Learning to Steal Electricity in Power Distribution Systems with Deep Reinforcement Learning

Osten Anderson
Department of Electrical and Computer Engineering
University of California, Riverside
Riverside, CA

Nanpeng Yu
Department of Electrical and Computer Engineering
University of California, Riverside
Riverside, CA

***Abstract*—Electricity theft detection is an area that has received a great deal of attention as proliferating smart meters create new avenues for data-driven power system monitoring and control. Very little attention, however, is being directed towards development of advanced power theft in consideration of the possibility that smart meters may be hacked. In this paper, we take a deep reinforcement learning approach to train an agent that may steal power and evade existing theft detection algorithms. The method is evaluated on a representative secondary feeder against two state-of-the-art defense mechanisms that are based on physics-informed and data-driven techniques. The numerical study results show that the RL agent beats the baseline theft strategy by stealing more power for an equivalent risk level against two strong defenders.**

***Index Terms*—Anomaly Detection, deep reinforcement learning, electricity theft, power distribution system.**

I. Introduction

In recent years, smart meters have proliferated in residential and commercial settings. This influx of data has enabled new data-driven techniques for power system monitoring and control, such as state estimation, volt-var control, and network reconfiguration [1]. However, the transition to a cyber-physical system opens new potential vulnerabilities for bad actors to attack the smart grid. Each smart meter poses a potential point at which a bad actor could access and alter the measurements relayed to the utility. In extreme cases, the aim for such an attack could be grid destabilization or equipment damage. Even in comparatively benign cases, a bad actor could alter the measurements of smart meters in order to steal power.

Considerable effort has been put into developing algorithms which can detect electricity theft, anomalous data, or false data injection attacks. However, little attention has been paid to the agents against which various algorithms are validated. Further, real data from such scenarios is typically not available. Experimental validation usually relies on synthetic data generated using potentially unrealistic assumptions. For example, electricity theft is often framed as a fixed percentage or kWh modification to the power consumption of a customer. A more advanced thief could steal with a theft profile designed to maximize stolen energy while minimizing the chance that they are detected based on network conditions. In this paper, we take the perspective of such a thief. Rather than in relation to a detection algorithm in particular, we seek to present a framework for training a thief to steal power against a defense mechanism in general using reinforcement learning.

What is needed is an agent which can simulate the actions of an advanced adversary, in order to enable the development of algorithms to prevent the actions of next-generation attacker.

The remainder of this work will be structured as follows. Section II will introduce literature in the domain of false data injection attacks and power theft. Section III will formulate the problem of optimal electricity theft and training a thief using reinforcement learning. Section IV will present a numerical study of the proposed framework. Section V will describe the conclusions.

II. Related Work

Numerous papers have been written on the topics of detecting power theft, as well as detecting anomalous data and false data injection (FDI) attacks in power systems. Over recent years, substantial effort has been directed towards the converse of this problem, studying the potential of FDI attacks to compromise grid operation.

[2] presents a vulnerability assessment of false data injection attacks on supervisory control and data acquisition (SCADA) systems, based on graph theory. In doing so, they discuss how an attacker may alter voltage data based on a power attack. [3] discusses detecting false data injection attacks, and describes an optimal attack based on minimum energy leakage. In [4], the authors discuss constructing an optimal false data injection attack with limited topology knowledge. These papers are concerned primarily with false data injection attacks, specifically directed towards state estimation and causing maximal damage to the state estimation of a network. The task of determining theft at a single meter has a substantially different construction. False data injection attacks have also been studied for the ability to mask a line outage by modifying a subset of system data [5], [6], as well as disrupting the operation of automatic generation control [7]. Generally, these approaches are physics-based. A thief may be able to access neighboring smart meter measurements, but not have access to topological information required to construct such an attack.

However, comparatively less attention is directed towards theft. The authors in [8] derive optimal attacks in the context

of electricity theft, and taking it a step further, fraud by overreporting generation by distributed energy resources. However, these attacks are derived in relation to specific algorithms, rather than a general framework agnostic to the details of the defender.

Thus, we propose reinforcement learning (RL) as a general, data-driven means of training a theft attacker. To the author's knowledge, reinforcement learning has not been used to train an adversary for theft detection. RL has however been studied in the context of disrupting grid function [9] and causing system failure [10]. Reinforcement learning has also proved highly effective at other tasks related to power systems, such as volt-var control [11] and network reconfiguration [12].

This work is somewhat related to the field of adversarial learning, although there are important distinctions. Adversarial learning typically focuses on tricking algorithms to give incorrrect output by modifying a real input by a small amount. The goal of this work is to modify the real input as much as possible without changing the output.

III. Technical Method

A. Problem Formulation

Let $m_{i,t}$ be the set of measurements collected by smart meter i in an arbitrary feeder with N_{cust} customers at time t. For ease of notation, the time index will be dropped, and unless otherwise stated, we will be referring to single time steps. Typically, smart meters measure voltage magnitude $|V|$ and real power consumed P, so let $m_i = \{P_i, |V|_i\}$ and the collection of measurements at all nodes in the feeder be M.

Consider an attack on node i consisting of a modification to the P_i, denoted by a_i. We will assume that the thief only modifies their own real power consumption measurement. Let this attack be defined by a percentage of power stolen, and thus the measured power P_i is scaled by factor $1-a_i$ to reach the altered measurement. Then $\widetilde{m_i}$ is the altered measurement set of node i, and $\widetilde{M}$ is the collection of measurements across all nodes including the action of the thief. In exchange for this attack, the thief gets a reduction in their power use on their electricity bill.

Consider an arbitrary defense mechanism or bank of defense mechanisms which takes a set of measurements M or $\widetilde{M}$ along with any requisite parameters or network topology data ρ and returns some anomaly score which quantifies the probability that power is being stolen as $D(\widetilde{M})$ or $D(\widetilde{M}, \rho)$.

Lastly, let the risk tolerance of the thief Γ denote the limit of the anomaly detection score which the thief is willing to risk. In other words, exceeding this score indicates the thief believes they might be caught, while keeping the score under this limit indicates the thief believes they will remain undetected.

A thief takes a series of theft actions indexed by time t, $a_{i,t}$, and receives an amount of free power in return. The goal of a thief can be expressed as the maximization of the amount of power stolen over time horizon T subject to the requirement that the activities of the agent remain undetected by the electric utility. In other words, the anomaly score reported by the defense mechanism is lower than Γ. As will be discussed later, the actions at one step may affect the scores of several subsequent steps, making this a sequential decision making problem.

$$\begin{aligned} \underset{\{a_{i,t}|t=1,...,T\}}{\text{maximize}} \quad & \sum_{t=0}^{T} a_{i,t} \cdot P_{i,t} \\ \text{subject to} \quad & D(\widetilde{M}_t, \rho) < \Gamma, t \in [0, T] \end{aligned} \tag{1}$$

This optimization problem would be difficult to solve, especially when the defense mechanism utilized by a utility is unknown and likely comprises more than one detection algorithm. Further, without network topology information, it may be more difficult for an attacker to construct a physics-based attack. For this reason, a data-driven method such as reinforcement learning is a compelling option for approaching this problem.

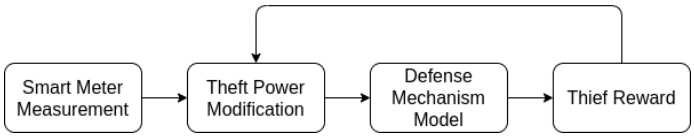

Fig. 1. Pipeline of the proposed theft agent.

B. Defense Model

Before discussing the training of an agent to steal power, we first must introduce the agent's adversary, the defense mechanism. Defense mechanisms proposed in the literature vary widely in terms of their inputs, outputs, assumptions, and limitations. The approach taken in this paper will aim to allow for as general as possible a defense mechanism.

We represent the theft mechanism as any arbitrary algorithm which maps a set of measurements $\widetilde{M}$ to a theft probability D. This theft probability will be referred to as the anomaly detection score. The setup of this environment is agnostic to detection method, as long as the method can be used to map a set of measurements to a probability of theft. The theft probability $D \in [0, 1]$ indicates the algorithms confidence that theft has occurred. A score $D = 1$ indicates the algorithm is most confident that there is theft, while $D = 0$ indicates that theft has not occurred. This also assumes some degree of linearity between these extremes, in that the algorithm should not map exclusively to very high or very low likelihoods.

Note that most methods, including those which are designed to assign theft as a boolean based on some value exceeding a threshold, can feasibly be mapped to a probability. Some algorithms make theft predictions for each customer, while others make a single prediction for the distribution network as a whole. In the former case, we will treat the largest anomaly score among all customers as D for that algorithm. The justification is that even if the thief's score does not increase as a result of their actions, if other customer's scores spike, it may set off alarm bells for the grid operator. A common approach in detection algorithms is the use of a window-based approach, where the algorithm makes a theft prediction on

multiple timesteps of measurements at once. Thus, a thief's actions may continue to affect the detection score multiple steps after the action was taken. We do not however, consider those algorithms which employ rolling windows, in which the parameters for the algorithm are continuously updated. The defense algorithms are considered fully trained before the agent begins learning how to steal electric power.

Some algorithms known in the literature are entirely data-driven [13], [14], while others employ partial or full topology information [3], [15]. We make no restrictions on which algorithms could be used. Thus an attacker with no topological information could mount an attack based only on training against data-driven detection algorithms, but an attacker with some topological information could feasibly train a more robust thief against a combination of physics-based and data-driven algorithms.

The thief has a difficult task in learning the response of the defense mechanism to attack actions, especially in the case of a defense mechanism with multiple algorithms. These assumptions, such as taking the maximum anomaly score in the case of multiple scores, not allowing continuously updating algorithms, and requiring the algorithm map to a probability of theft, are intended to simplify the agent's task, while still allowing for as many cases of defense algorithm as possible. In the numerical study, we will include defense mechanisms covering a range of these cases.

C. Formulation as Markov Decision Process

The problem of how to sequentially steal electric power is formulated as a Markov decision process (MDP). A MDP is a 4-tuple (S, A, P_a, R_a), where S represents the state space and A represents the action space. P_a is the state transition probability, in which an action from state s transitions to state s' with $Pr(s'|s, a)$, for a reward $R_a(s, s')$. The RL agent takes action a from state s according to the policy $\pi(s)$. The state-action value $Q(s, a)$ is defined as the value of taking action a from state s.

1) Action: This problem will be formulated as an episodic MDP. The thief is represented by the agent. In each step, the thief may take action a related to stealing power. We assume that the thief has knowledge of the real power and voltage magnitude measurements at each node in this system, but only modifies their own power measurement. Thus, the action space is the percentage of their load which is stolen, ranging between stealing no power and stealing their full load, $a \in [0, 1]$. In section II, we introduced an action indexed by time and the node of the thief, $a_{i,t}$. In formulating this problem as a MDP, the thief only acts on their own real power consumption measurement in the current time step. Thus, the action a is a single value. In return, the agent receives an amount of stolen power. However, the reward is a function not only of the amount of stolen power, but also the likelihood of detection, as a high rate of theft is worth relatively little to a thief if they are easily caught. This is similar to the optimization in equation 1, but with the constraint incorporated into the reward.

2) State: The reward is a function relying heavily upon the output of the defense mechanism, which in turn is primarily a function of the input P and $|V|$ measurements. Thus, the state definition should incorporate the P and $|V|$ measurements at each node, such that the agent can learn which network conditions allow it to steal more power, or which require it to steal less. The agent is also limited by its past actions. If the agent takes an aggressive theft action in a previous time step, it may need to reduce its activity over subsequent timesteps in order to evade detection. Accordingly, the state definition should also incorporate the detection score D.

Thus, a snapshot of the state for single timestep can be defined

$$\hat{S}_t = [M_t, D_{t-1}]$$

As previously discussed, algorithms frequently use more than a single time step of measurements to detect theft. The result is that not only the previous actions, but the previous measurements affect the current detection scores. Rather than only including the current network measurements, the state should include past measurements as well. With this in mind, we propose a state definition consisting of the real power and voltage magnitude measurements at each node in the system and the detection score over the last τ time steps. Thus, the agent makes a decision to steal power based on current and past measurements, and past detection scores.

$$S_t = [\hat{S}_t, \hat{S}_{t-1}, ..., \hat{S}_{t-\tau}]^T \tag{2}$$

3) Reward: In order for the agent to learn complex theft behavior, the reward function must be carefully designed. From the perspective of the thief, the best possible actions are those with high rates of theft and low chances of detection. The next best action is one with low theft rate and low chance of detection, as a cautious thief would value their safety over higher theft. The worst actions are those with high theft and high risk, and lastly low theft and high risk. While the detection score remains under the threshold, there should not be a strong penalty for increasing score. That is, to the thief's perspective, it matters little if their score is 10% or 30% if the threshold is 60% as they will not be detected in either case. However, the penalty to the reward for chance of detection should increase sharply as the score exceeds the threshold. The reward formulation is as follows:

$$R(D, a) = \frac{1}{1 + e^{p_s(D-\Gamma)}} \cdot (1 + a)^{a_f} \cdot (1 - D)^{d_f} \tag{3}$$

The parameters are choices of parameter tuning, selected to give the desired agent behavior. These values may be tuned to alter the behavior of the agent, for example by decreasing the sharpness of the penalty for detection, the agent may tend towards higher theft modes. The term Γ is the risk threshold, and relates where the reward begins to be penalized more harshly for detection, while p_s controls the slope of this penalty term. The terms a_f and d_f for the most part control the relative importance of the attack and the detection score

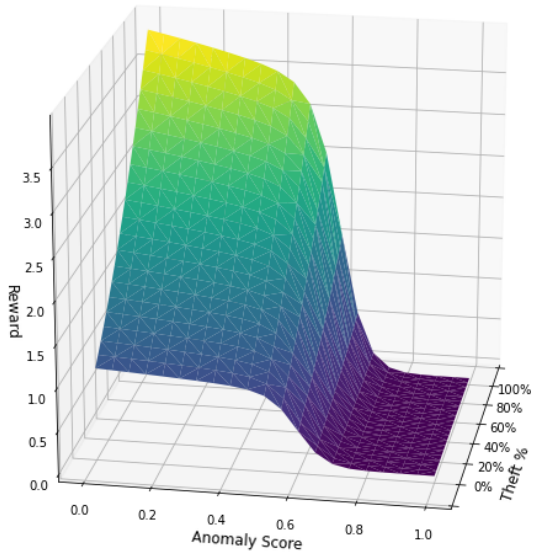

Fig. 2. Proposed reward function

in the $D < \Gamma$ regime. The reward function with $\Gamma = 0.6$, $p_s = 25$, $a_f = 2$, and $d_f = 0.2$ is shown in 2, and will be used in numerical validation.

There are two exceptions to the above reward definition. For choosing to not steal, the agent receives a fixed reward R_{fixed}. Defense mechanisms are prone to assigning a range of scores to non-theft cases, including false positives. Fixing the reward for non-theft counterbalances problems of false-positives, and encourages the agent to take the safe action when theft would likely be detected. However, this reward should be sufficiently low that the agent is encouraged to steal even small amounts of power if detection is unlikely over stealing none at all.

Lastly, it should be acknowledged that if there are regularly high probabilities of theft in a small feeder, the utility will likely investigate those customers. Thus, we define getting caught as having an theft score above threshold Γ when averaged over a window of N_{caught} most recent steps. The agent is also penalized for being caught in this way, in the form of a negative reward R_{caught}, and the episode is ended early. The magnitude of this penalty can be tuned as a hyperparameter, as a larger magnitude pushes the agent away from risky theft behaviors.

D. Reinforcement Learning Algorithm

To solve the above formulated MDP, the thief agent will be trained using a deep Q network (DQN) [16]. DQN is an off-policy, model-free algorithm for discrete action spaces. In DQN, the state-action value Q is defined by the Bellman optimality equation, where Q is the immediate reward of taking action a from state s, and the discounted future reward possible from the transition to state s', where discount factor γ balances immediate and future rewards.

$$Q(s,a) = r(s,a) + \gamma \max_{a'} Q(s',a') \tag{4}$$

DQN parametrizes the state-action function Q using a neural network. The agent then takes the action with the highest expected Q value $\pi(s_t) = \arg\max_{a \in A} Q(s_t, a)$. In order to assure the network is not updated with only the most recent action, to promote stability, updates to the neural network parameters are performed with samples from a replay buffer of past states and actions. A discrete action space is suitable for this problem as theft action can be readily represented by uniformly spaced actions between 0% and 100%.

Numerous variations on DQN have been proposed to achieve better stability or performance. We implement DQN with two variations, namely double DQN [17] and prioritized experience replay [18]. Double DQN seeks to reduce overestimation of Q values via a modified update rule. With prioritized experience replay, the replay buffer is sampled with priority given to those state transitions expected to yield larger amounts of learning, with the TD error of the last training update as the metric for the amount of learning.

Exploration is a vital component to RL training as it allows the agent to visit new state-action pairs. Rather than perturbing the greedy action or selecting a random action to explore the action space, we use parameter space exploration [19]. Exploration is carried out by perturbing the parameters of the Q network with Gaussian noise sampled once at the beginning of the episode.

Algorithm 1: Training an Agent to Steal Power with Reinforcement Learning

Input: Dataset $\mathcal{D}$, Defense mechanism $D(\widetilde{M}_t, \rho)$, ϵ-exploration definition
initialize replay buffer $\mathcal{B}$;
while *training* **do**
  Sample episode of measurement data from $\mathcal{D}$;
  Sample Gaussian noise for Q network parameters
  for $t = 1, ..., T$ **do**
    Compute action a from Q network;
    Modify power measurement with a;
    Pass modified measurement set to D;
    Calculate agent reward R;
    Move environment next time step;
    Add state transition, action and reward to $\mathcal{B}$;
    Update Q network parameters from $\mathcal{B}$;
  end
end

IV. Experimental Validation

A. Experimental Setup

The data set used to validate the proposed reinforcement learning-based energy theft approach consists of 150 days of electric power data at 30-minute intervals from the Irish Social Science Data Archive's Commission for Energy Regulation smart meter dataset [20]. These loads are modeled on a representative secondary feeder, consisting of two laterals with four customers on each. Each line in the secondary is

considered equal in construction. The line lengths are 20 feet from customers to lateral, with each segment on the lateral 5 feet in length. The per-mile impedance Z and shunt admittance Y are given below. Voltage measurements are obtained from the power flow solution. The choice of a relatively small secondary feeder is intended to give the advantage to the defender, as detecting theft in a simple 8-bus test case should be comparatively simple to a larger feeder.

$$Z = \begin{bmatrix} 1.6111 + 1.3759i & 0.2271 + 0.5344i \\ 0.2271 + 0.5344i & 1.6033 + 1.3871i \end{bmatrix} \frac{\Omega}{mile}$$

$$Y = \begin{bmatrix} 0.0000 + 0.2393i & 0.0000 - 0.0574i \\ 0.0000 - 0.0574i & 0.0000 + 0.2363i \end{bmatrix} \cdot 10^{-5} \frac{S}{mile}$$

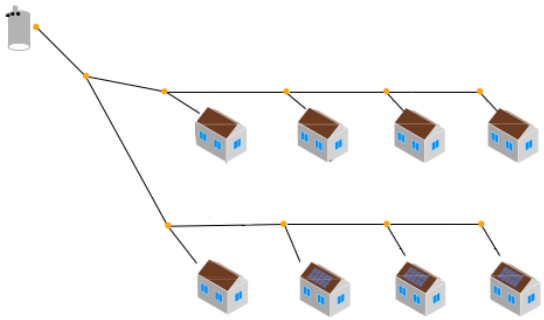

Fig. 3. Setup of the secondary feeder used for the environment.

The agent is trained on episodes of 7 days, with early stopping criterion $N_{caught} = 4$. Episodes are generated by randomly choosing a starting time from within the dataset. A validation set is reserved by picking 20 starting points throughout the dataset, evenly spaced in time. The other start times are available for training.

The chosen defense mechanism employs two state-of-the-art algorithms. The first algorithm is based on a physics-informed linear regression technique [13]. The second algorithm is a convolutional neural network (CNN)-based approach, as described in [14] but with the information-loading and meter sorting steps omitted. This choice means two of the main tracks in theft detection, the physics-informed and the data-driven, are represented. The latter algorithm uses a sigmoid output layer corresponding directly to the probability of theft D_{CNN}. The former outputs residuals, which are the differences between the estimated power consumption and reported power consumption for all customers. The authors propose using a threshold on residuals to determine theft. We map these residuals to a theft probability by first taking the ratio of the largest residual to a defined threshold $\alpha = \frac{max(residuals)}{threshold}$, then mapping this ratio α to the probability $D_{physics} = max(0, 1 - e^{\frac{1}{2} \cdot (1-\alpha)})$. Then, the detection score D is set as the mean of the scores from the two algorithms, $D = \frac{D_{CNN} + D_{physics}}{2}$.

The defense algorithms are trained on the same dataset as the RL agent. The CNN defender is trained with an 80% train-test split. The training data is arranged into windows of 8-timestep length. Synthetic theft cases for training and testing are created in 25% of windows by selecting a node at random and selecting a theft amount between 20% and 100% uniformly. The physics-based defense is trained on 40% of the data. This defender trains on clean data, but is validated on synthetic data with a $50 - 50$ split between theft and regular data, and the same theft percent distribution as the CNN defender. The CNN defense achieves 0.88 precision and 0.70 recall, and the physics-based defense achieves 1.0 precision and 0.76 recall in the test cases. Note that in both cases, the algorithms are trained towards precision over recall. In real-world operation by a utility, it is relatively expensive to investigate theft, so low precision can render the algorithm's results unfeasible to use. Both algorithms have better accuracy when theft percentages are higher.

The agent is trained with $\Gamma = 0.6$, $R_{caught} = -200$, $N_{caught} = 4$, and $R_{fixed} = 0.5$. The agent maintains measurements and detection scores for $\tau = 8$.

B. Results

The RL agent will be compared with fixed-percentage policies which are commonly used in theft detection validation. In essence, the thief reduces their load by the same percent amount in every step without regard to state. Physically, this is similar to a thief connecting some portion of their electrical load to the grid while circumventing their meter. In testing, we no longer stop episodes early for the detection condition. Early stopping would make it difficult to truly evaluate how often the agent would be caught under a policy, as the samples would be biased towards undetected examples.

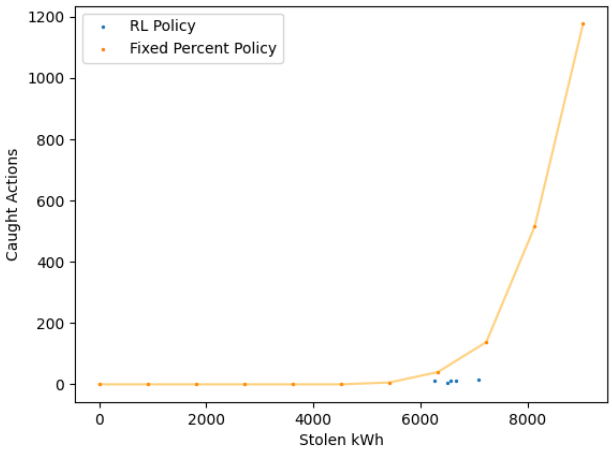

Fig. 4. Performance comparison of the fixed and the RL-based theft policy

Due to the stochastic nature of RL training, the policy which the agent converges to varies for each training run. The policies converged to range in the amount by which they outperform the baseline. Figure 4 shows a Pareto frontier for the number of actions exceeding the threshold and the total stolen electricity in kWh of the fixed-percentage policy for one-week testing episode. Five randomly-seeded RL agent's policies are shown.

The agent is able to learn a policy which, on average, allows it to steal more power than the 70% theft policy, while having a number of detection events comparable to the 60% policy.

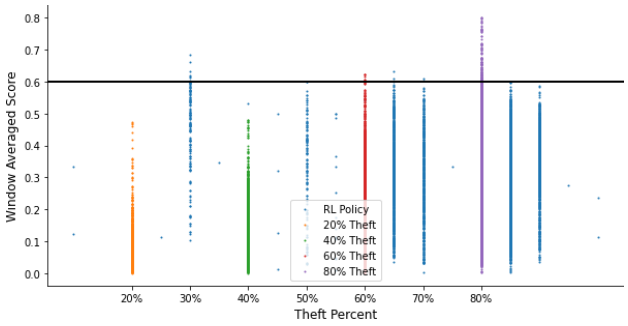

Fig. 5. Detection scores for actions taken under fixed-rate policies and the RL agent policy.

Figure 5 shows a scatter plot of the actions taken and window averaged detection scores, which are used for determining if the agent is caught during training, over the entire test set. The black horizontal line represents the threshold for detection. The learned policy is able to steal more power than the comparable 60% theft fixed policy, without having any detection events. It is notable that many actions taken by the agent are distributed at 30% theft and carry higher averaged detection scores than the 40% fixed policies. This indicates that the agent learns to lower its theft amount when the detection score rises to dangerous levels. This allows the agent to, on average, steal more power for less detection likelihood than a comparable fixed-percent policy.

In practice, a thief may take advantage of the framework proposed in this paper by training an agent offline using historical data and any number of theft detection methods, then deploying this agent in real-time to choose how they modify their smart meter measurements. Although the detection algorithms employed by utilities are likely different from the ones chosen by the thief, the attacks which can outsmart a sufficiently broad bank of detection methods in training may be able to overcome other detection methods, as has been shown similarly in the field of adversarial learning [21]. Conversely, the anomolous data detection research community may take advantage of this framework in a similar way, by training an agent with state of the art detection methods, and validating their proposed method against the attacks made by the advanced adversary.

V. Conclusion

In this paper, we proposed a deep reinforcement learning approach to training a distribution system electricity theft adversary. We introduce the goals of thief and formulate its activities as a sequential decision making problem. We design the state space, action space, and reward function of the environment that the thief lives in. We leverage DQN to train the agent to steal power while avoiding detection against two sophisticated defense algorithms. The RL agent is successful in learning a policy which allows more power to be stolen with similar or lower detection odds, while remaining undetected in all test cases.

References

[1] N. Yu, S. Shah, R. Johnson, R. Sherick, M. Hong, and K. Loparo, "Big data analytics in power distribution systems," in *2015 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, 2015.

[2] G. Hug and J. A. Giampapa, "Vulnerability assessment of AC state estimation with respect to false data injection cyber-attacks," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1362–1370, 2012.

[3] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.

[4] X. Liu, Z. Bao, D. Lu, and Z. Li, "Modeling of local false data injection attacks with reduced network information," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1686–1696, 2015.

[5] Z. Li, M. Shahidehpour, A. Alabdulwahab, and A. Abusorrah, "Analyzing locally coordinated cyber-physical attacks for undetectable line outages," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 35–47, 2018.

[6] X. Liu, Z. Li, X. Liu, and Z. Li, "Masking transmission line outages via false data injection attacks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1592–1602, 2016.

[7] R. Tan, H. H. Nguyen, E. Y. S. Foo, X. Dong, D. K. Y. Yau, Z. Kalbarczyk, R. K. Iyer, and H. B. Gooi, "Optimal false data injection attack against automatic generation control in power grids," in *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 1–10, 2016.

[8] V. B. Krishna, C. A. Gunter, and W. H. Sanders, "Evaluating detectors on optimal attack vectors that enable electricity theft and DER fraud," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 790–805, 2018.

[9] S. Paul, Z. Ni, and C. Mu, "A learning-based solution for an adversarial repeated game in cyber–physical power systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4512–4523, 2020.

[10] J. Yan, H. He, X. Zhong, and Y. Tang, "Q-learning-based vulnerability analysis of smart grid against sequential topology attacks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 200–210, 2017.

[11] W. Wang, N. Yu, J. Shi, and Y. Gao, "Volt-var control in power distribution systems with deep reinforcement learning," *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–7, 2019.

[12] Y. Gao, W. Wang, J. Shi, and N. Yu, "Batch-constrained reinforcement learning for dynamic distribution network reconfiguration," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5357–5369, 2020.

[13] Y. Gao, B. Foggo, and N. Yu, "A physically inspired data-driven model for electricity theft detection with smart meter data," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5076–5088, 2019.

[14] J. Shi, B. Foggo, and N. Yu, "Power system event identification based on deep neural network with information loading," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5622–5632, 2021.

[15] O. Anderson and N. Yu, "Distribution system bad data detection using graph signal processing," in *2021 IEEE Power Energy Society General Meeting (PESGM)*, pp. 1–5, 2021.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[17] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," 2015.

[18] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015.

[19] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz, "Parameter space noise for exploration," 2017.

[20] "CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010 [dataset]." www.ucd.ie/issda/CER-electricity, 2012.

[21] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," 2016.