PEV) battery management system (BMS) via false data injection. Attacks that tamper with energy requests and usage lead to out-of-service vehicles, power grid destabilization, and battery pack damage via overcharging. They devise a neural network model that learns to estimate current battery SoC based on vital metrics such as batteries’ temperature, OCV, capacity, power, etc. The authors use Sealed Lead Acid (SLA) batteries to represent EV batteries to collect charging data. The batteries’ temperature, OCV, capacity, and power were recorded for each cycle during multiple charging and discharging sessions. 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. The authors demonstrated that the neural network (NN) could detect tampering attempts; however, their approach comes with several disadvantages. First, the NN requires a large amount fine grained EV charging data that may not be readily available. Second, the authors do not provide rigorous detection conditions. Lastly, the generated attacks are not realistic and do not provably generalize well across different battery types.

To our knowledge, 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 new and customized attack schemes. Hence, the proposed detection models are not trustworthy enough to be deployed in practice.

III. Problem Statement and Contribution

We consider a charging system composed of a CC, aggregator, and community of EVs. 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 will, in turn, forward the charging request to the CC for scheduling. The requests contain essential information for scheduling such as the battery SoC and TCC. Once the CC receives all charging requests from a certain area, the charging coordination mechanism [19] prioritizes a subset of EVs such that the total allocated power does not exceed the energy capacity C. More concretely, for each charging request i, the CC receives SoC$_i$, TCC$_i$, and the energy demand P_i to construct a priority index PI$_i$:

$$PI_i = \epsilon f_1(SoC_i) + (1-\epsilon) f_2(TCC_i) \tag{1}$$

where $0 \leq \epsilon \leq 1$, 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 PI$_i$ by its energy demand (P_i) and selects the EVs with the highest ratios for charging such that the maximum charging capacity (C) is not exceeded. Consequently, for any EV i, the CC will either allow it to charge with power rate r_i or defer the request for a future time-slot.

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 time-to-complete-charge (TCC). With the lack of intelligent detection mechanisms, a malicious EV may report false SoC data to obtain higher priority and energy allocation. Consequently, any malicious EV with little false reporting capabilities can steal power, congest charging schedules, and destabilize the grid.

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 (DoC) to benign EVs, and disrupt the load on the grid. In contrast to previous works, we use the proposed framework to create intelligent 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 lying and honest EVs. Moreover, we show that using reinforcement learning to generate potential attacks gives rise to more novel attack schemes that the detection model can learn from and compare the robustness of our model to one trained on handcrafted attacks only.

The remainder of this paper is organized as follows: In the next section (Section IV), we outline the stakeholders associated with the charging infrastructure, the communicated data, and protocols used. We also introduce the threat model considered and methodology. Dataset generation is discussed in Section V. The detection model and RL agent are presented in Sections VI and VII respectively, followed by results in section VIII. Finally, we analyze our findings in Section IX and the conclusion is drawn in Section X.

IV. System Model

A. The Charging Infrastructure

The EV charging infrastructure 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.
- Smart meter and charger: Devices that measure electricity usage and regulate the charging schedule.
- Charging station (CS): A station equipped with many EVSEs.
- Control Centre: A central management system that manages the power grid and supervises the energy requests by charging stations and EVs.

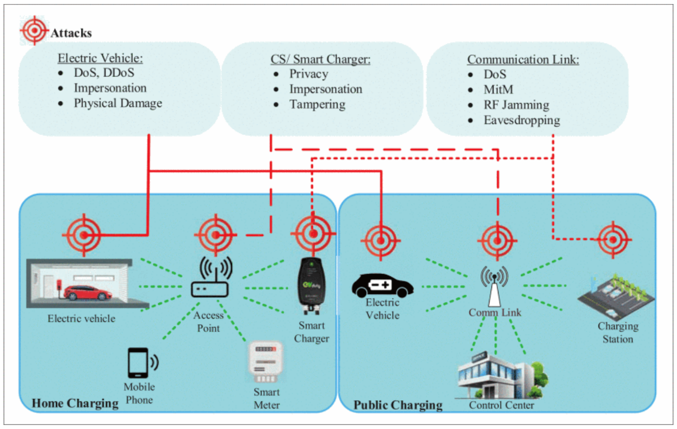


Fig. 1: EV Infrastructure Threat Models [7]

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

- The EV interacts with the CS (or control center) 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 centre exchange incoming EV data to ensure availability of EVSE.
- The control centre negotiates power usage, scheduling and pricing with the power distributor.
- 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 is used to govern all communication between EVSE, control center, and grid.

B. Threat Model

We assume the attacker aims to disrupt the proper operation of the charging coordination scheme. Hence, the attacker can intercept and modify all incoming charging requests before being sent to the CC. We focus on attacks that aim to give malicious EVs more charging power by faking their SoC values.

Various attack strategies have been proposed in [8], where an EV reports a false SoC value at day d and time t, and for EV i denoted by $RS_i(d,t)$, as shown in Table **??**. The existing research shows that such attacks disrupt the scheduling scheme; however, they can strongly deviate from normal charging behavior, making them unstealthy, 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 in that they can evade basic detection mechanisms and provide higher priority to the attacker. We use these generated attacks along with a dataset of normal behavior to train a detection model to classify an EV as honest or malicious more accurately.

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

TABLE I: SoC false reporting attacks [8].

In TABLE **??**, $S_i(d,t)$ 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 that is always between 0 and 1. In principle, the attacks randomly report a lower SoC value as charging coordinators generally give cars with low battery priority to charge.

C. Methodology

We propose using state-of-the-art FNNs to detect malicious behavior against CCs by leveraging historical data such as SoC and TCC. Our approach is similar to the one presented in [8], [10], except that we decompose training into two stages. First, we train (see Fig. 3) 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 it's attack strategy to any scheduling mechanism, as opposed to general handcrafted attacks, thus making the FNN more robust to a variety of 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 and, therefore, require little data to generate attacks. In contrast, generative models, such as GANs [20], require a dataset of attacks and can only generate samples that are as good as the dataset.

To train the adversarial agent within an RL framework, we must formulate the SoC attack problem as a Markov decision process (MDP):

- **State**: At timestep t, a state s_t consists of the incoming charging requests of all EVs, which is 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 $S_m(t)$ of the malicious vehicle m:

$$RS_m(t) = S_m(t) + a_t \tag{2}$$

- **Reward function**: The terminal reward R is defined as the sum of the power allocated to the malicious EV m 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:

$$R = \sum_{t}^{T} P_t - \gamma a_t \tag{3}$$

where P_t is the power allocated to the malicious EV at timestep t, and T is the total number of timesteps in an episode.
- **Policy**: We define a solution to be a set of actions $\pi = \{a_1, \ldots, a_n\}$ which represents the perturbations of the SoC values reported at each timestep. The policy network defines a stochastic policy $p(\pi|s)$ for selecting a solution a given the real sequence of the malicious EV's SoC values, denoted by s. It is parameterized by θ and can be factorized as:

$$p_\theta(\pi|s) = \prod_{t}^{n} p_\theta(a_t|s_t) \tag{4}$$

V. Dataset

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 PHEV taxis [21] 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 [22] 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:

$$SoC = SoC + \frac{Charging\ rate \times duration}{Battery\ Capacity} \tag{5}$$

$$SoC = SoC - \frac{Consumption\ Rate \times duration}{Battery\ Capacity} \tag{6}$$

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. 2 shows the average SoC values of 2 taxis reported over 23 days, which must be distinguished from malicious behavior. Therefore, we employ a complex model to learn the temporal behavior of the taxis and detect any malicious deviation.

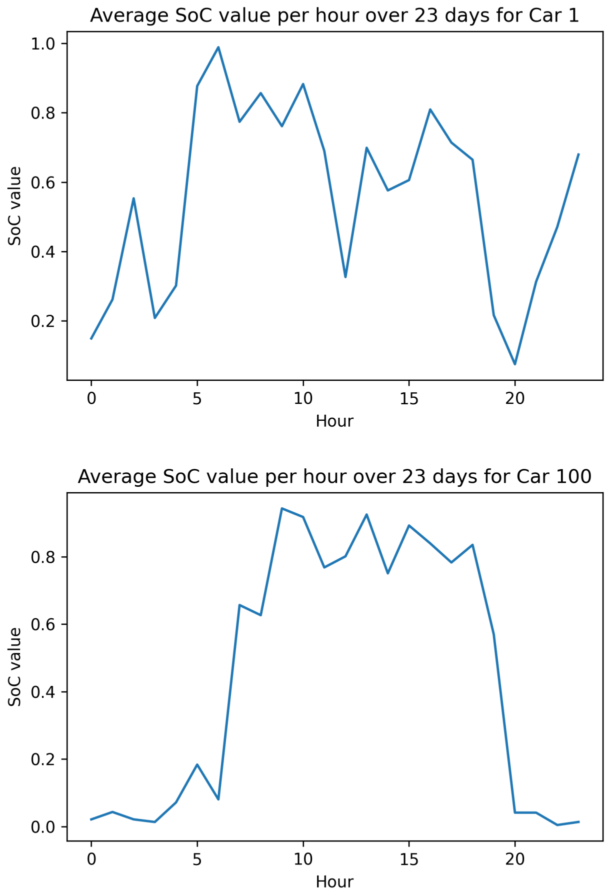


Fig. 2: Average SoC per hour of two taxis over 23 days

B. Malicious Dataset

In addition to the handcrafted attacks outlined in Table **??**, we use the trained RL agent to generate intelligent stealthy attacks from each data sample. That is, for each data sample in the benign dataset, we deploy the RL agent in the charging simulation to perturb the malicious EV's real SoC values. The resulting perturbed sequence will be labeled as a malicious data sample. To ensure that we have rich attack data, we generate different malicious samples for each benign sample. This is done by deploying the RL agent in the charging simulation multiple times, but with different random seeds each time, to get new attack samples. The ADASYN [23] method is, then, used as a data augmentation technique to balance ratio of benign samples to malicious samples. The full framework can be seen in Figure 3.

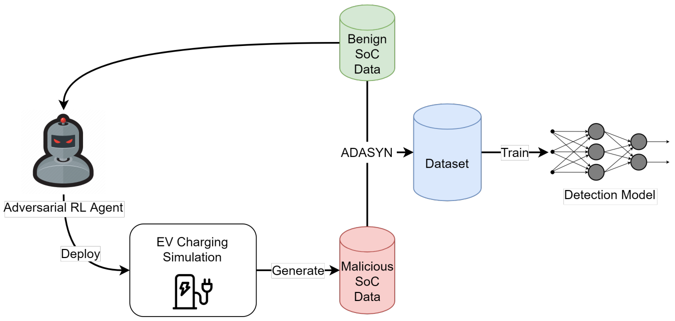


Fig. 3: The full training pipeline for the detection model. Note that the adversarial RL agent used here is already trained.

VI. 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.

A. Feed Forward Neural Network

Feed-forward neural networks (FNN) have displayed phenomenal results over the past few years for their generalization abilities and powerful learning capacity [16]. 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 [24] or Sigmoid. The successive layers of non-linearities gives the FNN powerful learning capacities. In our experiments, we use a similar detection model as in [8]. That is, our FNN is composed of 6 hidden layers of size 768 neurons with the ReLU non-linearity. The input layer receives the SoC sequence for one day $RS(d, *)$, 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 [25] layer after each hidden layer. Dropout is a technique that addresses overfitting by randomly dropping out some nodes in the network during training with probability p [1]. This prevents the nodes in a extensive network from co-adapting too much and thus gives better generalization performance.

[1] A tuned hyperparameter

VII. Adversarial RL Agent

In order 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 is presented in this section.

A. Simulation

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 will send a request to the CC with its current SoC. Once the CC receives all charging requests at time-step t, it will send back the power allocations for each charging request, with some requests receiving no power.

At the beginning of the episode, we sample a batch of benign SoC sequences from the benign training set discussed in Section V. These sequences will represent the real SoC values of the malicious vehicle for the span of the episode. The RL agent will perturb the malicious vehicle's actual SoC values at each timestep by a continuous amount $a_t \in [-1, 1]$ before being sent to the CC. The SoC values of the benign vehicles that need to charge are sampled from the uniform distribution $U[0, 1)$. All benign vehicles that do not need to charge will send SoC value 1 to the CC, so they don't get any power allocation.

In our simulation, all vehicles have a battery capacity 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 [26], [27]. We run the simulation 48 time-steps, equivalent to the length of the SoC sequences in the training set.

B. Architecture and Training

Our policy network $\pi_\theta(a|s)$ will utilize the feed forward architecture. The feed forward model will be inputted 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 [24] to introduce non-linearity. The final output layer consists of 2 neurons representing the mean μ and standard deviation σ of a normal distribution $N(a|\mu, \sigma)$. The continuous action a, which represents the value by which the malicious EV's SoC value is perturbed, is sampled from the normal distribution. This allows the policy to output a probability distribution over all possible continuous actions. We note that the policy network uses a different FNN than the detection model and is trained separately.

The action policy is trained using policy gradient reinforcement learning [28], both for its effectiveness and simplicity. That is, we learn the policy parameter θ by optimizing the loss $L(\theta|s)$ using gradient descent:

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

where $L(\pi) = -r$ with reward r. In principle, the policy gradient algorithm reinforces actions the maximize the expected outcome and discourages actions that give a low reward. To reduce gradient variance and noise, we add a baseline $b(s)$ which is the exponential moving average [29], $b(s) = M$, where M is the loss $L(\pi)$ in the first training iteration and the update step is $b(s) = \beta M + (1 - \beta)L(\pi)$ with *decay* β.

Algorithm 1 RL Agent training in Charging Simulation

```
Input: policy network $p_\theta$, charging coordinator $CC$, Total
Power capacity $C$, Number of training epochs $N$, benign
training set $D$, arrival rate $\lambda$, learning rate $\alpha$.
Initialize $numTimesteps = 48$.
Initialize $NumCars = 30$
for $i = 0$ to $N$ do
    Initialize $reward = 0$
    $S_m \leftarrow$ Sample benign SoC sequences for the malicious EV from $D$.

    for $t = 1$ to $numTimsteps$ do
        $c \sim P(\lambda)$   ▷ Number of arriving charging requests sampled from Poisson distribution
        $S_b(t) \sim U[0,1]$   ▷ Sample $c$ SoC values for benign vehicles at timestep $t$
        $R(t) \leftarrow S_b(t) \cup S_m(t)$   ▷ SoC values of all vehicles to be sent to the CC
        $\mu, \sigma = p_\theta(R(t))$
        $a_t \sim N(\mu, \sigma)$   ▷ sample action from normal distribution with parameters from policy
        $R(t) \leftarrow S_b(t) \cup (S_m(t) + a_t)$   ▷ Add perturbed SoC value for malicious vehicle
        $K = CC(R(t))$   ▷ Power allocations for each request
        $reward = reward + (K_m - \gamma a_t)$   ▷ Update reward using power allocated to malicious vehicle $K_m$
    end for
    $\theta \leftarrow \theta - \nabla L(\theta|s)$   ▷ update policy parameters based on reward
end for
```

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, one could use the RL agent to report fraudulent home power usage to disrupt the smart grid or steal power. We leave such attacks for future work.

VIII. Training Results

A. Implementation Details

The charging simulation has been implemented in Python 3.7. We use the Pytorch library [30] for training and evaluation of all deep learning models. All models are trained using the Adam optimizer [31]. 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.

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

B. RL Agent

Figure 3 shows the average reward per episode as training progresses for 120 epochs. 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.

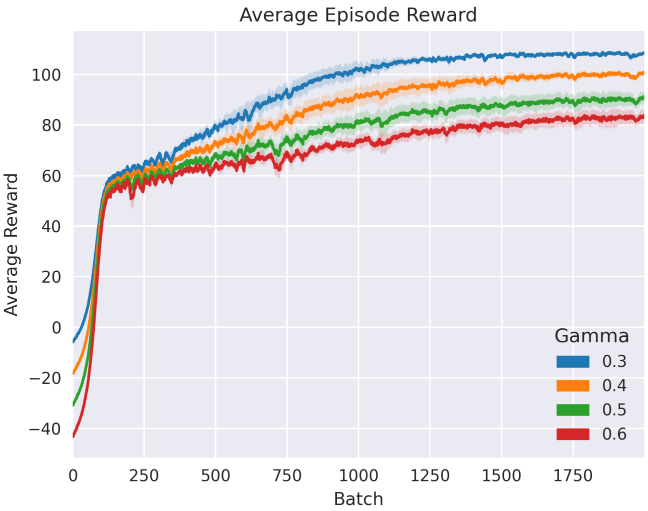


Fig. 4: Average Episode Reward throughout training for 4 RL agents. Results were averaged over 4 runs with different random seeds.

C. Detection Model

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 IV-C. 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 represent the hand-crafted samples described in Table I. γ attacks represent malicious samples generated by an RL agent trained with γ. The dataset size after ADASYN augmentation is shown in Table IV.

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

TABLE II: Attack types that each model is trained on.

We train all detection models for 200 epochs. Figure 4 shows the training results of Model 3, which is trained to detect the handcrafted attacks only. It is noteworthy that the addition of dropout layers allows the model to learn for a long period without over-fitting too much the training data, as evident in the first plot.

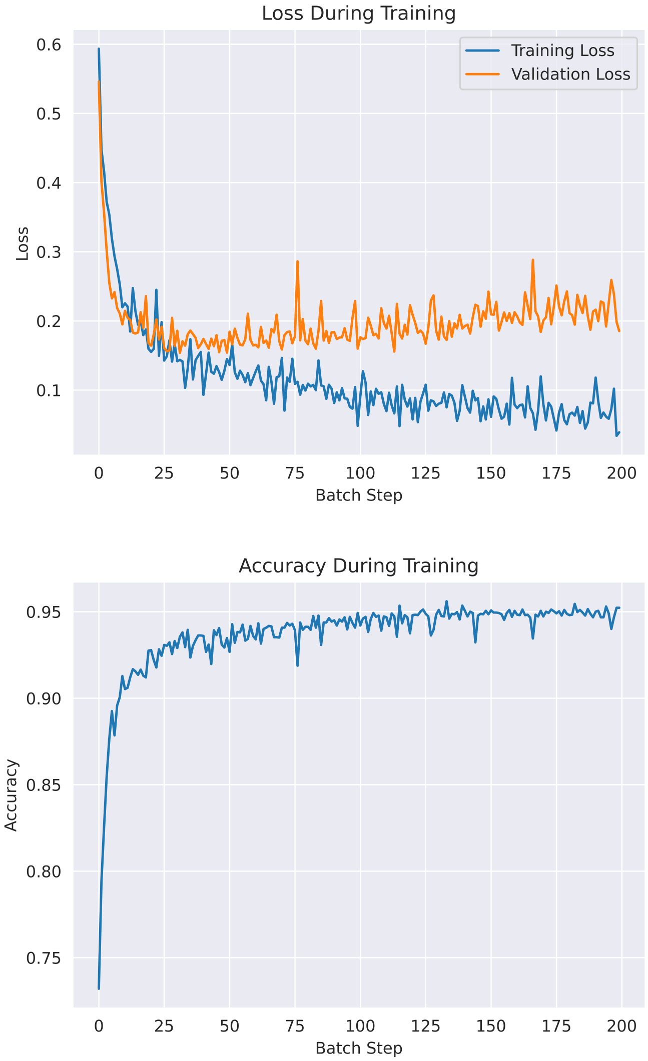


Fig. 5: Training/Validation loss for Model 3. Bottom figure displays validation accuracy throughout training.

IX. Numerical Evaluation

A. Effect of γ

In Figure 6, we visualize the SoC perturbed sequence reported by some malicious EV versus the real SoC sequence under different γ settings. For $\gamma = 0.0$, the RL agent only cares about maximizing the amount of power received across all time steps; Thus, the perturbed sequence is very stochastic compared to the actual sequence representing a real-world EV's behavior. As γ increases, the reported SoC values become closer to the valid values and therefore the SoC sequence across the day becomes more realistic.

Table III 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. For $\gamma = 0.6$, the malicious EV still is given more than 90% more power while staying close to the true values. Therefore, the RL agent can learn novel attack strategies that exploit the charging coordination mechanism while also remaining

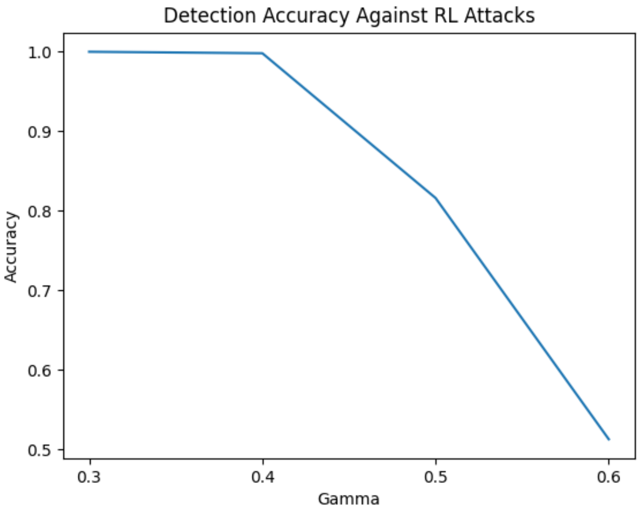


Fig. 6: Detection Accuracy of Model 1 on datasets of malicious samples generated by RL Agents with different γ.

stealthy. It is noteworthy that the CC, which is based on the knapsack algorithm, is designed to be fair and efficient to all requests. 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. 6 plots the detection accuracy of Model 1, trained on the handcrafted attacks, against the RL attacks. For $\gamma = 0.3, 0.4$, Model 1 can detect these attacks with total accuracy since they are too stochastic and resemble the handcrafted Attack 2. However, the accuracy steadily decreases with higher γ, eventually approaching a value of 0.5.

We note that Attacks 1 and 2 provide the malicious EV with approximately 10% more power than the average EV but are not as effective as the RL attacks. Interestingly, 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 displays good performance 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 due to them being completely different attack strategies. Therefore, for Model 3, we include both RL attacks and Attacks 1 and 2. We do not include Attacks 3 and 4 since such attacks are relatively ineffective and do not give the malicious EV a significant advantage. As a result, Model 3 can detect both handcrafted attacks and intelligent RL attacks with reasonable accuracy.

X. Conclusion

We developed a machine learning-based method to detect intelligent attacks against the SoC false reporting. 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 a FNN on a combination of handcrafted and RL-generated samples to obtain a more robust detection model against various attack strategies.

References

[1] I. E. Agency, *Transport Energy and CO2 : Moving towards Sustainability*, 2009. [Online]. Available: https://www.oecd-ilibrary.org/content/publication/9789264073173-en

[2] IEA, "Global ev outlook 2020," 2020. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2020

[3] Q. Wang, X. Liu *et al.*, "Smart charging for electric vehicles: A survey from the algorithmic perspective," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1500–1517, 2016.

[4] M. Mültin, "Iso 15118 as the enabler of vehicle-to-grid applications," in *2018 International Conference of Electrical and Electronic Technologies for Automotive*, 2018, pp. 1–6.

[5] C. Liu, K. T. Chau *et al.*, "Opportunities and challenges of vehicle-to-home, vehicle-to-vehicle, and vehicle-to-grid technologies," *Proceedings of the IEEE*, vol. 101, no. 11, pp. 2409–2427, 2013.

[6] M. Parchomiuk, A. Moradewicz, and H. Gawiński, "An overview of electric vehicles fast charging infrastructure," in *2019 Progress in Applied Electrical Engineering (PAEE)*, 2019, pp. 1–5.

[7] J. Antoun, M. E. Kabir *et al.*, "A detailed security assessment of the ev charging ecosystem," *IEEE Network*, vol. 34, no. 3, pp. 200–207, 2020.

[8] A. A. Shafee, M. M. Fouda *et al.*, "Detection of lying electrical vehicles in charging coordination using deep learning," *IEEE Access*, vol. 8, pp. 179 400–179 414, 2020.

[9] S. Rahman, H. Aburub *et al.*, "A study of ev bms cyber security based on neural network soc prediction," in *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T D)*, 2018, pp. 1–5.

[10] M. Nabil, M. Ismail *et al.*, "Ppetd: Privacy-preserving electricity theft detection scheme with load monitoring and billing for ami networks," *IEEE Access*, vol. 7, pp. 96 334–96 348, 2019.

[11] S. Lee, Y. Park *et al.*, "Study on analysis of security vulnerabilities and countermeasures in iso/iec 15118 based electric vehicle charging technology," in *2014 International Conference on IT Convergence and Security (ICITCS)*, 2014, pp. 1–4.

[12] S. Mousavian, M. Erol-Kantarci *et al.*, "A risk-based optimization model for electric vehicle infrastructure response to cyber attacks," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6160–6169, 2018.

[13] J. Antoun, M. E. Kabir *et al.*, "A detailed security assessment of the ev charging ecosystem," *IEEE Network*, vol. 34, no. 3, pp. 200–207, 2020.

[14] C. Alcaraz, J. Lopez, and S. Wolthusen, "Ocpp protocol: Security threats and challenges," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2452–2459, 2017.

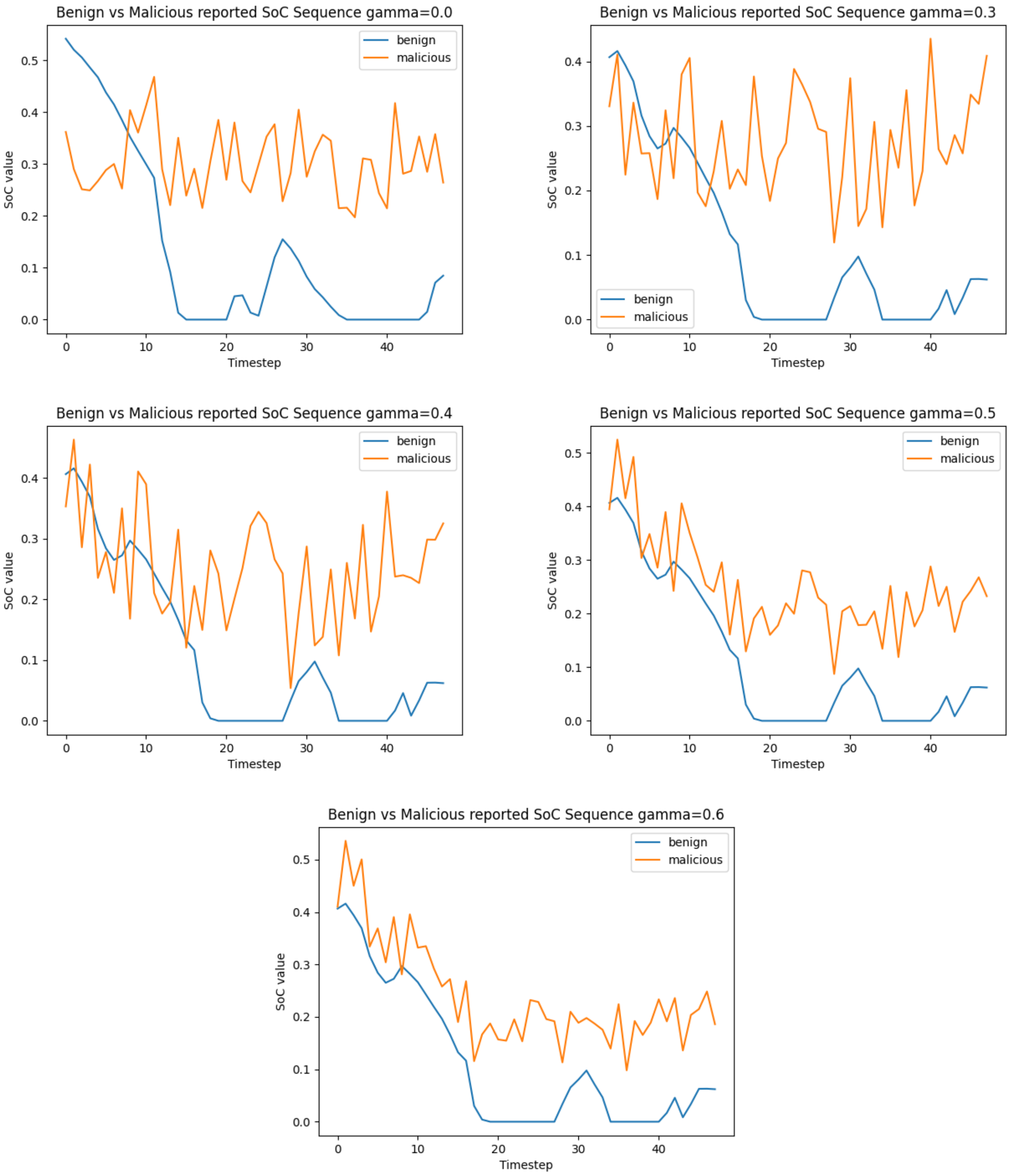
[15] A. Khraisat, I. Gondal *et al.*, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, 12 2019.

[16] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in deep learning," 2020.

[17] M. Basnet and M. Hasan Ali, "Deep learning-based intrusion detection system for electric vehicle charging station," in *2020 2nd International Conference on Smart Power Internet Energy Systems (SPIES)*, 2020, pp. 408–413.

TABLE III: Effectiveness of RL attacks on Charging Simulation and Detection Models

Attack Type	Avg. (KWh) Malicious Vehicle	Average KWh Benign Vehicle	Model 1 Accuracy	Model 2 Accuracy	Model 3 Accuracy
$\gamma = 0.3$	126.2 KWh	44.5	1.0	1.0	1.0
$\gamma = 0.4$	117.7 KWh	44.7	1.0	1.0	1.0
$\gamma = 0.5$	97.7 KWh	45.4	0.8	1.0	1.0
$\gamma = 0.6$	88.5 KWh	45.6	0.53	0.99	0.98
Attack 1	59.6 KWh	46.7	0.9	0.003	0.9
Attack 2	59.5 KWh	46.7	0.97	0.4	0.91
Attack 3	35.8 KWh	47.5	0.8	0.0	-
Attack 4	34.7 KWh	47.6	0.93	0.02	-
No Attacks	47 KWh	46.6	0.99	0.99	0.95


Benign vs Malicious reported SoC Sequence gamma=0.0

Benign vs Malicious reported SoC Sequence gamma=0.3

Benign vs Malicious reported SoC Sequence gamma=0.4

Benign vs Malicious reported SoC Sequence gamma=0.5

Benign vs Malicious reported SoC Sequence gamma=0.6

Fig. 7: Malicious Attacks generated by RL Agents

Model	Training	Validation	Test
Model 1 and 3	95,340	4000	4000
Model 2	43,350	4000	4000

TABLE IV: Dataset Split for each model

[18] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[19] M. Baza, M. Pazos-Revilla *et al.*, “Privacy-preserving and collusion-resistant charging coordination schemes for smart grids,” *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2021.

[20] I. Goodfellow, J. Pouget-Abadie *et al.*, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling *et al.*, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[21] H. Akhavan-Hejazi, H. Mohsenian-Rad, and A. Nejat, “Developing a test data set for electric vehicle applications in smart grid research,” in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, 2014, pp. 1–6.

[22] [Online]. Available: https://evdatabase.uk/car/1154/Kia-Soul-EV-64-kWh.

[23] H. He, Y. Bai *et al.*, “Adasyn: Adaptive synthetic sampling approach for imbalanced learning,” in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328.

[24] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, J. Fürnkranz and T. Joachims, Eds., 2010, pp. 807–814.

[25] N. Srivastava, G. Hinton *et al.*, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[26] M. Alizadeh, A. Scaglione *et al.*, “A scalable stochastic model for the electricity demand of electric and plug-in hybrid vehicles,” *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 848–860, 2014.

[27] Z. Wei, J. He, and L. Cai, “Admission control and scheduling for ev charging station considering time-of-use pricing,” in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1–5.

[28] R. S. Sutton, D. McAllester *et al.*, “Policy gradient methods for reinforcement learning with function approximation,” in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12. MIT Press, 2000. [Online]. Available: https://proceedings.neurips.cc/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf

[29] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Mach. Learn.*, vol. 8, no. 3–4, p. 229–256, May 1992. [Online]. Available: https://doi.org/10.1007/BF00992696

[30] A. Paszke, S. Gross *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle *et al.*, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[31] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980