IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 12, 2019, accepted July 22, 2019, date of publication August 2, 2019, date of current version August 22, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2933020*

# Defending Against Data Integrity Attacks in Smart Grid: A Deep Reinforcement Learning-Based Approach

**DOU AN[1], QINGYU YANG[1,2], (Member, IEEE), WENMAO LIU[3], AND YANG ZHANG[1]**

[1]MOE Key Laboratory for Intelligent Networks and Network Security, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[2]SKLMSE Lab, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[3]NSFOCUS Inc., Beijing 100089, China

Corresponding author: Qingyu Yang (yangqingyu@mail.xjtu.edu.cn)

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB0803501, in part by the National Science Foundation of China under Grant 61803295, Grant 61673315, and Grant 61833015, in part by the Shaanxi International Cooperation and Exchange Program under Grant 2017KW-039, and in part by the CCF-NSFOCUS Research Foundation.

**ABSTRACT** State estimation plays a critical role in monitoring and managing operation of smart grid. Nonetheless, recent research efforts demonstrate that data integrity attacks are able to bypass the bad data detection mechanism and make the system operator obtain the misleading states of system, leading to massive economic losses. Particularly, data integrity attacks have become critical threats to the power grid. In this paper, we propose a deep-Q-network detection (DQND) scheme to defend against data integrity attacks in alternating current (AC) power systems. DQND is a deep reinforcement learning scheme, which avoids the problem of curse of dimension that conventional reinforcement learning schemes have. Our strategy in DQND applies a main network and a target network to learn the optimal defending strategy. To improve the learning efficiency, we propose the quantification of observation space and utilize the concept of slide window as well. The experimental evaluation results show that the DQND outperforms the existing deep reinforcement learning-based detection scheme in terms of detection accuracy and rapidity in the IEEE 9, 14, and 30 bus systems.

**INDEX TERMS** Cyber-physical systems, smart grid, data integrity attacks, deep reinforcement learning, Q-learning.

## I. INTRODUCTION

As a typical energy Cyber-physical Systems (CPS), the smart grid is designed to effectively monitor and control the two-way power and information flow between consumers and the grid by integrating advanced sensing, control and measurement technologies [1]–[5], [48], [54]. Nonetheless, the smart grid is more vulnerable to threats from the cyber space than traditional power systems because of the diversified and open network environment [6]–[9]. For example, Ukrainian power grid was attacked by Blackenergy and Kill Disk in 2015, causing several substations to be powered down for up to 3 hours [48]. Moreover, the Stuxnet attacked SIMATIC system in 2010, invaded the industrial control system and forged the centrifuge operating data, causing damage to nearly 1,000 centrifuges in the Iran's nuclear power system [9].

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

During the recent past, cyber attacks in the smart grid have drawn great attentions [12], [14]. Among them, data integrity attacks [15]–[19] are extremely dangerous to the power system due to the difficulty in analyzing the attack behavior, the difficulty in detecting the abnormal information, and the difficulty in recovering the system state. Depending on the target of the adversary, data integrity attacks fall into the following categories: attacks against state information (i.e., measurements) [16], [20], [32], [33], attacks against interactive electricity information (i.e., load demands, electricity price) [22], [42]. For instance, regarding to the attack against state information, Sandberg *et al.* proposed security parameters to quantify the minimum number of measurements to compromise so that an attack could be successfully launched, and utilized graph theory to derive data integrity attack vector [33]. Regarding to attack against interactive electricity information, Bi and Zhang *et al.* investigated the problem that the adversary could manipulate

  

the real-time electricity price and designed corresponding countermeasures [22]. Moreover, our prior works [8], [16], [49], [50], [52] presented the investigation of data integrity attacks with least efforts against static state estimation, dynamic state estimation, optimal power flow, multiple-step electricity price in power systems.

To defend against data integrity attacks, detection-based and protection-based schemes have been proposed. In terms of detection-based schemes, most of the existing research efforts are targeted at Energy Management System (EMS) modules in traditional power system such as static state estimation, optimal power flow [14], [21], [24], [36], [39], [45]. For example, Lee *et al.* proposed an adaptive denial-of-service attack mitigation scheme by applying historical data statistics information of the power system to filter data integrity attacks [30]. To deal with data integrity attacks, Yang *et al.* proposed a detection scheme using a Gaussian-Mixture Model-based mechanism to improve the detection accuracy [39]. Likewise, Ashok *et al.* [21] investigated a bad data detection algorithm for smart grid real-time monitoring based on load forecasting, power generation planning and phasor measurement data to defend against data integrity attacks. Likewise, Esmalifalak *et al.* analyzed the data distribution of the historical state information and proposed a detection scheme based on support vector machine [24]. Related to this efforts, an optimal PMU placement-based protection scheme and an integrated detection schemes were proposed as well [9], [51].

In recent years, due to the benefits of obtaining optimal action policies, reinforcement learning techniques have received great attention in addressing the sequential decision problems, i.e., game, control, as well as attack detection [10], [11], [53]. For example, Chen *et al.* adopted a novel Q-learning schemes that utilized the concept of the nearest sequence memory to learn the optimal attack strategy from the adversary's perspective [23]. Kurt *et al.* utilized SARSA reinforcement learning scheme to detect cyber-attacks in the state estimation module of smart grid [26]. Nonetheless, the features of power system state estimation model are not fully formalized and characterized in the reinforcement learning model of existing research efforts.

To this end, we propose a deep reinforcement learning-based scheme to detect data integrity attacks in AC power grid. The key contributions of our paper are as follows:

- First, we review the model of nonlinear AC power system and the data integrity attack model. Furthermore, we formulate the defensive process against data integrity attacks as a Markov Decision Process (MDP). In such a process, we present the formulation of state space, action space, reward function, and observation space, which eliminates the effects of noise and improve the accuracy and rapidity of detection strategy.
- Second, we propose a Deep-Q-Network Detection (DQND) scheme to defend against data integrity attacks. In our scheme, a main network and a target network are set up to learn the defensive strategy. We also apply the concept of slide window and quantify the observation space to avoid the curse of dimension so that the learning efficiency can be improved.
- Finally, we conduct an extensive performance evaluation of our detection scheme, we define three evaluation metrics: delay-alarm error rates, false-alarm error rates, and detect-failure rates. We design two attack models in our evaluation: continuous attack model and discontinuous attack model. We carry out extensive performance evaluation on IEEE 9, 14 and 30 bus systems, respectively. In comparison with the SARSA detection strategy and $SARSA_{imp}$ detection strategy, our DQND achieves the best performance in terms of delay-alarm error, false-alarm error, and detect-failure rates.

The remainder of this paper is organized as follows. We first introduce the system model of nonlinear AC power flow, the attack model, and then briefly review the concept of deep reinforcement learning in Section II. In Section III, we first introduce the Markov Decision model of our strategy, and then introduce the quantification of observation space and slide window. After that, we present our DQND scheme in detail. In Section IV, we show evaluation results to demonstrate the effectiveness of DQND scheme in detecting data integrity attacks. We discuss the future research directions and ongoing works of this paper in Section V. We conclude the paper in Section VI.

## II. BACKGROUND

In this section, we first present the nonlinear AC power system state estimation model. We then introduce the data integrity attack model from the adversary's perspective. Finally, we briefly review the concept of deep reinforcement learning.

### A. SYSTEM MODEL

Denote that there are $M$ smart meters under an $N$-bus AC power system model. We denote the state of system at time $t$ as $x_t = [x_{1,t}, \ldots, x_{N,t}]$, the state $x_{i,t}$ contains phase angles and voltage magnitudes of bus $i$ at time $t$. The measurement vector is expressed as $y_t = [y_{1,t}, y_{2,t}, \ldots, y_{M,t}]$, where $y_{j,t}$ is denoted as the measurement of smart meter $j$ at time $t$. It is worth mentioning that we have $M > N$ that assures the robustness of measurement system.

In AC power system, the measurement vector and the state are related as:

$$y = h(x) + e, \tag{1}$$

where $h : R^N \rightarrow R^M$ is denoted as a nonlinear function and is determined by the power system model and $e$ is a measurement error vector. The error vector $e$ follows a Gaussian distribution, in which $S$ denotes the covariance matrix. The system model of IEEE 30 bus system is illustrated in Figure 1.

In this paper, the weighted least squares mechanism is used to estimate the system state. The basic principle of the

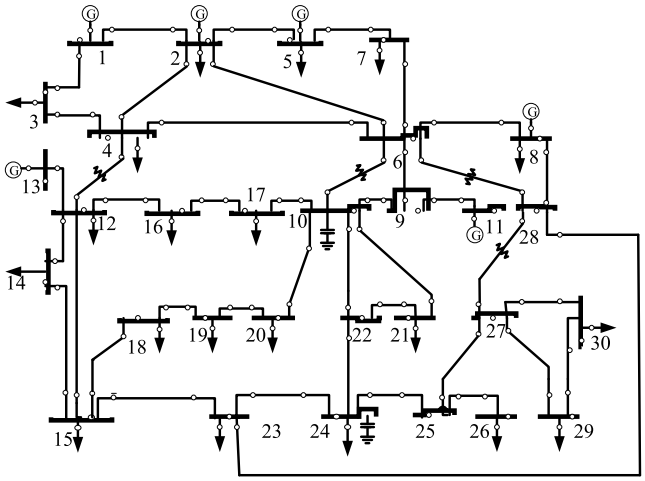

**FIGURE 1.** IEEE-30 bus system.

weighted least squares mechanism is to learn the optimal estimate value of the real state vector $x$ by minimizing the sum of error's squares. Therefore, the optimal state estimation is expressed as:

$$\hat{x} = \arg\min_x J(x) = \arg\min_x (y - h(x))^T S^{-1} (y - h(x)), \quad (2)$$

where $J(x)$ denotes the objective function. The Gauss-Newton method is then applied to obtain the optimal state estimation. In each iteration, the state is updated as:

$$x_{i+1} = x_i + \Delta x, \quad (3)$$

$$\Delta x = \left(H^T S^{-1} H\right)^{-1} S^{-1} H^T S^{-1} (y - h(x)), \quad (4)$$

where $H = \frac{\partial h(x)}{\partial x}$ is the Jacobian matrix of the measurements, which is updated during state estimation iterations.

To detect the bad data in the above process, the detection threshold is defined as $\tau$, if the estimation error is not greater than $\tau$, we believe that there is no bad data. Then, the detector is defined as:

$$\|y - h(x)\| \leq \tau. \quad (5)$$

Note that when the above inequality is satisfied, we know that the estimated state information is trustworthy and there is no bad data.

### B. ATTACK MODEL

Note that the objective of the attacker is to perform data integrity attacks to disturb the normal operation of the power grid and bypass the bad data detection mechanism. Most of the existed research efforts assume that the attacker has learned the full information of power system state, which may not be practical in real world [26], [31], [37]. Therefore, $h_f$ is defined to indicate the adversary's information of the power grid system and $\hat{x}_f$ denotes the error between the state estimation result obtained by the adversary and the true value according to [55].

The attack vector injected by the adversary is denoted as $a$, which is tempered into the measurement vector $y$, the measurement vector after attack can be expressed as:

$$y_a = y + a. \quad (6)$$

Moreover, the state estimator obtains the erroneous estimation result by weighted least squares mechanism after receiving the tampered measurement vector. Then, the erroneous estimation result is denoted as

$$\hat{x}_{bad} = \hat{x} + c. \quad (7)$$

To bypass the bad data detection mechanism in Equation 5, the attack vector should satisfy the following equations:

$$\begin{aligned} \|y_a - h(\hat{x}_{bad})\| &= \|y_a - h(\hat{x} + c)\| \\ &= \|y + a - h(\hat{x} + c) + h(\hat{x}) - h(\hat{x})\|, \\ &\leq \|y - h(\hat{x})\| + \|a - h(\hat{x} + c) + h(\hat{x})\|, \\ &\leq \|y - h(\hat{x})\| + \|a - h_f(\hat{x}_f + c) + h_f(\hat{x}_f)\| \\ &\quad + \|h_f(\hat{x}_f + c) - h(\hat{x} + c)\| \\ &\quad + \|h(\hat{x}) - h_f(\hat{x}_f)\|, \\ &\leq \tau. \end{aligned}$$

We can find out that the attack is able to bypass the bad data detection mechanisms when the inequality in the last row is satisfied. In order to meet this requirement, we need to minimize the following equation:

$$\begin{aligned} g(x) = &\|a - h_f(\hat{x}_f + c) + h_f(\hat{x}_f)\| \\ &+ \|h_f(\hat{x}_f + c) - h(\hat{x} + c)\| \\ &+ \|h(\hat{x}) - h_f(\hat{x}_f)\| \end{aligned} \quad (8)$$

From the perspective of the attacker, to minimize $g(x)$, the following equation should be satisfied:

$$a = h_f(\hat{x}_f + c) + h_f(\hat{x}_f). \quad (9)$$

By satisfying the above equations (8)-(9), the attacker can determine the measurements to be falsified and the optimal attack vector $a$. Moreover, if the attacker has already learned the full information of the accurate network topology, the attacking vector can be established according to Equation (9).

### C. DEEP REINFORCEMENT LEARNING

As a typical machine learning approach, reinforcement learning is used to learn an optimal action strategy to achieve the maximization of total rewards [29]. During the learning process, the agent interacts with environment to obtain the environment knowledge and updates the action policy over episodes. Nonetheless, the learning efficiency of conventional reinforcement learning method is relatively low so that it is not suitable for the problem with large state space. To address this issue, deep learning [53] technology that is also called deep neural network can be integrated to reinforcement learning well to improve the efficiency, the integrated field is called deep reinforcement learning [34].

Combined with reinforcement learning and deep neural networks, deep reinforcement learning scheme has mainly the following three advantages: First, deep neural networks is capable of approximating the value of state and action, which are critical in the process of learning optimal policy. Second, deep neural networks perform feature engineering and reduce the dependence of the reinforcement learning process on domain knowledge. In addition, the curse of dimensionality is a difficult problem that reinforcement learning based approach faces, especially when the state space and action space are huge. In this regard, deep reinforcement learning addresses the curse of dimensionality issue well, since that the output of neural networks can estimate the value of state and action, meaning that there is no need for storing the values of state and action and avoids the curse of dimension as a result.

Note that deep reinforcement learning approaches have been widely investigated in several areas. For instance, Deepmind has proposed AlphaGo which can beat the best human player in chess in 2016 [43], and then, Deepmind also proposed Alpha zero to strengthen the ability of AlphaGo. In addition, Su *et al.* proposed an online Active Reward Learning in Spoken Dialogue Systems in [44]. Ho *et al.* proposed a model-free imitation learning algorithm in directly extracting a policy from data [25]. Likewise, Narasimhan *et al.* employ deep-Q-network with a novel reward function to improve information extraction [35].

**TABLE 1. Notations.**

| Symbols | Descriptions |
|---|---|
| $T$ | Total time steps in one episode |
| $s_t$ | State of agent at time $t$ |
| $a_t$ | Action of agent takes at time $t$ |
| $r_t$ | The reward of agent at time $t$ |
| $o_t$ | Observation of agent at time $t$ |
| $l_t$ | The discrete value of observation value at time $t$ |
| $w_t$ | Slide window at time $t$ |
| $y_t$ | The measure vector at time $t$ |
| $\hat{x}_t$ | The estimated state at time $t$ |
| $a$ | The attack vector |
| $c_1$ | The coefficient of delay-alarm error |
| $c_2$ | The coefficient of false-alarm error |
| $\theta$ | The parameters of main network |
| $\theta'$ | The parameters of target network |
| $L$ | Loss function |
| $E$ | The number of episodes in training stage |
| $E_2$ | The number of episodes in testing stage |

## III. DQND SCHEME

In this section, we first introduce two attack scenarios, and then present the Markov Decision Process of our DQND scheme, including the state space, action space, observation space and reward function. After that, we present the DQND scheme in detail. The critical notations are listed in Table 1.

### A. ATTACK SCENARIOS

Before introducing the detail procedure of DQND scheme, we first define two attacks that the adversary might launch: continuous attack and discontinuous attack as follows:

- **Continuous attack:** The adversary launches a sequential attack over time until the attack is detected. Generally speaking, the damage caused by continuous attack to state estimation is greater than the discontinuous attack. However, the continuous attack is easy to be detected due to the continuity over time.
- **Discontinuous attack:** The adversary launches intermittent attack over time until the attack is detected. Due to the fact that the adversary can choose different intermittent time to launch attack, discontinuous attack is difficult to be detected, but the damage that caused is smaller than the continuous attack.

The continuous attack model and discontinuous attack model is shown in Figure 2.

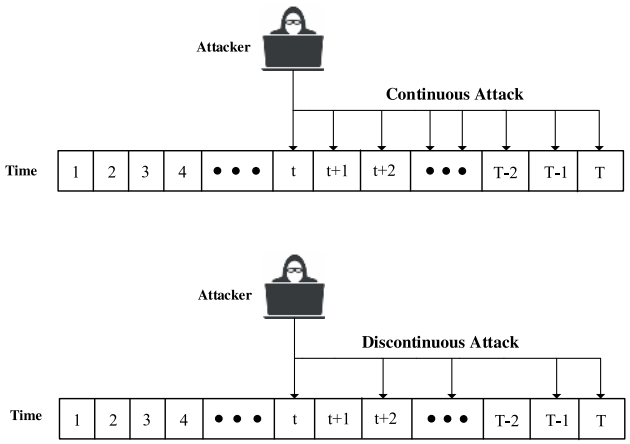


**FIGURE 2. Attack model.**

### B. MARKOV DECISION PROCESS

In this paper, the power system operator is also called as the agent, whose responsibility is to ensure the normal operation of the power grid. The Markov decision process (MDP) [38] of our scheme is defend as $(S, A, P, R, O)$, in which $S$ denotes the state space, $A$ denotes the action space, $P$ is the transition probability, $R$ is the reward function, and $O$ is the observation space. We introduce these five elements of MDP in detail as follows:

$s_t \in S$ is the state of system, $S$ is the set of possible states the agent lies in. We define the state space in this paper as $[S_n, S_a]$, where $S_n$ indicates the system operates well and $S_a$ denotes that the system is under attack. Obviously, the state is unknown for the agent, meaning that the agent cannot make sure that system is under normal operation state or under attack, the agent can only estimate the state of system according to the observation of current state.

$a_t \in A$ is denoted as the agent's action at time $t$, $A$ indicates the set of actions the agent might take. We define the action

space as $[A_c, A_s]$, where $A_c$ denotes that the agent infers that the system is in normal operation based on the observation and the system should be kept running, and $A_s$ denotes that the agent infers that the system is under attack and should be stopped.

$p_t : s_t \times a_t \rightarrow P(s_{t+1})$ indicates the probability the agent steps into $s_{t+1}$ when it takes the action $a_t$ at state $s_t$. This probability is unknown to the agent since we propose a model-free deep reinforcement learning method.

$r_t$ means the reward that the agent obtains when taking action $a_t$ at state $s_t$. We define the reward function as follows:

$$r_t = \begin{cases} 0, & s_t = s_a, \quad if\, a_s = a_s, \\ c_1 * |t - \lambda|\,; & if\, s_t = s_a, \quad a_s = a_c. \end{cases} \tag{10}$$

$$r_t = \begin{cases} 0, & s_t = s_n, \quad if\, a_s = a_c, \\ c_2 * 1000 * \dfrac{\left\| y - h\left(\hat{x}\right)\right\|}{\|w\|}, & if\, s_t = s_n, \quad a_s = a_s. \end{cases} \tag{11}$$

where $\lambda$ is the time when the attack is launched, $w$ is the size of noise that is added into the system model. Also, $c_1$ and $c_2$ are the corresponding coefficients, which balance the delay-alarm and false-alarm error. To make $c_1$ and $c_2$ satisfy the same order of magnitude, we multiply the coefficient by 1000 in Equation (11). Delay-alarm denotes the time when the agent detects the attack, which is later than the time when the attack is launched. False-alarm denotes that the agent considers that the system is under attack and stops the operation of system but there exists no attack in fact. In this paper, we aim to minimize the probability of these two types of errors.

Moreover, $o_t$ is the observation that the agent learns from the system, which is different from the system state, because the system state is not observable to the agent. We define the $o_t$ as follows:

$$o_t = \frac{\left\| y - h\left(\hat{x}\right)\right\|}{\|w\|}. \tag{12}$$

From the equation, we can see that when the system is in normal operation, $y$ is close to $h(\hat{x})$ and $o_t$ is a small value. On the other hand, when the system is under attack, the value of $h(\hat{x})$ differs greatly from the $y$ and $o_t$ will be a large value. Therefore, the size of $o_t$ accurately reflects whether the system has been attacked. In addition, the definition of observation eliminates the impact of noise. Specifically, when the system is under normal operation and the size of noise is large, the observation $o_t$ is small and the effect of noise can be ignored. As a consequence, the situation that the agent mistakenly detects that the system is being attacked can be avoided.

### C. QUANTIFICATION OF OBSERVATION SPACE AND SLIDE WINDOWS

Recall that the curse of dimension is a common issue in reinforcement learning. The existence of high dimensional feature space makes that the reinforcement learning problem cannot be solved in limited time with limited resources. To address this issue, in this paper we quantify the observation space into limited discrete value. We set up multiple non-overlapping intervals as: $[a_1, a_2]$, $[a_2, a_3]$, $[a_3, a_4]$, $\ldots$, $[a_m, a_{m+1}]$, when the observation value $o_t$ falls into the interval $[a_i, a_{i+1}]$, we quantify $o_t$ as $l_i$, which means that:

$$o_t \rightarrow l_i, \quad when a_i \leq o_t < a_{i+1}. \tag{13}$$

In this way, the continuous observation space can be quantified as the set of limited discrete values: $[l_1, l_2, \ldots, l_m]$. We denote $l_t$ as the discrete value of the observation value $o_t$. The process of quantification reduces the observation space significantly to improve the learning efficiency.

It is worth noting that when the observed value suddenly rises, meaning that the system is suffered from an attack at this time. On the other hand, when the observations change a little in a period of time, meaning that the system operates smoothly without being attacked. Therefore, we utilize the concept of slide window to expand the neighboring observations at one time slot [26]. The size of slide window is denoted as $N$, meaning that there are $N$ recent observation values in one slide window and the window keeps sliding and updating over time. For example, the slide window is $[l_{t-N+1}, l_{t-N+2}, \ldots, l_t]$ at time $t$, and at time $t+1$, the slide window is $[l_{t-N+2}, l_{t-N+3}, \ldots, l_{t+1}]$.

Considering that the size of slide window is $N$ and the size of discrete observation value is $M$, there are $M^N$ possible values of slide observation window. We utilize the slide observation window as the input of our deep reinforcement learning method.

### D. DQND SCHEME

The DQND scheme proposed in this paper mainly includes two stages: the training stage and testing stage. In training stage, the neural networks are trained with the system data, and in testing stage, we test DQND scheme with trained networks. The detailed model of DQND scheme is shown in Figure 3. We define that the number of episodes in training stage as $E$, and each episode consists of three steps:

#### 1) INTERACTING

The agent interacts with environment at this step. The agent first observes the system and obtains the observation $o_t$ at time $t$, which reflects the system operation state. Second, the agent takes action $a_t$ based on the policy that is learned from the third step. Then, the agent updates the action during each networking learning step of the episode. Third, after the interaction, a reward will be received to the agent as the reflection of his action, and then the agent steps into the next state of the interacting and obtains the new observation $o_{t+1}$.

#### 2) REPROCESS

The observation $o_t$, action $a_t$, reward $r_t$ and the next time observation $o_{t+1}$ constitute the transition space $[o_t, a_t, r_t, o_{t+1}]$. Regarding to the slide window, we denote it as:

$$w_t = [l_{t-N+1}, l_{t-N+2}, l_{t-N+3}, \ldots, l_t], \tag{14}$$

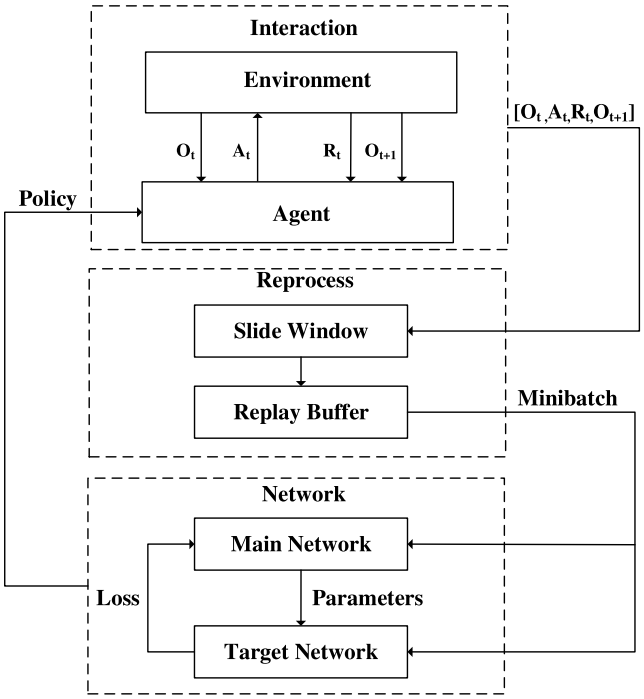


**FIGURE 3.** The model of DQND scheme.

where $l_i$ is the discrete value of observation $o_t$ at time $t$. Therefore, new transition can be denoted as $[w_t, a_t, r_t, w_{t+1}]$. Then, the transition is stored in replay buffer. At each time of iteration, we randomly select a minibatch, whose size is $K$ from the replay buffer as the training data to update the neural networks.

#### 3) NETWORK LEARNING

In our scheme, two types of neural networks are utilized: the main network and the target network. The main network generates a value $Q(o_t, a_t, \theta_t)$ that evaluates the action of agent, and $\theta_t$ is the parameter of the main network. The target network also generates a value $Q'(o_t, a_t, \theta_t')$ that is utilized to generate the loss function based on the next observation. To compute the loss function, we first compute the target $y$ at time $t$ as:

$$y = r + \gamma * \max_{a'} Q'\left(o', a', \theta'\right), \tag{15}$$

in which, $\gamma$ is the discount factor that denotes the proportion of future reward on the state's value and $o'$ is the next observation. The loss function is the square of difference between the target $y$ and value $Q(o_t, a_t, \theta_t)$:

$$L = (y - Q\left(o_t, a_t, \theta_t\right))^2. \tag{16}$$

In each time, the main network is updated in the direction of negative gradient of loss $L$. After every $C$ steps, the parameter of target network $\theta'$ is replaced by the parameter of main network $\theta$ as the updating of target network.

We apply $\epsilon - greedy$ strategy to choose the optimal action from action space. Particularly, the agent chooses action randomly from the action space with probability $1 - \epsilon$, and the agent takes action $a = \arg\max Q\left(s_t, a_t, \theta\right)$ with probability $\epsilon$. Different from the traditional $\epsilon - greedy$ strategy, to balance exploration and exploitation, the value of $\epsilon$ is increased in the process of learning until the $\epsilon$ increases to a maximum $\epsilon_{max}$. The increasing $\epsilon$ strategy ensures that the policy is able to explore more at the beginning of the learning and the action policy converges at the end of the learning with sufficient knowledge of environment.

In the testing stage, the effect of our approach is tested with the trained networks. In each episode of test, we choose the action that minimizes the value of $Q(o_t, a_t, \theta_t)$. It is important to mention that when taking action $a_t$, the training and testing of the episode terminate because the system will stop and the training and testing process will turn into the next episode.

The detailed procedure of training stage of DQND scheme is presented in Algorithm 1. In which, the agent first learns the detection strategy at training stage, the training stage contains several training episodes, and the detection strategy is developed in each episode by interacting between the agent and the environment, data reprocessing and network learning. The testing stage is presented in Algorithm 2. In which, the detection strategy is tested through the trained network to show the performance. During each episode of the test stage, the agent chooses the action that maximize the network output. Once the agent believes that the system is being attacked, the operation of the system will be stopped.

## IV. PERFORMANCE EVALUATION

We show the simulation results of DQND scheme in this section. First, we present the methodology of the evaluation, and then we introduce the results in detail.

### A. EVALUATION METHODOLOGY

#### 1) SIMULATION SETUP

The evaluation is performed on IEEE 9, 14 and 30 bus systems. The initial state vector (phase angles and voltage magnitudes) is determined based on MATPOWER [47]. The system matrix $A$ is set as an identity matrix and the measurement matrix is set according to IEEE-9, 14 and 30 bus respectively.

In DQND scheme, we set the size of slide window $N$ as 4, the discrete observation interval is set as $[0, 0.01]$, $[0.01, 0.05]$, $[0.05, 0.1]$, $[0.1, 1]$. The number of episodes in training stage $E$ is set as 1000 and the number of episodes in testing stage $E_2$ is set as 100. In addition, there are 100 tests in one episodes. The time steps in one episode is set as 200. The learning rate in reinforcement learning is set as 0.001, the $\epsilon$ is set as 0.7 initially and the maximum of $\epsilon$ is set as 0.99. The time interval of updating the target network $C$ is set as 5. The size of replay buffer is set as 500 and the minibatch size is set as 32. In the main network, we set 2 hidden layers and a fully connected layer and there are 100 nodes in each layer. The target network follows the same structure as the main network.

In attack model, we apply the continuous attack model in training stage, the continuous attack is launched since the

**Algorithm 1** The Training Stage of DQND

**Input**: The number of time steps $T$ during one episode, training episodes $E$.
Initialize $RM$, $Q$ with random parameters $\theta$.
Initialize $Q'$ with parameters $\theta'$.
Initialize $\epsilon$, $\epsilon_{increase}$ and $\epsilon_{max}$.
**Output**: The defending strategy.

**for** *episode* $= 1$ *to* $E$ **do**
  **for** *step* $t = 1$ *to* $T$ **do**
    Collect measurement vector $y_t$;
    Estimate the state of system $\hat{x}_t$ by Equation (3) and Equation (4);
    Compute the observation $o_t$ by Equation (12);
    Compute the discrete observation value by Equation (13);
    $\epsilon = \epsilon + \epsilon_{increase}$;
    **if** $\epsilon > \epsilon_{max}$ **then**
      $\epsilon = \epsilon_{max}$;
    **end**
    Generating a random number $ra$ from (0, 1).
    **if** $ra < \epsilon$ **then**
      $a_t =$ a random action in action space.
    **else**
      $a_t = argmaxQ(s_t, a_t, \theta)$
    **end**
    **if** $s_t = s_a$ **then**
      **if** $a_t = a_s$ **then**
        $r_t = 0$
      **else**
        $r_t = c_1 * |t - \lambda|$
      **end**
    **else**
      **if** $a_t = a_s$ **then**
        $c_2 * \frac{\|y_t - h(\hat{x}_t)\|}{\|w\|}$
      **else**
        $r_t = 0$
      **end**
      Takes action $a_t$ and observe next observation $o_{t+1}$.
      Compute the slide window $w_t$ and $w_{t+1}$.
      Store $(w_t, a_t, r_t, w_{t+1})$ in replay buffer.
      Sample random minibatch $(o_t, a_t, r_t, o_{t+1})$ from replay buffer.
      **if** $t = T$ **then**
        $y_j = r_j$;
      **else**
        $y_j = r_j + \gamma \max Q'(s_j, a_j; \theta_i)$;
      **end**
      Compute loss function $L = (y - Q(o_t, a_t, \theta_t))^2$;
      Perform a gradient descent step on the loss function $L$
      Every $C$ steps update parameters $\theta'$ as the main network parameters $\theta$.
      **if** $a_t = a_s$ **then**
        Go to the next training episode.
      **end**
    **end**
  **end**
**end**

**Algorithm 2** The Testing Stage of DQND

**Input**: The trained neural network $Q$.
The number of time steps $T$ in one episode, the number of test episodes $E_2$.
**Output**: The testing result.

**for** *episode* $= 1$ *to* $E_2$ **do**
  **for** *step* $t = 1$ *to* $T$ **do**
    Collect the measurement vector $y_t$;
    Employ the Gauss-Newton iterative algorithm to estimate the state of system $\hat{x}_t$ by Equation (3) and Equation (4);
    Compute the observation $o_t$ by Equation (12);
    Compute the discrete observation value by Equation (13);
    $a_t = argmaxQ(s_t, a_t, \theta)$
    **if** $a_t = a_s$ **then**
      Go to the next training episode.
    **end**
  **end**
**end**

time slot 100, and the attack vector $a$ is a uniform variable $\pm U[-0.1, 0.1]$. In testing stage, we apply both the continuous attack model and the discontinuous attack model, the time when continuous attack is launched is a uniform variable $U[40, 140]$ and the size of continuous attack is also a uniform variable $U[-0.1, 0.1]$. The discontinuous attack is launched at time 100, and the probability of an attack occurred at each time step is 0.5, the size of discontinuous attack is the same as that of continuous attack model.

### 2) EVALUATION METRICS

To demonstrate the performance of our scheme, we present several evaluation metrics as follows:

- **Delay-alarm error rate (DAE):** We define the size of delay-alarm error as $t - \lambda$, and the delay-alarm error rate is defined as the sum of delay-alarm error in all episodes divided by the number of tests. $t$ is the time when the agent detects the attack and $\lambda$ is the time when the attack is launched initially.
- **False-alarm error rate (FAE):** We define the size of false-alarm error as $\lambda - t$, and the false-alarm error rate is denoted as the sum of false-alarm error in all episodes divided by the number of tests.
- **Detect-failure rate (DF):** The detect-failure rate is defined as the sum of the times of detect-failure occurred in all episodes divided by the number of tests.

### 3) BENCHMARKS

We compare the DQND scheme with the following benchmarks to demonstrate the effectiveness:

- **SARSA scheme** is an online reinforcement learning method and Kurt *et al.* employs it to detect data integrity attack in power system [26]. A table is set up to store the

value of actions in each state and the system operator can learn the optimal defending strategy during the updating of the table.

- $SARSA_{imp}$ **scheme** is an enhanced scheme based on SARSA. First, the observation is defined as Equation (12), which eliminates the effects of noise. Second, the reward function is denoted as Equation (10) and Equation (11), which reflects the influence of delay-alarm error and false-alarm error.

### B. EVALUATION RESULTS

We first compare DQND scheme with SARSA and $SARSA_{imp}$ under continuous attack model. In Table 2, we compare three schemes in terms of delay-alarm error rates on IEEE 9, 14 and 30 bus system. In Table 3, we compare three approaches with detect-failure rates on the three bus systems.

**TABLE 2. Delay-alarm error rates in continuous attack model among SARSA, $SARSA_{imp}$, and DQND scheme.**

| Schemes \ Systems | IEEE-9 | IEEE-14 | IEEE-30 |
|---|---|---|---|
| $SARSA$ | 6.4246 | 1.3584 | 12.4223 |
| $SARSA_{imp}$ | 0.6639 | 0.2164 | 4.2132 |
| $DQND$ | 0.0237 | 0.0240 | 0.1249 |

**TABLE 3. Detect-failure rates in continuous attack model among SARSA, $SARSA_{imp}$ and DQND schemes.**

| Schemes \ Systems | IEEE-9 | IEEE-14 | IEEE-30 |
|---|---|---|---|
| $SARSA$ | 0 | 0.0083 | 0.0001 |
| $SARSA_{imp}$ | 0 | 0 | 0 |
| $DQND$ | 0 | 0 | 0 |

From Table 2, we can observe that DQND scheme achieves the lowest delay-alarm error rates in all three IEEE bus systems when the system is under continuous attack, which indicates that our scheme is able to detect the attack in the shortest time compared with the baseline schemes. Furthermore, we can also conclude that $SARSA_{imp}$ performs better than SARSA, since that the observation and reward function are improved in $SARSA_{imp}$.

Table 3 illustrates the accuracy of DQND scheme in detecting continuous attack. From the table, we can observe that there is no detect-failure in the test of DQND scheme in all three IEEE bus systems, as well as $SARSA_{imp}$ scheme. In contrast, several detect failures are existed in the test of $SARSA$ scheme. In summary, we can conclude that $DQND$ and $SARSA_{imp}$ perform better in terms of detection accuracy when defend against continuous attack.

In Table 2 and Table 3, we compare the accuracy and detection speed of three detection schemes when the systems are under continuous attack. In addition, we compare these three schemes under discontinuous attack in Table 4 and Table 5.

**TABLE 4. Delay-alarm error rates in discontinuous attack model among SARSA, $SARSA_{imp}$, and DQND scheme.**

| Schemes \ Systems | IEEE-9 | IEEE-14 | IEEE-30 |
|---|---|---|---|
| $SARSA$ | 16.8176 | 5.6829 | 7.2793 |
| $SARSA_{imp}$ | 1.0500 | 0.4972 | 2.3532 |
| $DQND$ | 0.1357 | 0.0490 | 1.4430 |

**TABLE 5. Detect-failure rates in discontinuous attack model among SARSA, $SARSA_{imp}$, and DQND scheme.**

| Schemes \ Systems | IEEE-9 | IEEE-14 | IEEE-30 |
|---|---|---|---|
| $SARSA$ | 0.0150 | 0.0128 | 0 |
| $SARSA_{imp}$ | 0 | 0 | 0 |
| $DQND$ | 0 | 0 | 0 |

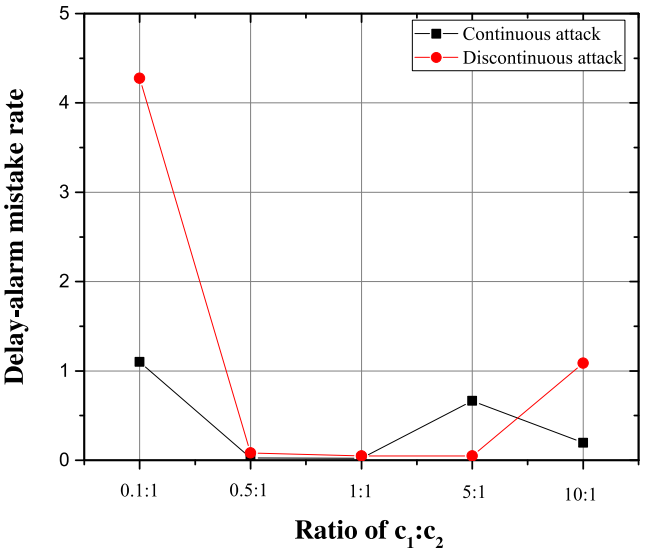


**FIGURE 4. Delay-alarm error rates with the change of the ratio of $c_1 : c_2$.**

From Table 4, we observe that DQND scheme achieves the lowest delay-alarm error rates in all three IEEE bus systems when the system is under discontinuous attack, while SARSA scheme spends the most time to detect existence of the attack.

The evaluation results in Table 5 are similar to the results in Table 3, there is also no detect-failure in the results of DQND and $SARSA_{imp}$ schemes when there exists discontinuous attack in AC power system. However, there exists detect-failure in the test of SARSA scheme in IEEE-9 and IEEE-14 bus systems.

In addition, we study the relationship between the performance of our scheme and the ratio of $c_1 : c_2$. Since that the false-alarm error rates and detect-failure rates are both very small and close to 0, we consider to show the variation of delay-alarm error rates. For simplicity, we take 9 bus system as example, the evaluation result is shown in Figure 4. From the figure, we observe that when the ratio of $c_1 : c_2$ is

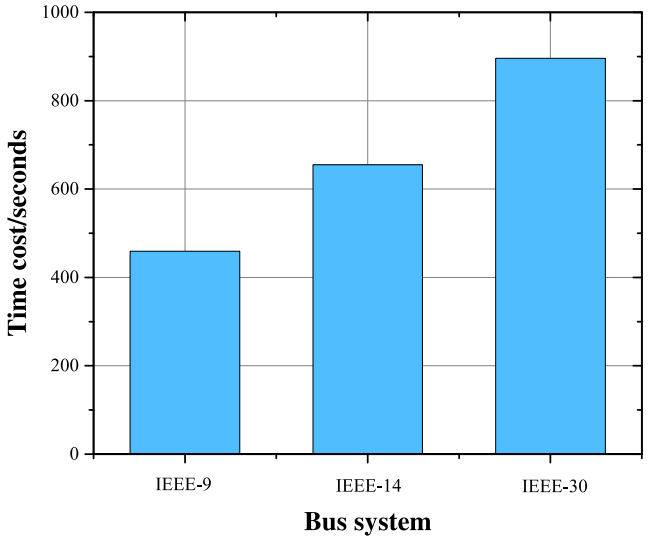


**FIGURE 5.** Time cost.

around 1 : 1, the delay-alarm error rates is the lowest. Note that when the value of $c_1$ : $c_2$ is too large or too small, the delay-alarm error rates is very high and the algorithm cannot learn the optimal detection policy. The reason behind this is that the algorithm cannot detect the attack fast when $c_1$ is small, and the algorithm cannot ensure the accuracy of detection when $c_1$ is too large. We can also observe that the delay-alarm error rates under continuous attack is smaller than that under discontinuous attack, meaning that continuous attack is easier to detect than discontinuous attack.

Finally, we evaluate the time cost of our scheme in Figure 5, the time cost of the DQND scheme in IEEE 9, 14 and 30 bus systems are 459 seconds, 655 seconds and 896 seconds, respectively. The results demonstrate that the time consumption increases as the complexity of the system increases. Nonetheless, by leveraging the advanced computing technologies, i.e., distributed and parallel computing, the time consumption of the proposed approach can be reduced significantly.

## V. DISCUSSION

We now discuss some future directions of the work in this paper, regarding to the application scenarios and advanced reinforcement learning strategies.

- **Application modules:** In this paper, we propose a deep Q-learning based approach to detect the cyber-attack in AC power system state estimation in smart grid. As it is reported that other critical modules, i.e., dynamic state estimation, economic dispatch, load frequency control are also suffered from the threatens from the cyber space. Moreover, it is difficult to investigate a generical strategy to detect the data integrity attacks in those modules. To this end, as a future direction, it is an urgent need to investigate suitable deep reinforcement learning approaches that fully integrates the cyber-physical characteristics of those modules in the smart grid to assist the system operator making optimal action strategy. We shall also investigate the application of our designed scheme in other CPS [40], [41].
- **Advanced reinforcement learning strategy:** In this paper, a deep-Q-learning based detection scheme has been proposed to defend against data integrity attacks in the smart grid. Although our scheme achieves better results than the baseline schemes in terms of detection accuracy and rapidity, the time overhead of our scheme is still relatively large. Especially, as the number of nodes in the power system increases, the time consumption grows exponentially, and the practicability of the strategy will be affected. In view of this, improving the convergence speed and designing a more stable neural network to learn the optimal policy are considered as ongoing works in investigating deep reinforcement learning-based detection to deal with data integrity attacks.

## VI. CONCLUSION

In this paper, we addressed the issue of defending against data integrity attacks in AC power system state estimation. We first formulated the model of AC power system and the data integrity attack model. To detect data integrity attacks, we proposed a DQND scheme to learn the optimal defending strategy. Specifically, DQND scheme applies a main network and a target network to learn the detection strategy during the training stage with real data. To improve the learning efficiency, we applied the quantification of observation space and the concept of slide window, which could prevent the curse of dimension. Finally, we evaluated our scheme based on IEEE 9, 14 and 30 bus systems, respectively. We validated our scheme with two attack models in the evaluation: continuous attack model and discontinuous attack model. The evaluation results show that our scheme achieves higher detection accuracy and speed compared with two baseline schemes.

## ACKNOWLEDGEMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the agencies.

## REFERENCES

[1] S. Chen, S. Song, L. Li, and J. Shen, "Survey on smart grid technology," *Power Syst. Technol.*, vol. 33, no. 8, pp. 1–7, Apr. 2009.
[2] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid—The new and improved power grid: A survey," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 944–980, 4th Quart., 2012.
[3] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart grid technologies: Communication technologies and standards," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529–539, Nov. 2011.
[4] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Security Privacy*, vol. 8, no. 1, pp. 81–85, Jan./Feb. 2010.
[5] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Security Privacy*, vol. 7, no. 3, pp. 75–77, Jun. 2009.
[6] R. Deng, G. Xiao, and R. Lu, "Defending against false data injection attacks on power system state estimation," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 198–207, Feb. 2017.

[7] Y. Mo, T. H.-J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–physical security of a smart grid infrastructure," *Proc. IEEE*, vol. 100, no. 1, pp. 195–209, Jan. 2012.
[8] Q. Yang, D. Li, W. Yu, Y. Liu, D. An, X. Yang, and J. Lin, "Toward data integrity attacks against optimal power flow in smart grid," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1726–1738, Oct. 2017.
[9] Q. Yang, D. An, R. Min, W. Yu, X. Yang, and W. Zhao, "On optimal PMU placement-based defense against data integrity attacks in smart grid," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1735–1750, Jul. 2017.
[10] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, and P. Li, "Energy-efficient scheduling for real-time systems based on deep Q-learning model," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 1, pp. 132–141, Mar. 2019.
[11] Z. Cheng, Q. Zhao, F. Wang, Y. Jiang, L. Xia, and J. Ding, "Satisfaction based Q-learning for integrated lighting and blind control," *Energy Buildings*, vol. 127, pp. 43–55, Sep. 2016.
[12] D. Kundur, X. Feng, S. Liu, T. Zourntos, and K. L. Butler-Purry, "Towards a framework for cyber attack impact analysis of the electric smart grid," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 244–249.
[13] X. Li, X. Liang, R. Lu, H. Zhu, X. Lin, and X. Shen, "Securing smart grid: Cyber attacks, countermeasures, and challenges," *IEEE Commun. Mag.*, vol. 50, no. 8, pp. 38–45, Aug. 2012.
[14] D. Wei, Y. Lu, M. Jafari, P. Skare, and K. Rohde, "An integrated security system of protecting smart grid against cyber attacks," in *Proc. Innov. Smart Grid Technol. (ISGT)*, Jan. 2010, pp. 1–7.
[15] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla, "Smart grid data integrity attacks: Characterizations and countermeasures$^{\pi}$," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Oct. 2011, pp. 232–237.
[16] Q. Yang, L. Chang, and W. Yu, "On false data injection attacks against Kalman filtering in power system dynamic state estimation," *Secur. Commun. Netw.*, vol. 9, no. 9, pp. 833–849, 2016.
[17] A. Hahn and M. Govindarasu, "Cyber attack exposure evaluation framework for the smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 835–843, Dec. 2011.
[18] S. Sridhar and G. Manimaran, "Data integrity attack and its impacts on voltage control loop in power grid," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2011, pp. 1–6.
[19] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 998–1010, 4th Quart., 2012.
[20] X. Liu and Z. Li, "False data attacks against AC state estimation with incomplete network information," *IEEE Trans. Smart Grid.*, vol. 8, no. 5, pp. 2239–2248, Sep. 2017.
[21] A. Ashok, M. Govindarasu, and V. Ajjarapu, "Online detection of stealthy false data injection attacks in power system state estimation," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1636–1646, May 2018.
[22] S. Bi and Y. Zhang, "False-data injection attack to control real-time price in electricity market," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 772–777.
[23] 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*, vol. 10, no. 2, pp. 2158–2169, Mar. 2018.
[24] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1644–1652, Sep. 2017.
[25] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4565–4573.
[26] M. Kurt, O. Ogundijo, C. Li, and X. Wang, "Online cyber-attack detection in smart grid: A reinforcement learning approach," *IEEE Trans. Smart Grid*, to be published.
[27] J. Peng and R. Williams, "Incremental multi-step Q-learning," in *Machine Learning*. Amsterdam, The Netherlands: Elsevier, 1994, pp. 226–232.
[28] P. Glorennec and L. Jouffe, "Fuzzy Q-learning," in *Proc. 6th Int. Fuzzy Syst. Conf.*, vol. 2, Jul. 1997, pp. 659–662.
[29] H. V. Hasselt, "Double Q-learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2613–2621.
[30] G. Lee, Y. Kim, and J. Kang, "An adaptive dos attack mitigation measure for field networks in smart grids," in *Advances on Broad-Band Wireless Computing, Communication and Applications*, L. Barolli, F. Xhafa, and K. Yim, Eds. Cham, Switzerland: Springer, 2017, pp. 419–428.
[31] X. Liu and Z. Li, "False data attacks against ac state estimation with incomplete network information," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2239–2248, Sep. 2016.
[32] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, p. 13, 2011.
[33] H. Sandberg, A. Teixeira, and K. Johansson, "On security indices for state estimators in power networks," in *Proc. 1st Workshop Secure Control Syst. (SCS)*, Stockholm, Sweden, 2010, pp. 1–6.
[34] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, and M. G. Bellemare, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.
[35] K. Narasimhan, A. Yala, and R. Barzilay, "Improving information extraction by acquiring external evidence with reinforcement learning," 2016, *arXiv:1603.07954*. [Online]. Available: https://arxiv.org/abs/1603.07954
[36] S. Pal, B. Sikdar, and J. H. Chow, "Classification and detection of PMU data manipulation attacks using transmission line parameters," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5057–5066, Sep. 2018.
[37] Z.-H. Pang, G.-P. Liu, D. Zhou, F. Hou, and D. Sun, "Two-channel false data injection attacks against output tracking control of networked systems," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3242–3251, May 2016.
[38] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of Markov decision processes," *Math. Oper. Res.*, vol. 12, no. 3, pp. 441–450, 1987.
[39] X. Yang, X. Zhang, J. Lin, W. Yu, and P. Zhao, "A Gaussian-mixture model based detection scheme against data integrity attacks in the smart grid," in *Proc. 25th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2016, pp. 1–9.
[40] J. Lin, W. Yu, N. Zhang, X. Yang, and L. Ge, "Data integrity attacks against dynamic route guidance in transportation-based cyber-physical systems: Modeling, analysis, and defense," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8738–8753, Sep. 2018.
[41] H. Xu, W. Yu, D. Griffith, and N. Golmie, "A survey on industrial Internet of Things: A cyber-physical systems perspective," *IEEE Access*, vol. 6, pp. 78238–78259, 2018.
[42] M. A. Rahman, E. Al-Shaer, and R. Kavasseri, "Impact analysis of topology poisoning attacks on economic operation of the smart power grid," in *Proc. IEEE 34th Int. Conf. Distrib. Comput. Syst.*, Jun./Jul. 2014, pp. 649–659.
[43] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, Jan. 2016.
[44] P. Su, M. Gasic, N. Mrksic, L. Rojas-Barahona, S. Ultes, D. Vandyke, T.-H. Wen, and S. Young, "On-line active reward learning for policy optimisation in spoken dialogue systems," 2016, *arXiv:1605.07669*. [Online]. Available: https://arxiv.org/abs/1605.07669
[45] D. Wang, X. Guan, T. Liu, Y. Gu, C. Shen, and Z. Xu, "Extended distributed state estimation: A detection method against tolerable false data injection attacks in smart grids," *Energies*, vol. 7, no. 3, pp. 1517–1538, 2014.
[46] J. Zhang, Z. Chu, L. Sankar, and O. Kosut, "False data injection attacks on power system state estimation with limited information," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Jul. 2016, pp. 1–5.
[47] R. Zimmerman and C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2010.
[48] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.
[49] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao, "On false data-injection attacks against power system state estimation: Modeling and countermeasures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 717–729, Mar. 2014.
[50] J. Lin, W. Yu, X. Yang, G. Xu, and W. Zhao, "On false data injection attacks against distributed energy routing in smart grid," in *Proc. IEEE/ACM 3rd Int. Conf. Cyber-Phys. Syst.*, Washington, DC, USA, Apr. 2012, pp. 183–192.
[51] W. Yu, D. Griffith, L. Ge, S. Bhattarai, and N. Golmie, "An integrated detection system against false data injection attacks in the smart grid," *Secur. Commun. Netw.*, vol. 8, no. 2, pp. 91–109, 2015.

[52] J. Lin, W. Yu, and X. Yang, "Towards multistep electricity prices in smart grid electricity markets," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 286–302, Jan. 2016.
[53] W. G. Hatcher and W. Yu, "A survey of deep learning: Platforms, applications and emerging research trends," *IEEE Access*, vol. 6, pp. 24411–24432, 2018.
[54] X. Liu, C. Qian, W. G. Hatcher, H. Xu, W. Liao, and W. Yu, "Secure Internet of things (IoT)-based smart-world critical infrastructures: Survey, case study and research opportunities," *IEEE Access*, vol. 7, pp. 79523–79544, 2019.
[55] J. Zhao, L. Mili, and M. Wang, "A generalized false data injection attacks against power system nonlinear state estimator and countermeasures," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 4868–4877, Sep. 2018.

**DOU AN** received the B.S. degree in mathematics and applied mathematics from Northwestern Polytechnical University, Xi'an, China, in 2011, and the Ph.D. degree with the Department of Automation Science and Technology from Xi'an Jiaotong University, Xi'an, in 2017, where he is currently a Lecturer with the Department of Automation Science and Technology, School of Electronics and Information Engineering. His research interests include cyber-physical systems, smart grid security and privacy, and incentive mechanisms design for smart grid.

**QINGYU YANG** received the B.S. and M.S. degrees in mechatronics engineering and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, China, in 1996, 1999, and 2003, respectively, where he is currently a Professor with the School of Electronics and Information Engineering and also with the State Key Laboratory for Manufacturing System Engineering. His current research interests include cyber-physical systems, power grid security, control and diagnosis of mechatronic system, and intelligent control of industrial process.

**WENMAO LIU** received the Ph.D. degree in information security from the Harbin Institute of Technology, in 2013. He served as a Researcher with NSFOCUS Inc. During the first two years in NSFOCUS, he was also with Tsinghua University as a Postdoctoral. He is currently the Director of the Innovation Center, NSFOCUS. He has published a book *Software-Defined Security, in the next generation inspired by SDN/NFV technology* and participate cloud security related national and industrial standards. His research interests include network security, cloud security, the IoT security, threat intelligence, and advanced security analytics. Now he has been promoting the adoption of container security, and DevSecOps.

**YANG ZHANG** received the B.S. degree in automation science and technology from Xi'an Jiaotong University, Xi'an, China, in 2018, where he is currently pursuing the Ph.D. degree with the Department of Automation Science and Technology, School of Electronics and Information Engineering. His research interests include cyber-physical systems, incentive mechanisms design for the IoT/smart grid, and reinforcement learning.