Defending Smart Electrical Power Grids against Cyberattacks with Deep *Q*-Learning

Mohammadamin Moradi,[1] Yang Weng,[1] and Ying-Cheng Lai[1,2,*]

[1] *School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, Arizona 85287, USA*
[2] *Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*

(Received 6 June 2022; revised 23 September 2022; accepted 10 October 2022; published XX XX 2022)

A key to ensuring the security of smart electrical power grids is to devise and deploy effective defense strategies against cyberattacks. To achieve this goal, an essential task is to simulate and understand the dynamic interplay between the attacker and defender, for which stochastic game theory and reinforcement learning stand out as a powerful mathematical and computational framework. Existing works are based on conventional Q-learning to find the critical sections of a power grid to choose an effective defense strategy, but the methodology is only applicable to small systems. Additional issues with Q-learning are the difficulty in considering the timings of cascading failures in the reward function and deterministic modeling of the game, while attack success depends on various parameters and typically has a stochastic nature. Our solution for overcoming these difficulties is to develop a deep Q-learning-based stochastic zero-sum Nash strategy solution. We demonstrate the workings of our deep Q-learning solution using the benchmark Wood and Wollenberg 6-bus and the IEEE 30-bus systems; the latter is a relatively large-scale power-grid system that defies the conventional Q-learning approach. Comparison with alternative reinforcement learning methods provides further support for the general applicability of our deep Q-learning framework in ensuring secure operation of modern power-grid systems.

DOI: 10.1103/PRXEnergy.0.XXXXXX

I. INTRODUCTION

Electric power grids, a critical infrastructure, are vulnerable to random failures and, more alarmingly, to hostile physical and/or cyberattacks that can often trigger large-scale cascading types of breakdowns. The US-Canadian blackout in 2003 affected approximately 50 million people in eight US states and two Canadian provinces. In the same year, there were two other significant blackouts in Europe [1]. The gigantic impacted geophysical area of these events and the economic consequences highlight the need for developing effective defense strategies against attacks on the power grids. In the past two decades, research on cybersecurity systems has attracted increasing attention. An important requirement is to make these systems automated and "intelligent," as many power grids are unmanned and located in isolated, remote, rural, or mountainous areas [2]. In the field of cyberphysical systems and security, the year 2010 was a turning point, when the first ever cyberwarfare weapon, known as Stuxnet [3], was created. Documented significant events of cyberattacks include a synchronized and coordinated attack in December 2015, which compromised three Ukrainian regional electric power distribution companies and resulted in power outages affecting approximately 225 000 customers for several hours [4]. Due to the extraordinarily large scale and complexity of the power-grid networks, developing effective defense strategies against attacks to prevent breakdown of the networks has become one of the most challenging problems of interdisciplinary research in science and engineering in the present time. In this regard, a pioneering approach is to use state estimation to detect the attack modes to power systems [5,6], assuming that the topology and parameters are known to both the attacker and defender in the transmission grid. Recently, this approach was extended to the distribution grid [7,8]. It is also recognized that attacks are possible, even if the attackers do not know the topology and parameters of the distribution grid [9].

From a general and mathematical point of view, cybersecurity is determined by the dynamic interplay between the attacker and the defender, where the former seeks to maximize, while the latter strives to minimize, damage to the power grid. Game theory [10], a well-established branch of mathematics for analyzing strategic interactions among rational players, thus represents a powerful

*Ying-Cheng.Lai@asu.edu

tool to probe the dynamics of cybersecurity, where the attacker-defender interactions can be modeled as a noncooperative game. There are two categories of such games: static and dynamic. In a static game, time and information do not affect the action choice of the players, so the game can be regarded as a one-shot process, in which the players take their actions only once. In contrast, in a dynamic game [11], the players have some information about each other's choices and can act more than once, where time plays a central role in the decision-making. Different game-theoretic techniques have been devised to study the security of smart grids, such as the network formation game technique used in smart grid communications systems, the Nash game and auction game methods in demand-side management applications, and coalition games used in microgrid distribution networks [12].

Recently, machine learning has been introduced to study the security of smart power grids. For example, in Ref. [13], the most vulnerable areas in a power grid are identified using unsupervised learning. Several state-of-the-art machine-learning techniques have been devised to generate, detect, and mitigate cyberattacks in smart grids [14]. As one of the most developed machine-learning frameworks, reinforcement learning (RL) has proven to be particularly useful for cybersecurity systems. Specifically, RL is employed to derive false data injection attack policies against automatic voltage control systems in power grids [15]. In Ref. [16], a RL-based strategy was introduced that aimed to choose the appropriate detection interval and the number of CPUs allocated based on the defense preferences through implementation inside the control center of the power grid. Moreover, Q-learning [17] is used to analyze the vulnerability of smart grids against sequential topological attacks, where the attacker can use Q-learning to worsen the damage of sequential topology attacks toward system failures with the least effort [18]. A fundamental difficulty with Q-learning is that it can become extremely inefficient in the case of increasing numbers of state-action pairs, as in a larger power grid. To overcome this difficulty, deep RL has been employed in large-scale power grids for topology attacks [19]; cyberattack mitigation [20]; and, more recently, to solve the latency cyberattack detection problem [21]. In general, deep Q-learning [22] uses neural networks to approximate the Q function using only the state as the input and generate the Q values of all actions as the output. As a result, deep Q-learning is suited to problems with a large state-action space, since it leverages the extent of deep neural networks to deal with complex cyberphysical systems, such as the IEEE 30-bus system. Figure 1 provides a schematic comparison of Q-learning and deep Q-learning. Q4

Here, we develop a deep Q-learning-based defense strategy for smart power-grid systems using transmission line outages and generation loss as the concrete failure settings. Broadly, we conceive the scenario in which the defense management of a given large power grid performs stochastic game playing to simulate the dynamic interplay between the attacker and the defender. The goal is to uncover the "best" attack strategies that can result in the maximal damage to the grid. Accordingly, protecting the components in the grid that such attack strategies entail provides the optimal defense tactics. We model the attacker-defender interaction as a zero-sum game and solve it by using deep Q-learning, where solving a game entails finding its Nash equilibria (see Sec. II B for details). We introduce a customized reward function for achieving the desired objectives as directly as possible. Importantly, we demonstrate that our deep Q-learning framework can be used to address problems of cascading failures and timing delays, which, to the best of our knowledge, have not been studied previously in the context of machine-learning-enhanced or guaranteed security of power grids. Our defense algorithm leads to the best protection sets based on the defined objectives, taking into consideration the defender's policy. To demonstrate the workings and advantages of our deep Q-learning scheme, we compare its performance not only with the conventional Q-learning method but also with other state-of-the-art algorithms, such as actor-critic (AC), policy gradient (PG), and proximal policy optimization (PPO). Overall, our deep

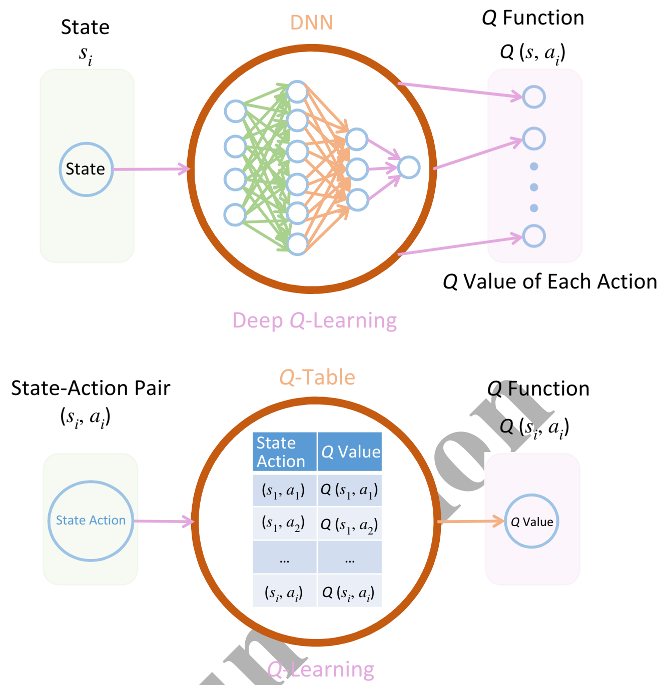

FIG. 1. Q-Learning versus deep Q-learning. Implementation of the Q table is the main difference between Q-learning and deep Q-learning. Instead of mapping a state-action pair to a Q value using the Q table, as is done in Q-learning, deep Q-learning uses neural networks to map the states to the action-Q value pairs—the core reason that deep Q-learning can be used to solve large-scale problems. Q5

Q-learning approach opens the door to applying RL to large-scale smart grid cybersecurity problems to significantly enhance the security of the system in an automated manner.

The rest of this paper is organized as follows. The RL formulation of the attacker-defender stochastic zero-sum game, problem description, reward function definition, and an illustration of why Q-learning is not viable for large-scale problems are given in Sec. II. In Sec. III, we formulate our deep Q-learning method and present the optimal defense strategy. Simulation scenarios and comparative results are detailed in Sec. IV. Section V presents a discussion.

II. REINFORCEMENT-LEARNING-BASED FORMULATION OF ATTACKER-DEFENDER GAME

We describe the essential quantities needed for modeling the attacker-defender interactions using a stochastic zero-sum game and Q-learning algorithm. We then define the reward function based on the objectives of the attack scenarios. The efficiencies of Q-learning and deep Q-learning are compared using an illustrative example. In the formulation below, player 1 is the attacker, while player 2 is the defender.

A. Attacker-defender stochastic zero-sum game and Nash equilibrium

A game is closely related to a Markov decision process that can be viewed as a single-player decision problem, so its extension to two players results in a stochastic game [23]. Mathematically, a *two-player stochastic zero-sum game* is a sextuple $\langle S, A^1, A^2, r^1, r^2, p\rangle$, where S is the discrete state space, A^i is the discrete action space of player i (for $i = 1, 2$), $r^i{:}S \times A^1 \times A^2 \to \mathbb{R}$ is the payoff function for player i, whereas $r^1(s, a^1, a^2) = -r^2(s, a^1, a^2)$ for all $s \in S, a^1 \in A^1, a^2 \in A^2$. For the cases studied in this work, intuitively, rewards are the game payoffs that are either the generation loss caused by the attacks or a function of the transmission line outages [cf., Eq. (10) below]. Moreover, $p : S \times A^1 \times A^2 \to \Delta(S)$ is the transition probability mapping, with $\Delta(S)$ being the set of probability distributions over the state space, S. During a game, player 1 aims to maximize, but player 2 strives to minimize, the sum of the discounted rewards. Given an initial state s, discount factor γ, and π^1 and π^2 (the strategies of players 1 and 2, respectively), the values of the game for the two players are

$$v^1(s, \pi^1, \pi^2) = \sum_{t=0}^{\infty} \gamma^t\, \mathbb{E}\{r_t^1 | \pi^1, \pi^2, s_0 = s\}, \qquad (1)$$

$$v^2(s, \pi^1, \pi^2) = \sum_{t=0}^{\infty} \gamma^t\, \mathbb{E}\{r_t^2 | \pi^1, \pi^2, s_0 = s\}, \qquad (2)$$

where $\pi^{1,2} = (\pi_0^{1,2}, \ldots, \pi_t^{1,2}, \ldots)$, with $\pi_t^{1,2}$ denoting the decision rules of players 1 and 2 at time t and $\mathbb{E}\{.\}$ is the conditional expectation. For instance, $\mathbb{E}\{r_t^i | \pi^1, \pi^2, s_0 = s\}$ is the expectation of the player i's instant reward at time t, following the decision rules $\pi^{1,2}$ with s as the initial state. These strategies are "stationary," in the sense that the decision rules are fixed over time, in contrast to the "behavior" strategies often used in economics, where the decision rules depend on the history of states and the actions up to the present time. Assuming each player has complete information about the reward function of the other player, a Nash equilibrium can emerge. Specifically, *the Nash equilibrium for a two-player stochastic zero-sum game* is a pair of strategies, (π_*^1, π_*^2), such that for all $s \in S$, the following hold:

$$v^1(s, \pi_*^1, \pi_*^2) \quad \geq v^1(s, \pi^1, \pi_*^2) \quad \forall \pi^1 \in \Pi^1, \qquad (3)$$

$$v^2(s, \pi_*^1, \pi_*^2) \quad \geq v^2(s, \pi_*^1, \pi^2) \quad \forall \pi^2 \in \Pi^2, \qquad (4)$$

where Π^i is the set of all possible policies for player i. Intuitively, a Nash equilibrium means that each player's strategy is the best response to the other player's strategy: neither one has anything to gain by changing only their own strategy.

In general, based on the structure of the information that the players possess, attacker-defender stochastic zero-sum games can be classified into four categories, depending on whether the information is complete or incomplete, perfect or imperfect. In particular, in a complete information game, the players know the structure of the game being played, such as the number of players and their payoff functions. Any missing information will lead to an incomplete information game. In addition, a game is regarded as being of the perfect information type if all the players know the historical actions of each other at the time of their move; otherwise, the game is of the imperfect information type [24]. For simplicity, in our work, we assume both the attacker and defender can observe each other's immediate reward and have access to their actions throughout the learning process. This assumption, while ideal and offering mathematical convenience, is based on the consideration that the goal of our work is to solve the attacker-defender stochastic zero-sum game for defensive planning. In fact, our aim is to find the best scenario for the attacker, so that the defender can be prepared for the worst, and thus, assuming the availability of complete information may not be unreasonable. Possible scenarios to obtain the required information include the observation of the state of the transmission lines by the defender, the defender's access to the resulting generation loss when an attack happens, and some insider information about the defender obtained by the attacker.

B. Q-Learning-based solution to attacker-defender stochastic zero-sum game

Reinforcement learning belongs to the field of decision-making, where the "agent" explores the "environment," interacts with it, and observes its reactions to find an optimal behavior to maximize a long-term "reward." Contrary to supervised learning, in RL, the agent must act independently to find an optimal sequence of actions that maximizes a given reward function in an unknown environment.

While RL is capable of directly solving certain cybersecurity problems, it can also serve as a powerful vehicle to gain insights into the attacker-defender interactions modeled as a game. In general, solving a game means finding its Nash equilibria. Especially, an appealing feature of RL is that it can yield solutions (Nash equilibria) of both the attacker-defender interplay and cybersecurity in a knowledge-free manner, i.e., based solely on data. For example, the Nash equilibrium for the two-player zero-sum game can be determined online based on RL [25]. RL has also been employed to solve a zero-sum stochastic game [26]. The min-max solutions of a dynamic Markov zero-sum game are derived using Q-learning [27], yielding optimal risk management strategies to meet the performance criteria with the parameters of the Markov game model completely unknown. A distributed RL algorithm is proposed to solve a non-zero-sum stochastic game in which each player needs only minimal information about the other player [28]. RL is also used in a stochastic adversarial game coupled with an expert advice framework to analyze the optimal attack strategies against predictors [29]. While game theory has been applied to many problems that require rational decision-making, there are some limitations in applying such methods to security games. Q-Learning was introduced to secure the system by devising proper actions against the adversarial behavior of a suspicious user [30]. Q-Learning has also been employed in solving security games, as studied in Refs. [31,32].

In Q-learning, the Q function is a mapping of all possible state-action pairs (where actions refer to action profiles of each player) to a scalar value and represents the total discounted reward that a player is expected to obtain, starting from a determined state taking a specified action. For a two-player stochastic game, the optimal Q function for each player can be defined as

$$Q^1_*(s,a^1,a^2) = r^1(s,a^1,a^2) + \gamma \sum_{s'=1}^{N} p(s'|s,a^1,a^2)v^1(s',\pi^1,\pi^2), \quad (5)$$

$$Q^2_*(s,a^1,a^2) = r^2(s,a^1,a^2) + \gamma \sum_{s'=1}^{N} p(s'|s,a^1,a^2)v^2(s',\pi^1,\pi^2), \quad (6)$$

where s' is the next state evolving from state s taking actions a^1 and a^2. Equations (5) and (6) define Q_*, the optimal value of the Q function associated with state s and action pair (a^1,a^2). For each player, the optimal value is equal to the total discounted reward received by the player, when both the attacker and defender perform actions (a^1,a^2) in state s and subsequently follow their Nash equilibrium strategies (π^1,π^2). For each player, the value of Q_* can be solved [Eq. (8)]. A player then generates a policy by following the action with the largest Q value in each state.

We remark that, in the reinforcement learning literature, the notation r is usually reserved for "instant reward" or "instant payoff," whereas v is the "value function." In Eq. (5), the term $r^1(s,a^1,a^2)$ means the instant payoff that player 1 gets when the game is in state s and player 1 chooses action a^1 while player 2 selects action a^2. The quantity $v^1(s',\pi^1,\pi^2)$ denotes the total discounted payoff starting from the next state s' while the players follow the policies π^1 and π^2. Thus, $Q^1_*(s,a^1,a^2)$ in Eq. (5) represents the instant reward added to the best possible future rewards for player 1. Intuitively, this means the best reward player 1 can achieve starting from state s with the two players taking actions a^1 and a^2, respectively.

Because of the zero-sum nature of the game, $Q^1_*(s,a^1,a^2) + Q^2_*(s,a^1,a^2) = 0$, or

$$Q^1_*(s,a^1,a^2) = -Q^2_*(s,a^1,a^2), \quad (7)$$

the learning agent needs to learn (or approximate) only one Q function. This should be contrasted with a general sum game characterized by $Q^1_*(s,a^1,a^2) + Q^2_*(s,a^1,a^2) \neq 0$, where two Q functions need to be learned, increasing substantially the computation complexity. To solve Eqs. (5) and (6), we use the following algorithm [23]:

$$Q_{t+1}(s,a^1,a^2) = (1-\alpha_t)Q_t(s,a^1,a^2) + \alpha_t\left[r_t + \gamma \max_{\pi^1(s')\in\sigma(A^1)} \min_{\pi^2(s')\in\sigma(A^2)} \pi^1(s')Q_t(s')\pi^2(s')\right], \quad (8)$$

where $Q_{t+1}(s,a^1,a^2) = Q^1_{t+1}(s,a^1,a^2)$. Convergence requires that all state-action pairs be visited infinitely often, which is practically infeasible. To obtain a reasonable functional approximation, a sufficiently large state-action space needs to be explored. This is the main reason that prevents Q-learning from being applicable to large-scale smart grids.

C. Transmission line outage, generation loss, and reward functions

We focus on two representative attack scenarios on smart power grids [33–35]. The first is the switching line problem, where the attacker attempts to cause a predetermined percentage of the transmission lines to go down. In the second scenario, the attacker attempts to maximize the generation loss in the power system through a sequence of attacks. In both cases, the defender strives to mitigate the attack consequences, regardless of whether they are due to transmission line outages or are caused by generation loss. [We use a dc load flow simulator of cascading (separation) in power systems, named DCSIMSEP [33,34], to calculate the generation loss.] The state space for both attacks is the state of transmission lines denoted as a $l \times 1$ binary-valued vector, where l is the number of transmission lines; this value for each transmission line is 0 if the respective line is down and is 1 otherwise. The attacker's actions for both attacks are chosen from the set $A = \{1, 2, 3, \ldots, l\}$, where action i means attacking transmission line i. The defender's action for both attacks is considered to be a set consisting of n transmission lines, denoted as the protection set. The attacker's reward for the line switching attack is given by Eq. (10) and for the generation loss attack is the average generation loss [Eq. (9)] caused by the attack. Since the game is considered to be zero sum, for the defender, the payoff is the negative of the attacker's reward for both attacks. The transition probability distribution is represented with power-grid transitions simulated with the DCSIMSEP tool.

We incorporate the cascading failure timing into the reward function. We assume that the attacker's next attack will be launched at time $T = 1.2t_{\text{cas}}$, where t_{cas} is the cascading failure length caused by the attacks. The proportional constant 1.2 is chosen somewhat arbitrarily, insofar as it is greater than 1, so that the system settles into a steady state after an attack on the transmission lines. The choice of the value T does not have a significant effect because the generation loss is relative among different attacks and our goal is to minimize the total loss. To take into account the timing delays of the cascading failures, we use a weighted average of generation loss during a reasonable time interval. Specifically, the average generation loss $\bar{G}_{\text{loss}}$ is

$$\bar{G}_{\text{loss}} = G_{\text{loss}}^{\text{init}} \frac{t_{\text{cas}}}{T} + G_{\text{loss}}^{\text{stead}} \frac{T - t_{\text{cas}}}{T}, \tag{9}$$

where $G_{\text{loss}}^{\text{init}}$ is the generation loss caused initially by the attack, while $G_{\text{loss}}^{\text{stead}}$ represents the generation loss during the steady state of the system after a transient phase caused by the attack. The reason is that, after an attack, the power grid will enter into a transient state, during which cascading failures occur. We assume that the defender's policy is passive while the attacker's policy evolves according to deep Q-learning (as described in Sec. II D). The defender's protection set is updated at the end of each run, meaning that the attacker must learn the optimal sequences in a constantly updated environment. In general, the defender is not able to protect all lines simultaneously because of limited resources. This highlights the need for Q-learning because the defender should wisely select the set of lines to protect.

For the first attack scenario, the reward function is given by

$$\begin{aligned} r &= r_1, \quad \text{for IO} > \text{AO}, \\ r &= r_2, \quad \text{if attack is final}, \\ r &= \text{IO}/\text{AO}, \quad \text{otherwise}, \end{aligned} \tag{10}$$

where IO is the instant number of transmission line outages caused by the attack, AO is the attack objective, and $r_1 > r_2$. For example, in the Wood and Wollenberg (W&W) 6-bus system shown in Fig. 2, when the protection set consists of lines 1 and 2, attacking line 5 will cause an instant outage of five lines (IO = 5), which is more than the attack objective (AO = 4). In this case, the reward of attacking line 5 is equal to r_1. This is the best scenario, and therefore, r_1 is chosen to be large enough to persuade the agent to learn this action, if possible. This will also lead to $G_{\text{loss}}^{\text{init}} = 210$ MW and $G_{\text{loss}}^{\text{stead}} = 83.5$ MW, and the cascading failure length is $t_{\text{cas}} = 331.61$ s. The cascading failure timing delays caused by attacking line 5 in the W&W 6-bus system are illustrated in Fig. 3. Equation (9) provides the average generation loss, taking into account the timing delay of cascading failures as $\bar{G}_{\text{loss}} = 167.83$ MW. Likewise, attacking line 3 will cause lines 1, 2, and 3 to go down, leading to the reward $r = 3/4$. Eventually, if the number of currently downed transmission lines is less than AO, but an attack causes the number of downed lines to be equal to or larger than AO, the attacker will have achieved the objective in this specific step, executing the chosen action. In this case, the attack is called final and the reward is r_2, as the attacking agent is motivated to take the final blow when an opportunity rises.

D. Necessity of deep Q-learning

A standard way to implement Q-learning is through the sample base variant called "tabular Q-learning." In a Q table, the rows list the states of the underlying system, and the columns are indexed by the action set. Training the table is helpful in finding an optimal action for each state with the goal of maximizing the long-term reward. This is a straightforward yet powerful approach to the security of small cyberphysical systems. For example, a one-shot game with a multiline switching attack between the attacker and defender in a smart grid was studied [36]. In another work [37], the dynamics of the electric power grid were taken into account and the attacks were modeled as a multistage game, where the percentage of visited states with respect to the total number of states was 1.81% for the W&W 6-bus system (37 states out of

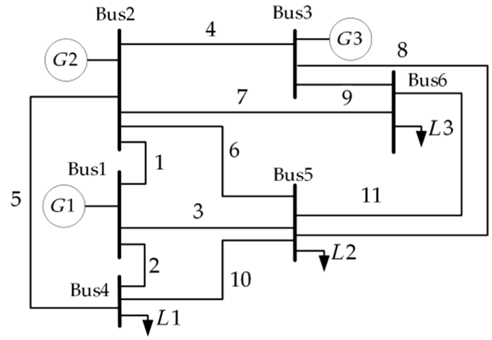

FIG. 2. Wood and Wollenberg 6-bus system. It has 6 buses, 3 generators (denoted by G), 3 loads (denoted by L), and 11 transmission lines. IEEE 30-bus system simulated in this paper has a similar topological structure but at a much larger scale: it has 6 generators, 30 buses, and 41 transmission lines. Simulation of the smart power grids (they are "smart" because they support renewable sources) is performed using the DCSIMSEP package, a simulator of cascading failures in power systems. DCSIMSEP does not use any specific stress-mitigating controls under the assumption that the cascades are propagating too fast for the operators to react, so it is suitable for cyberattack problems.

a possible 2^{11} states) and $1.87 \times 10^{-8}\%$ for the IEEE 39-bus system (13 130 states out of a possible 2^{46} states). The tabular Q-learning method is thus incapable of sufficient state-space exploration, leading to suboptimal policies for the given reward functions. In general, for larger power-grid systems, such as the benchmark IEEE 30-bus system that has 41 transmission lines, tabular Q-learning is impractical. This is because each line has two states, operational or out of service, so there are 2^{41} number of states for all the transmission lines. If only a single line is attacked, the total number of actions is 41. Because there are 2^{41} states for each action, the table will have $2^{41} \times 41$ cells, rendering infeasible any computation based on the table.

To appreciate the necessity of adopting deep Q-learning in tackling the cybersecurity problem of smart power-grid systems in a concrete way, we use the switching line problem as a prototypical example. For the W&W 6-bus system, consider the specific formulation in which AO is 4, the protection set is $[1, 2]$, the maximum number of attacks is 4, and the reward function is given by Eq. (10) with $r_1 = 4$ and $r_2 = 1$. The optimal attacking sequence derived using Q-learning after 20 independent runs (each with 2000 episodes) is to attack line 5, which will lead to a maximum reward of 4. However, the optimal attacking sequence derived using deep Q-learning is to attack line 9, then line 8, and finally line 6. In particular, the outage of line 9 will lead to reward $r = 0.25$; attacking line 8 will bring down lines 8 and 4 together, so the reward is $r = 0.5$; and disabling line 6 will cause lines 1, 2, 3, 6, 10, and 11 to go down, leading to the reward $r = 4$. As a result, the deep Q-learning strategy will result in a total reward of 4.75. A detailed comparison of the rewards achieved as a function of time from executing the optimal attack strategies from Q-learning and deep Q-learning is shown in Fig. 4. It can be seen that, while there is a brief time period (between 200 and 500 episodes of the game) in which the reward of Q-learning is greater than that of deep Q-learning, after 500 episodes, deep Q-learning leads to a persistently higher reward than Q-learning.

The main reason that the tabular Q-learning results in lower reward in the long run lies in insufficient state-space exploration, generating a suboptimal policy for the defined reward function. In a larger power grid, such as the IEEE 30-bus system that has 41 transmission lines, there are 2^{41} distinct states. Practically, a state space of this large size cannot be solved using conventional tabular Q-learning

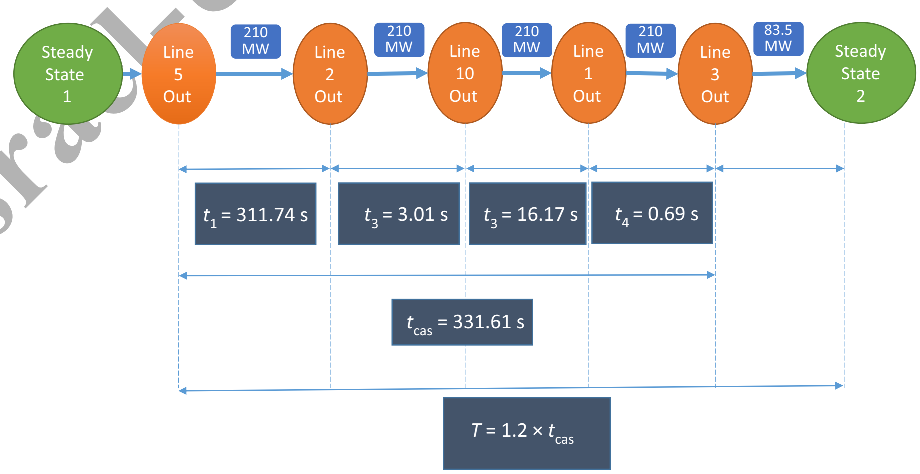

FIG. 3. Cascading failure timing delays caused by attacking line 5 in the W&W 6-bus system derived using DCSIMSEP package. Average generation loss ($\bar{G}_{\text{loss}}$) caused by this attack can be calculated using these timings in Eq. (9).

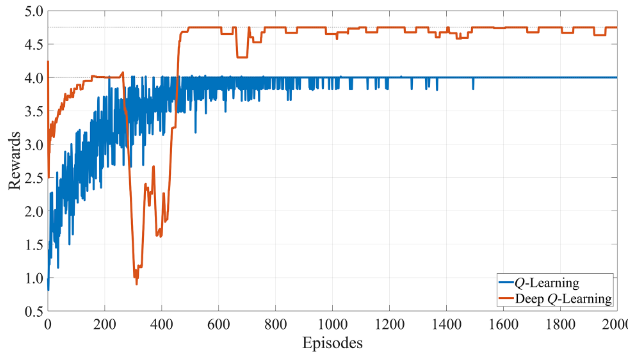

FIG. 4. Comparison of the performance of deep Q-learning and conventional tabular Q-learning using a concrete example. Setting is the switching line problem in the W&W 6-bus system. Shown are the values of reward function [Eq. (10) with $r_1 = 4$ and $r_2 = 1$] from deep Q-learning and conventional Q-learning with similar simulation parameter values. Deep Q-learning algorithm manages to find an optimal attack sequence, which results in the reward of $r = 4.75$, while conventional Q-learning is unable to find a sequence with a reward of larger than $r = 4$.

[38]. This difficulty with Q-learning is fundamental. As the system becomes larger, the deficiency of Q-learning will become more apparent and pronounced. To address the cyberattack and defense problem for large-scale power grids, invoking deep Q-learning is necessary.

III. DEEP Q-LEARNING-BASED FORMULATION OF ATTACKER-DEFENDER GAME

We introduce the deep Q-learning algorithm and exploit it to formulate and solve the attacker-defender stochastic zero-sum game problem. We also analyze the proposed defense strategy for smart power grids against cyberattacks. The zero-sum nature of the game dynamics stipulates that the deep Q-learning agent needs to learn (or approximate) only one Q function. It should be noted that, mathematically, convergence to a Nash equilibrium requires that all state-action pairs be visited infinitely often, which is practically infeasible. To obtain a reasonable functional approximation, a sufficiently large state-action space needs to be explored, which can be accomplished by deep Q-learning.

A. Deep Q-learning solution to attacker-defender stochastic zero-sum game

The core of deep Q-learning is an online multilayered neural network [39] that for any given state s outputs a vector of action values $Q(s, ., .; \theta)$, where θ denotes the set of parameters of the online network. Two foundations of the deep Q-learning algorithm are the target network and the use of experience replay. The target network, with parameter set θ^*, is the same as the online network, except that, for every c episodes, its parameters are copied from the online network, $\theta_t^* = \theta_t$, which are kept fixed during the c episodes. The target used by deep Q-learning can be described as

$$Q_t^* = r_{t+1} + \gamma \max_a Q_t(s_{t+1}, a^1, a^2; \theta_t^*). \tag{11}$$

The deep Q-learning agent gets the initial state and computes the Q-function values for all possible actions, which in our problem is the transmission lines of the power grid. We use the epsilon greedy method [40] to select a proper action, where the action with the largest Q-function value is chosen with the probability of $1 - \epsilon$, and a random action is performed with the probability of ϵ. The state, attacker, and defender's actions; the next state derived from the stochastic transition function; and the gained reward are stored for some time. These data are then sampled uniformly from this memory bank to update the network, which is called experience replay, as some random batches of transition are sampled. The difference between the target Q function and the predicted Q function is calculated as

$$\text{error} = Q_t^* - Q_t(s_{t+1}, a^1, a^2; \theta_t), \tag{12}$$

where a small error indicates a well-trained algorithm. Typically, a gradient descent algorithm can be used to optimize the online network parameter values to minimize the error. The target network's parameters are updated periodically to match the ones of the online network. Both the target network and experience replay can dramatically improve the performance of the algorithm [38]. Using the Q functions defined in Eqs. (5) and (6) for the stochastic zero-sum game, we determine the optimal attacking sequence so that the defender can choose the best defense strategy.

The main difference between Q-learning and deep Q-learning lies in the implementation of the Q table. In a problem with a large number of state-action pairs, the Q table becomes unmanageably large and impractical. This is because the greater the number of rows and columns, the more time it requires for the agents to explore the states and to update their values. In deep Q-learning, the idea is that, rather than mapping a state-action pair to a Q value using the Q table, neural networks can be exploited to map the states to the action–Q-value pairs. That is, instead of visiting different state-action pairs and filling in the Q table, a deep neural network is trained to approximate the Q function.

B. Defensive strategy algorithm using deep Q-learning

Figure 5 presents the proposed algorithm for articulating a defense strategy to protect a smart power grid from cyberattacks. The attacker and defender play a stochastic

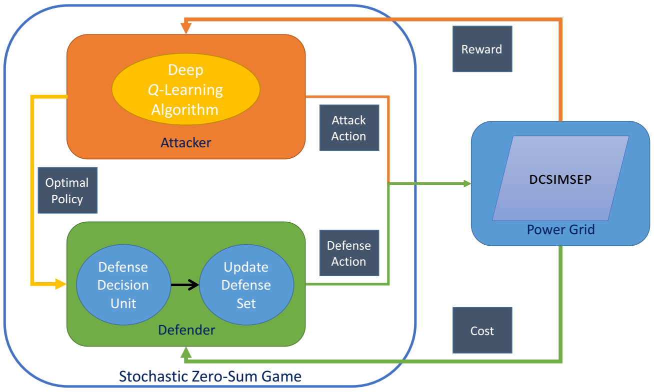

FIG. 5. Defensive strategy algorithm based on deep Q-learning in a stochastic zero-sum game. Attacker and defender are the two players of this game. Attacker uses the deep Q-learning algorithm to find an optimal attack sequence to maximize the generation loss or transmission line outage, while the defender updates its defense set based on the attacker's previous policy. Chosen actions of both players are given to the DCSIMSEP power flow simulator and the reward (cost) is then calculated and returned to the players. Process continues until the defender's protection set remains unchanged for a number of cycles.

zero-sum game with the defined objective of disabling a fixed number of transmission lines or maximizing (minimizing) the generation loss. The attacker attacks the power system while the defender protects some transmission lines. The payoff, which is either the generation loss or the number of downed transmission lines, is determined using DCSIMSEP based on the players' actions. Both players receive the reward for (cost of) their actions. The attacker uses deep Q-learning to optimize the attack sequence. Once an optimal attacking strategy is reached, it is transmitted to the defender. The defense decision management unit will decide whether or not to update the protection set. More specifically, the decision unit will simply update the protection set with the sweet targets of the previous learning process, which are the transmission lines that have the largest Q-function value for the current state. The defense decision unit will not update the protection set in the case of periodic alternation of sweet targets, which is the indicator of convergence of the algorithm. This procedure continues until a Nash equilibrium (equilibria) is reached.

IV. RESULTS

To demonstrate the workings and power of our deep Q-learning algorithm in generating optimal defense strategies against attacks, we use the benchmark W&W 6-bus and IEEE 30-bus systems. Specifically, for the relatively small W&W 6-bus system, the generation loss problem is studied in more detail with physical insights. For the larger IEEE 30-bus system, we focus on both the switching line (transmission line outage) and the maximum generation

TABLE I. Simulation parameters for W&W 6-bus system generation loss and IEEE 30-bus system generation loss and switching line problems.

Parameters	W&W6 gen	IEEE30 switch	IEEE30 gen
Trans. lines	11	41	41
Episodes	2e3	2e3	1e4
Attack length	5	4	5
Epsilon	1	1	1
Eps. decay	0.005	0.0008	0.005
Eps. min	0.01	0.001	0.01
Learn. rate	0.001	0.001	0.001
Disc. factor	0.7	0.7	0.8
Minibatch size	256	1024	256
FF. neurons	100	200	200
Attack succ. prob.	0.8	0.9	0.9

loss problems. All the simulations are carried out using the MATLAB R2021b reinforcement learning toolbox on a desktop PC with an Intel Core i7-6850K CPU and 128 GB of RAM. Table I lists the simulation parameter values for each problem. In our simulations, we assume that an attack on a specific line is successful with a preassigned probability that depends on the defender's protection set, which is updated after the attacker's learning process. For example, in the W&W 6-bus system, suppose the defender protects line 5. If the attacker attacks any line other than 5, the probability of that line's outage will be p. However, if the attacker attacks line 5, it will not go down, since the defender protects it, but failures can occur with the same probability p. The value of p may depend on the available resources allocated to the defender or the attacker at each time step. During the dynamic interplay between the attacker and defender, the value of p is treated as a constant. The reason lies in the tacit assumption that both sides have equal access to the resources, so assigning extra resources to any specific transmission line is disallowed. It is worth noting that deep Q-learning generally runs much faster than the equivalent Q-learning algorithm on a per episode basis, because the computation complexity of deep Q-learning can be significantly reduced when neural networks are used instead of a table, as in conventional Q-learning. In all cases, the core of our deep Q-learning system is a neural network consisting of two fully connected and two ReLu layers. Q8

A. Optimal defense strategy for W&W 6-bus system against generation loss

We study the maximum generation loss problem, a stochastic zero-sum game in which the attacker aims to maximize, but the defender aims to minimize, the generation loss caused by the attacks, with probabilistic state transitions. The attacker's reward at each step is equal to $\bar{G}_{\text{loss}}$ defined in Eq. (9). The zero-sum nature of the

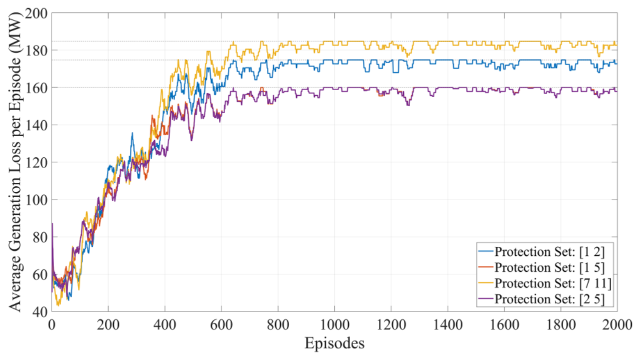

FIG. 6. Effect of choosing an effective protection set in the worst-case scenario of generation loss in the W&W 6-bus system. Attacker uses deep Q-learning to find an optimal attack sequence, while the defender updates its protection set according to the attacker's policy. Starting from a random protection set $\{7, 11\}$, the defender finds the optimal defense set to be $\{2, 5\}$, which causes the worst-case scenario of the generation loss to be reduced by %13.41.

game dynamics stipulates that the defender's reward must be $-\bar{G}_{\text{loss}}$. To be concrete, we assume that the defender is able to defend two lines at a time, while the attacker can attack up to five lines in a sequential manner. The specific numbers can be chosen arbitrarily. Figure 6 depicts $\bar{G}_{\text{loss}}$ per episode for different protection sets. First, for a random protection set $\{7, 11\}$, we apply deep Q-learning to find the attacker's sweet targets, the transmission lines that have the largest Q-function value for the initial state. From the specific random protection set, the sweet targets are determined to be lines 1 and 2, so the protection set is updated to lines $\{1, 2\}$. We apply deep Q-learning again, resulting in lines 1 and 5 becoming the updated sweet targets. For the protection set $\{1, 5\}$, the new sweet targets are lines 2 and 5. Further steps of the game plan will result in a Nash equilibrium of 159.93 MW generation loss, alternating between the protection sets $\{1, 5\}$ and $\{2, 5\}$, which represent the solution of the optimal defense sets to this problem. Intuitively, the derived sequence of the attacker's actions and the protection set constituting a Nash equilibrium can be interpreted as pairs of actions from which neither the attacker nor the defender is inclined to deviate unilaterally. As shown in Fig. 6, this optimal choice of the protection set results in a 13.41% decrease in the worst-case scenario of generation loss where the attacker plays the optimal sequence strategy.

B. Optimal defense strategy for IEEE 30-bus system against attacks on switching lines

In the switching line problem, the attacker has a fixed objective of disabling a specific set of transmission lines. Our concrete setting is that the defender is able to defend up to three lines at a time, while the attacker can attack up to four lines sequentially with the AO set to five lines. The reward function is given by Eq. (10) with $r_1 = 10$ and $r_2 = 1$. Starting with a random protection set $\{1, 2, 3\}$, we apply our deep Q-learning algorithm and identify the sweet targets as lines 15 and 16. The protection set is then updated to $\{15, 16\}$, and the worst-case scenario reward is decreased significantly, as shown in Fig. 7. Further gaming steps result in the protection set $\{15, 16\}$ as the Nash equilibrium. The intuitive reason is that, when protecting lines $\{15, 16\}$, the attacker is not able to find a sequence that will result in a large instantaneous outage. As a result, the attack receives a much smaller reward compared to the case when the defender defends a random protection set. This phenomenon is helpful for the defender in the scenario where the generation loss can be compensated for by somewhere else for the demand, making the transmission line outage a priority.

C. Optimal defense strategy for IEEE 30-bus system against attack-induced generation loss

We demonstrate the power of our deep Q-learning algorithm to solve the generation loss problem for the IEEE 30-bus system, which otherwise is not solvable using conventional tabular Q-learning. Figure 8 shows $\bar{G}_{\text{loss}}$ per episode for different protection sets, where the simulation setting is that the defender is able to defend up to three lines at a time, while the attacker can attack up to five lines sequentially. Starting from a random protection set $\{1, 2, 3\}$, with the worst-case scenario generation loss per episode of 74.87 MW, the protection set evolves from $\{16, 11, 14\}$ to $\{16, 11, 15\}$ and finally to the optimal protection set $\{16, 15, 28\}$ that results in 50.49 MW generation

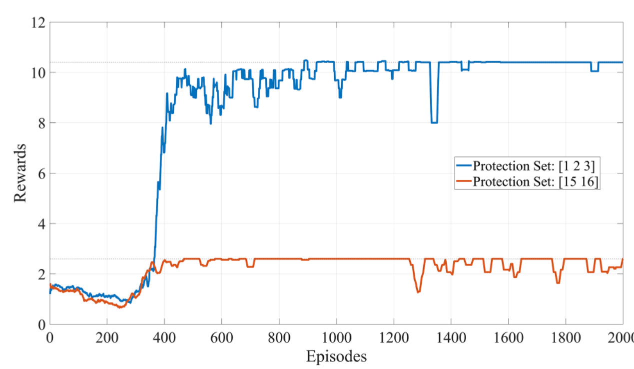

FIG. 7. Evolution of reward function values during the learning phase in the switching line problem in the IEEE 30-bus system for a random and an optimal protection set. While the defender chooses a random protection set $\{1, 2, 3\}$, the attacker finds an optimal sequence to obtain the reward of $r = 10.4$ [calculated by Eq. (10) with $r_1 = 10$ and $r_2 = 1$]. After a number of cycles, the defender chooses $\{15, 16\}$ as its protection set. As a result, the attacker fails to find a sequence with a reward of more than $r = 2.6$.

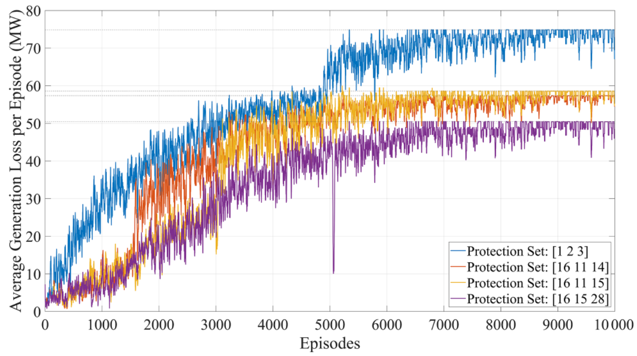

FIG. 8. Optimal protection set against the worst-case scenario of generation loss in the IEEE 30-bus system. Defender chooses a random protection set $\{1, 2, 3\}$, whereas the attacker finds an optimal policy to maximize the generation loss. After a number of cycles, the defender chooses $\{16, 15, 28\}$ as its protection set and, as a result, the worst-case scenario generation loss caused by the optimal attack sequence is reduced by 48.28%.

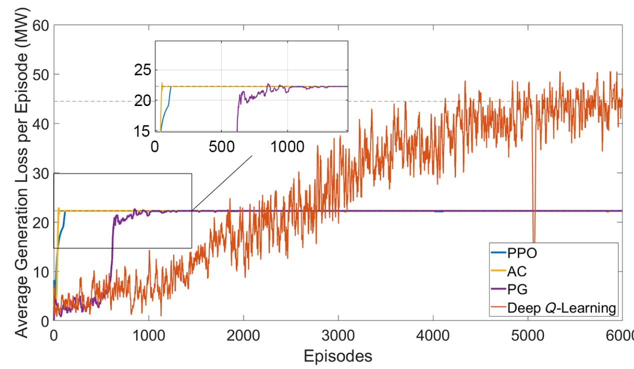

FIG. 9. Comparison with representative existing RL algorithms. Shown is the performance comparison of the deep Q-learning with PG, AC, and PPO algorithms for the generation loss problem in the IEEE 30-bus system. Maximum generation loss caused by the optimal attack sequences derived by the PPO, AC, and PG agents is 22.24 MW, while our deep Q-learning agent is able to obtain 50.49 MW. While the deep Q-learning algorithm takes a longer time to converge, reliability and efficiency are guaranteed.

loss. Using the optimal protection set can result in 48.28% mitigation of the worst-case generation loss, even if the attacker chooses the optimal attacking sequence.

It is worth noting that the IEEE 30-bus system simulation is used to demonstrate that conventional Q-learning is unable to deal with this system, while our deep Q-learning can. The system is only regarded as "large" in a relative sense: it is much larger than the W&W 6-bus benchmark system. Much larger systems are available, e.g., the IEEE 300-bus or IEEE 3000-bus systems, which can be simulated using specific power-grid software, such as Gridlab-D. Deep RL methods are applicable to these larger systems, but the required computations are beyond our current capability.

D. Comparison with alternative RL algorithms

We compare the performance of our deep Q-learning algorithm with three widely used RL algorithms for discrete state-action space systems: PG, AC, and PPO. The PG algorithm [41] is a rudimentary policy-based model-free online on-policy method, while the AC algorithm aims to optimize the policy (actor) directly and train a critic to estimate the return or future rewards [42]. PPO [43] is an actor-critic model-free online on-policy algorithm that alternates between data sampling by interacting with the environment and optimization of a clipped objective function, which leads to improved training stability by limiting the size of the policy change at each step. We set the learning rate, discount factor, and other applicable key simulation parameters to the same values as in deep Q-learning. The actor and critic networks in both the PPO and AC algorithms have the same structure as the critic network in our deep Q-learning algorithm and the actor network in the PG algorithm for fair comparison. The protection set for all algorithms is set to $\{16, 15, 28\}$, which is the Nash equilibrium in Sec. IV C. Figure 9 shows that the maximum generation loss caused by the attacker in the PPO, AC, and PG algorithms converges to 22.24 MW, while that in our deep Q-learning algorithm converges to 50.49 MW. Generally, the deep Q-learning algorithm takes a long time to converge, but the reliability and efficiency compensate for the slow convergence since real-time computation is not needed in strategy planning. Moreover, due to the large size of action and state spaces, asymmetric and stochastic state transitions, and insufficient exploration of the state space intrinsic to the other algorithms, our deep Q-learning algorithm significantly outperforms the PPO, AC, and PG algorithms.

V. DISCUSSION

The problem of devising optimal defense strategies to protect smart power grids from cyberattacks is of significant current interest. Given a grid system, a general principle is to simulate attacks to identify the scenario(s) that can result in the most severe damage to define the best possible defense strategies. This attacker-defender interaction problem can be modeled as a stochastic zero-sum game, for which machine learning can provide effective solutions. In recent years, conventional RL, in particular, Q-learning, has been applied to the attacker-defender game problem, but a fundamental shortcoming is the exponentially growing state space as the size of the system increases linearly. We articulate a general deep Q-learning framework to solve the game problem in arbitrarily large power-grid systems. We demonstrate that our deep Q-learning algorithm typically leads to a Nash equilibrium, and the corresponding strategy represents the optimal solution. We test the proposed framework under different attack-defense scenarios for the W&W 6-bus system used in the current Q-learning literature and the relatively large IEEE 30-bus system that cannot be solved with the conventional Q-learning algorithm. We also compare the results of our deep Q-learning algorithm to those from three alternative but state-of-the-art RL algorithms and demonstrate the superiority of our method.

Immediate future work is expanding the deployment of the deep RL algorithms to a general sum problem, in which both the attacker and defender have limited resources that they can use for their actions. The reward function would also be different from the one used in this paper, where the defender attempts to mitigate the consequences, whereas the attacker has a set objective. The results in this paper suggest that deep Q-learning can be effective at addressing the general sum game to devise the optimal resource allocation policy.

ACKNOWLEDGMENTS

This work is mainly supported by the US-Israel Energy Center managed by the Israel-US Binational Industrial Research and Development (BIRD) Foundation. This work is also supported by AFOSR under Grant No. FA9550-21-1-0438.

APPENDIX: A DETAILED DESCRIPTION OF THE DEEP Q-LEARNING METHOD

Deep Q-learning is a model-free framework in which the agent uses a neural network architecture to train a critic to estimate the future cumulative rewards characterizing how valuable one action is at each state. While there are reinforcement learning methods for continuous action spaces (e.g., deep deterministic policy gradient [44] and twin-delayed deep deterministic policy gradient [45]), deep Q-learning is only applicable to discrete action spaces.

The structure of the deep Q-learning method in our work is shown Fig. 10, which illustrates what happens inside the attacker block in Fig. 5. Modeling the attacker-defender interaction as a zero-sum game has the advantage of learning a single Q function (in a general sum game, learning multiple Q functions would be necessary). For each state input, the deep Q-learning structure returns an approximation of the Q function for that state and all possible actions. In our problem, by "state" we mean the state of the transmission lines in the power grid, which is denoted as a binary-valued vector. The attacker's action is chosen from the set $A = \{1, 2, 3, \ldots\}$, where action i means attacking transmission line i. The defender's action is a set consisting of n transmission lines denoted as the protection set. The environment block in Fig. 10 represents the power grids studied in this paper. As described in the main text, we employ DCSIMSEP, a dc load flow simulator of cascading (separation) in power systems, to simulate the dynamics of the power grid. Using our modified DCSIMSEP code, we generate the observation and rewards for each attack (and defense) actions and feed them to the algorithm in the next step.

A deep Q-learning agent is represented by a critic neural network. During the training phase, this critic is trained to approximate the expectation of the cumulative future rewards. The critic neural network is parameterized. During training, the agent tunes the parameter values to improve the accuracy of the estimation. The neural network structure consists of two fully connected and two ReLu layers (as detailed in Table I). In particular, a fully connected layer multiplies the input by a weight vector and adds a bias into it, which is similar to a nonlinear principal component analysis for improving the estimation accuracy. The ReLu layers set the negative values of the input to zero and perform a threshold operation on the input; these are nonlinear transformations to expedite the training process.

Here, we model the attacker and defender interaction as a zero-sum game, with the goal of disabling a fixed number of transmission lines or maximizing (minimizing) the generation loss. Both players receive the reward for (or cost of) their actions. The attacker uses deep Q-learning to optimize the attack sequence. During the training process, the agent explores the state space, i.e., the attacker attacks different transmission lines to observe the results. This exploration follows a standard greedy algorithm method, where sometimes the attacker launches random attacks and at other times the attack is based on what the attacker has learned so far. The past experiences are stored using an experience buffer. The critic neural network is updated based on a pool of experiences randomly sampled from this buffer. Once an optimal attacking strategy is reached, it is transmitted to the defender, and the defender will update its protection set to be better prepared against future attacks. This process continues until the Nash equilibrium of the game is reached.

We perform the simulation using MATLAB's reinforcement learning toolbox. For the deep Q-learning algorithm, we use the rlDQNAgent object. The options set for rlDQNAgentOptions are listed in Table I. The state space is defined using rlNumericSpec, and the action space type is selected as rlFiniteSetSpec. No external lower or upper limits are applied to these spaces. The environment (*env* object) is customized using the modified DCSIMSEP. Eventually, the critic is a rlQValueRepresentation object with the neural network layer depicted in Fig. 10. The codes and simulation results are available at Github [46].

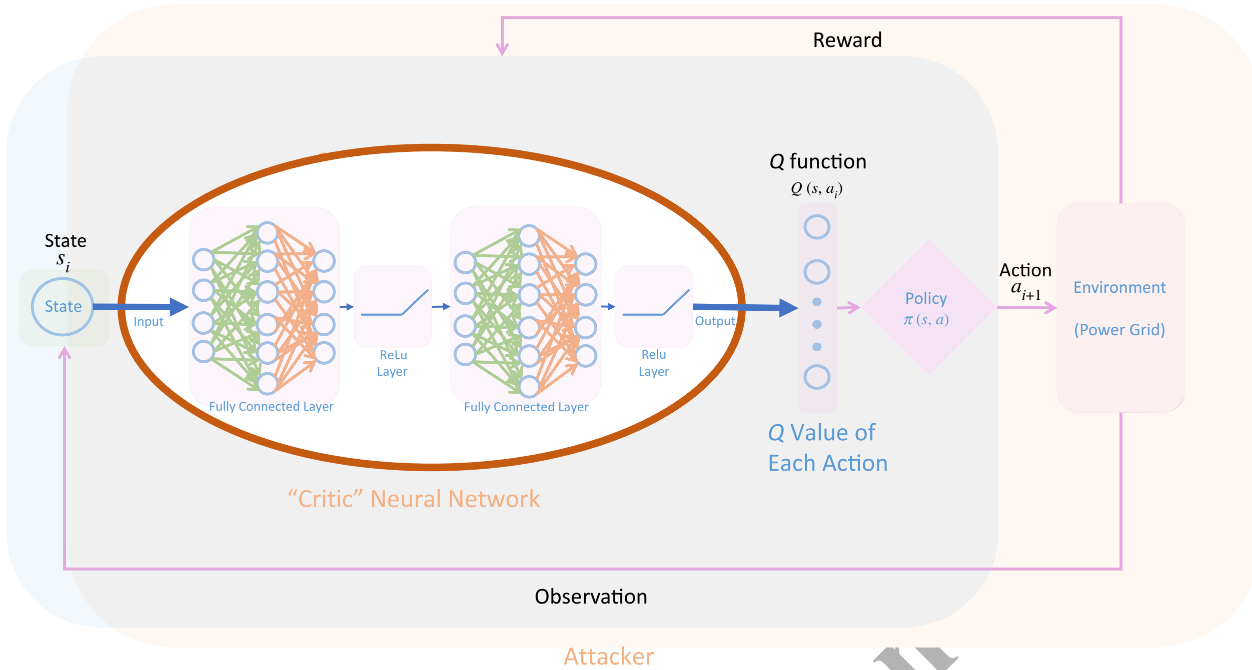

FIG. 10. Structure of deep Q-learning algorithm used in this paper. Structure describes the processes inside the attacker block in Fig. 5. Environment block contains the power grids simulated using our modified DCSIMSEP algorithm. DCSIMSEP generates the observation and rewards for each attack (and defense), which are fed to the algorithm in the next step. Through interacting with the environment, the critic returns an approximation of the Q function for the input state (the state of transmission lines) and all possible actions (attack actions or protection sets). This critic neural network is parameterized. During training, the agent tunes the parameter values to make the estimation more accurate. Critic consists of two fully connected and two ReLu layers, the specifications of which are listed in Table I. Attacker uses this algorithm to optimize the attack sequence. Once an optimal attacking strategy is reached, the defender will update its protection set (Fig. 5) to be better prepared against future attacks. This repeats until the optimal protection set has been found.

[1] P. Pourbeik, P. S. Kundur, and C. W. Taylor, The anatomy of a power grid blackout—root causes and dynamics of recent major blackouts, IEEE Power Energy Mag. **4**, 22 (2006).

[2] J. Xie, A. Stefanov, and C.-C. Liu, Physical and cyber security in a smart grid environment, Wiley Interdis. Rev. Energy Envir. **5**, 519 (2016).

[3] R. Langner, Stuxnet: Dissecting a cyberwarfare weapon, IEEE Secu. Priv. **9**, 49 (2011).

[4] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, The 2015 Ukraine blackout: Implications for false data injection attacks, IEEE Trans. Power Sys. **32**, 3317 (2017).

[5] Y. Liu, P. Ning, and M. K. Reiter, False data injection attacks against state estimation in electric power grids, ACM Trans. Info. Sys. Secu. **14**, 1 (2011).

[6] L. Xie, Y. Mo, and B. Sinopoli, Integrity data attacks in power market operations, IEEE Trans. Smart Grid **2**, 659 (2011).

[7] M. Mohammadpourfard, Y. Weng, and M. Tajdinian, Benchmark of machine learning algorithms on capturing future distribution network anomalies, IET Gene. Transmi. Distri. **13**, 1441 (2019).

[8] A. Shefaei, M. Mohammadpourfard, B. Mohammadi-ivatloo, and Y. Weng, Revealing a new vulnerability of distributed state estimation: A data integrity attack and an unsupervised detection algorithm, IEEE Trans. Cont. Net. Sys. (2021),.

[9] N. Enriquez and Y. Weng, in *Asian Conference on Machine Learning (ACML), PMLR 157* (2021) p. 1333.

[10] J. F. Nash, Equilibrium points in *n*-person games, Proc. Nat. Aca. Sci. (USA) **36**, 48 (1950).

[11] T. Başar and G. J. Olsder, *Dynamic Non-Cooperative Game Theory* (SIAM, 1998), 2nd ed. Q9

[12] W. Saad, Z. Han, H. V. Poor, and T. Basar, Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications, IEEE Sig. Proc. Mag. **29**, 86 (2012).

[13] S. Poudel, Z. Ni, X. Zhong, and H. He, in *2016 International Joint Conference on Neural Networks (IJCNN)*, p. 2730.

[14] N. I. Haque, M. H. Shahriar, M. G. Dastgir, A. Debnath, I. Parvez, A. Sarwat, and M. A. Rahman, Machine learning in generation, detection, and mitigation of cyberattacks in smart grid: A survey, arXiv preprint (2020), ArXiv:2010.00661.

[15] Y. Chen, S. Huang, F. Liu, Z. Wang, and X. Sun, Evaluation of reinforcement learning-based false data injection attack to automatic voltage control, IEEE Trans. Smart Grid **10**, 2158 (2019).

[16] B. Ning and L. Xiao, in *2021 40th Chinese Control Conference (CCC)* (IEEE), p. 8598.

[17] C. J. C. H. Watkins and P. Dayan, *Q*-learning, Mach. Learn. **8**, 279 (1992).

[18] J. Yan, H. He, X. Zhong, and Y. Tang, *Q*-Learning-based vulnerability analysis of smart grid against sequential topology attacks, IEEE Trans. Info. Foren. Secu. **12**, 200 (2017).

[19] Z. Wang, H. He, Z. Wan, and Y. Sun, Coordinated topology attacks in smart grid using deep reinforcement learning, IEEE Trans. Indust. Info. **17**, 1407 (2020).

[20] C. Roberts, S.-T. Ngo, A. Milesi, S. Peisert, D. Arnold, S. Saha, A. Scaglione, N. Johnson, A. Kocheturov, and D. Fradkin, in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)* (IEEE), p. 1.

[21] Y. Li and J. Wu, Low latency cyberattack detection in smart grids with deep reinforcement learning, (2022),. Q10

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, Playing Atari with deep reinforcement learning, (2013), ArXiv:1312.5602.

[23] J. Hu and M. P. Wellman, in *ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning* (1998), p. 242.

[24] S. R. Etesami and T. Basar, Dynamic games in cyber-physical security: An overview, Dyn. Games Appl. **9**, 884 (2019).

[25] D. Vrabie and F. Lewis, in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, p. 1.

[26] Q. Zhu, H. Tembine, and T. Bbar, Heterogeneous learning in zero-sum stochastic games with incomplete information, 49th IEEE Conference on Decision and Control (CDC), 219 (2010).

[27] P. Bommannavar, T. Alpcan, and N. Bambos, Security risk management via dynamic games with learning, 2011 IEEE International Conference on Communications (ICC), 1 (2011).

[28] A. Truong, S. R. Etesami, J. Etesami, and N. Kiyavash, Optimal attack strategies against predictors - learning from expert advice, IEEE Trans. Info. Foren. Secu. **13**, 6 (2018).

[29] Q. Zhu, H. Tembine, and T. Başar, in *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control* (2013), p. 303.

[30] K.-W. Chung, C. A. Kamhoua, K. A. Kwiat, Z. T. Kalbarczyk, and R. K. Iyer, in *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)* (2016), p. 1.

[31] X. He, H. Dai, and P. Ning, Improving learning and adaptation in security games by exploiting information asymmetry, 2015 IEEE Conference on Computer Communications (INFOCOM), 17872015.

[32] K. K. Trejo, J. B. Clempner, and A. S. Poznyak, in *2016 IEEE 55th Conference on Decision and Control (CDC)*, p. 5484.

[33] M. J. Eppstein and P. D. H. Hines, A "random chemistry" algorithm for identifying collections of multiple contingencies that initiate cascading failure, IEEE Trans. Power Sys. **27**, 1698 (2012).

[34] P. Rezaei, P. D. H. Hines, and M. J. Eppstein, Estimating cascading failure risk with random chemistry, IEEE Trans. Power Sys. **30**, 2726 (2015).

[35] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, A framework for cyber-topology attacks: Line-switching and new attack scenarios, IEEE Trans. Smart Grid **10**, 1704 (2019).

[36] S. Paul and Z. Ni, in *2018 International Joint Conference on Neural Networks (IJCNN)*, p. 1.

[37] Z. Ni and S. Paul, A multistage game in smart grid security: A reinforcement learning solution, IEEE Trans. Neural Netw. Learn. Syst. **30**, 2684 (2019).

[38] D. S. H. van Hasselt and A. Guez, in *Proceedings of the thirtieth AAAI Conference on Artificial Intelligence* (2016), p. 1928.

[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. 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 **518**, 529 (2015).

[40] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (The MIT Press, 2018), 2nd ed.

[41] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. **8**, 229 (1992).

[42] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, in *Proceedings of the 33rd International Conference on Machine Learning Research 2016*, Vol. 48 (2016), p. 1928.

[43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal policy optimization algorithms, (2017), ArXiv:1707.06347.

[44] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, Continuous control with deep reinforcement learning, (2015), arXiv preprint ArXiv:1509.02971.

[45] S. Fujimoto, H. van Hoof, and D. Meger, Addressing function approximation error in actor-critic methods, (2018), arXiv preprint ArXiv:1802.09477.

[46] https://github.com/AminMoradiXL/DQN_grid.