IEEE TRANSACTIONS ON
SMART GRID

http://ieeexplore.ieee.org/Xplore

Online Cyber-Attack Detection in Smart Grid: A Reinforcement Learning Approach

Mehmet Necip Kurt, Oyetunji Ogundijo, *Student Member, IEEE*, Chong Li, *Member, IEEE*,
and Xiaodong Wang, *Fellow, IEEE*

***Abstract*—Early detection of cyber-attacks is crucial for a safe and reliable operation of the smart grid. In the literature, outlier detection schemes making sample-by-sample decisions and online detection schemes requiring perfect attack models have been proposed. In this paper, we formulate the online attack/anomaly detection problem as a partially observable Markov decision process (POMDP) problem and propose a universal robust online detection algorithm using the framework of model-free reinforcement learning (RL) for POMDPs. Numerical studies illustrate the effectiveness of the proposed RL-based algorithm in timely and accurate detection of cyber-attacks targeting the smart grid.**

***Index Terms*—Smart grid, model-free reinforcement learning, partially observable Markov decision process (POMDP), cyber-attack, online detection, Kalman filter.**

I. Introduction

A. Background and Related Work

The next generation power grid, i.e., the smart grid, relies on advanced control and communication technologies. This critical cyber infrastructure makes the smart grid vulnerable to hostile cyber-attacks [1]–[3]. Main objective of attackers is to damage/mislead the state estimation mechanism in the smart grid to cause wide-area power blackouts or to manipulate electricity market prices [4]. There are many types of cyber-attacks, among them false data injection (FDI), jamming, and denial of service (DoS) attacks are well known. FDI attacks add malicious fake data to meter measurements [5]–[8], jamming attacks corrupt meter measurements via additive noise [9], and DoS attacks block the access of system to meter measurements [8], [10], [11].

The smart grid is a complex network and any failure or anomaly in a part of the system may lead to huge damages on the overall system in a short period of time. Hence, early detection of cyber-attacks is critical for a timely and effective response. In this context, the framework of quickest change detection [12]–[15] is quite useful. In the quickest change detection problems, a change occurs in the sensing environment at an unknown time and the aim is to detect the change as soon as possible with the minimal level of false alarms based on the measurements that become available sequentially over time. After obtaining measurements at a given time, decision maker either declares a change or waits for the next time interval to have further measurements. In general, as the desired detection accuracy increases, detection speed decreases. Hence, the stopping time, at which a change is declared, should be chosen to optimally balance the tradeoff between the detection speed and the detection accuracy.

If the probability density functions (pdfs) of meter measurements for the pre-change, i.e., normal system operation, and the post-change, i.e., after an attack/anomaly, cases can be modeled sufficiently accurately, the well-known cumulative sum (CUSUM) test is the optimal online detector [16] based on the Lorden's criterion [17]. Moreover, if the pdfs can be modeled with some unknown parameters, the generalized CUSUM test, which makes use of the estimates of unknown parameters, has asymptotic optimality properties [13]. However, CUSUM-based detection schemes require perfect models for both the pre- and post-change cases. In practice, capabilities of an attacker and correspondingly attack types and strategies can be totally unknown. For instance, an attacker can arbitrarily combine and launch multiple attacks simultaneously or it can launch a new unknown type of attack. Then, it may not be always possible to know the attacking strategies ahead of time and to accurately model the post-change case. Hence, universal detectors, not requiring any attack model, are needed in general. Moreover, the (generalized) CUSUM algorithm has optimality properties in minimizing a least favorable (worst-case) detection delay subject to false alarm constraints [13], [16]. Since the worst case detection delay is a pessimistic metric, it is, in general, possible to obtain algorithms performing better than the (generalized) CUSUM algorithm.

Considering the pre-change and the post-change cases as hidden states due to the unknown change-point, a quickest change detection problem can be formulated as a partially observable Markov decision process (POMDP) problem. For the problem of online attack/anomaly detection in the smart grid, in the pre-change state, the system is operated under normal conditions and using the system model, the pre-change measurement pdf can be specified highly accurately. On the other hand, the post-change measurement pdf can take different unknown forms depending on the attacker's strategy. Furthermore, the transition probability between the hidden states is unknown in general. Hence, the exact model of the POMDP is unknown.

Reinforcement learning (RL) algorithms are known to be effective in controlling uncertain environments. Hence, the described POMDP problem can be effectively solved using RL. In particular, as a solution, either the underlying POMDP model can be learned and then a model-based RL algorithm for POMDPs [18], [19] can be used or a model-free RL algorithm [20]–[24] can be used without learning the underlying model. Since the model-based approach requires a two-step solution that is computationally more demanding and only an approximate model can be learned in general, we prefer to use the model-free RL approach.

This work was supported in part by the U.S. National Science Foundation (NSF) under Grant ECCS-1405327, and in part by the U.S. Office of Naval Research under Grant N000141410667.

The authors are with the Department of Electrical Engineering, Columbia University, New York, NY 10027, USA (e-mail: m.n.kurt@columbia.edu; oeo2109@columbia.edu; cl3607@columbia.edu; wangx@ee.columbia.edu).

Digital Object Identifier: 10.1109/TSG.2018.2878570

Outlier detection schemes such as the Euclidean detector [25] and the cosine-similarity metric based detector [26] are universal as they do not require any attack model. They mainly compute a dissimilarity metric between actual meter measurements and predicted measurements (by the Kalman filter) and declare an attack/anomaly if the amount of dissimilarity exceeds a certain predefined threshold. However, such detectors do not consider temporal relation between attacked/anomalous measurements and make sample-by-sample decisions. Hence, they are unable to distinguish instantaneous high-level random system noise from long-term (persistent) anomalies caused, e.g., by an unfriendly intervention to the system. Hence, compared to the outlier detection schemes, more reliable universal attack detection schemes are needed.

In this work, we consider the smart grid security problem from the defender's perspective and seek for an effective detection scheme using RL techniques (single-agent RL). Note that the problem can be considered from an attacker's perspective as well, where the objective would be to determine the attacking strategies leading to the maximum possible damage on the system. Such a problem can be particularly useful in vulnerability analysis, i.e., to identify the worst possible damage an attacker may introduce to the system and accordingly to take necessary precautions. In the literature, several studies investigate vulnerability analyses using RL, see e.g., [27] for FDI attacks and [28] for sequential network topology attacks. We further note that the problem can also be considered from both defender's and attacker's perspectives simultaneously, that corresponds to a game-theoretic setting.

Extension of single-agent RL to multiple agents is the multi-agent RL framework that quite involves game theory since in this case, the optimal policies of agents depend both on the environment and the policies of the other agents. Moreover, stochastic games extend the Markov decision processes to multi-agent case where the game is sequential and consists of multiple states, and the transition from one state to another and also the payoffs (rewards/costs) depend on joint actions of all agents. Several RL-based solution approaches have been proposed for stochastic games, see e.g., [29]–[33]. Further, if the game (the underlying state of the environment, actions and payoffs of other agents, etc.) is partially observed, then it is called a partially observable stochastic game, for which finding a solution is more difficult in general.

B. Contributions

In this paper, we propose an online cyber-attack detection algorithm using the framework of model-free RL for POMDPs. The proposed algorithm is universal, i.e., it does not require attack models. This makes the proposed scheme widely applicable and also proactive in the sense that new unknown attack types can be detected. Since we follow a model-free RL approach, the defender learns a direct mapping from observations to actions (*stop* or *continue*) by trial-and-error. In the training phase, although it is possible to obtain/generate observation data for the pre-change case using the system model under normal operating conditions, it is generally difficult to obtain real attack data. For this reason, we follow a robust detection approach by training the defender with low-magnitude attacks that corresponds to the worst-case scenarios from a defender's perspective since such attacks are quite difficult to detect. Then, the trained defender becomes sensitive to detect slight deviations of meter measurements from the normal system operation. The robust detection approach significantly limits the action space of an attacker as well. That is, to prevent the detection, an attacker can only exploit very low attack magnitudes that are practically not much of interest due to their minimal damage on the system. To the best of our knowledge, this work is the first attempt for online cyber-attack detection in the smart grid using RL techniques.

Symbol	Meaning
Γ	Stopping time
τ	Change-point
I	Number of quantization levels
M	Window size
$Q(o, a)$	Q-value corresponding to observation-action pair (o, a)
α	Learning rate
ϵ	Exploration rate
T	Maximum length of a learning episode
E	Number of learning episodes

TABLE I: Common symbols/parameters in the paper.

C. Organization and Notation

We introduce the system model and the state estimation mechanism in Sec. II. We present the problem formulation in Sec. III and the proposed solution approach in Sec. IV. We then illustrate the performance of the proposed RL-based detection scheme via extensive simulations in Sec. V. Finally, we conclude the paper in Sec. VI. Boldface letters denote vectors and matrices, all vectors are column vectors, and $\boldsymbol{o}^{\mathrm{T}}$ denotes the transpose of $\boldsymbol{o}$. Further, P and E denote the probability and expectation operators, respectively. Table I summarizes the common symbols and parameters used in the paper.

II. System Model and State Estimation

A. System Model

Suppose that there are K meters in a power grid consisting of $N+1$ buses, where usually $K > N$ to have the necessary measurement redundancy against noise [34]. One of the buses is considered as a reference bus and the system state at time t is denoted with $\mathbf{x}_t = [x_{1,t}, \dots, x_{N,t}]^{\mathrm{T}}$ where $x_{n,t}$ denotes the phase angle at bus n at time t. Let the measurement taken at meter k at time t be denoted with $y_{k,t}$ and the measurement vector be denoted with $\mathbf{y}_t = [y_{1,t}, \dots, y_{K,t}]^{\mathrm{T}}$. Based on the widely used linear DC model [34], we model the smart grid with the following state-space equations:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{v}_t, \tag{1}$$

$$\mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{w}_t, \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the system (state transition) matrix, $\mathbf{H} \in \mathbb{R}^{K \times N}$ is the measurement matrix determined based on the network topology, $\mathbf{v}_t = [v_{1,t}, \dots, v_{N,t}]^{\mathrm{T}}$ is the process noise vector, and $\mathbf{w}_t = [w_{1,t}, \dots, w_{K,t}]^{\mathrm{T}}$ is the measurement noise vector. We assume that $\mathbf{v}_t$ and $\mathbf{w}_t$ are independent additive white Gaussian random processes where $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \sigma_v^2 \mathbf{I}_N)$, $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I}_K)$, and $\mathbf{I}_K \in \mathbb{R}^{K \times K}$ is an identity matrix. Moreover, we assume that the system is observable, i.e., the observability matrix

$$\mathbf{O} \triangleq \begin{bmatrix} \mathbf{H} \\ \mathbf{HA} \\ \vdots \\ \mathbf{HA}^{N-1} \end{bmatrix}$$

has rank N.

The system model given in (1) and (2) corresponds to the normal system operation. In case of a cyber-attack, however, the measurement model in (2) is no longer true. For instance,

1) in case of an FDI attack launched at time τ, the measurement model can be written as
$$\mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{w}_t + \mathbf{b}_t \mathbb{1}\{t \geq \tau\},$$
where $\mathbb{1}$ is an indicator function and $\mathbf{b}_t \triangleq [b_{1,t}, \ldots, b_{K,t}]^{\mathrm{T}}$ denotes the injected malicious data at time $t \geq \tau$ and $b_{k,t}$ denotes the injected false datum to the kth meter at time t,
2) in case of a jamming attack with additive noise, the measurement model can be written as
$$\mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{w}_t + \mathbf{u}_t \mathbb{1}\{t \geq \tau\},$$
where $\mathbf{u}_t \triangleq [u_{1,t}, \ldots, u_{K,t}]^{\mathrm{T}}$ denotes the random noise realization at time $t \geq \tau$ and $u_{k,t}$ denotes the jamming noise corrupting the kth meter at time t,
3) in case of a hybrid FDI/jamming attack [9], the meter measurements take the following form:
$$\mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{w}_t + (\mathbf{b}_t + \mathbf{u}_t) \mathbb{1}\{t \geq \tau\},$$
4) in case of a DoS attack, meter measurements can be partially unavailable to the system controller. The measurement model can then be written as
$$\mathbf{y}_t = \mathbf{D}_t(\mathbf{H}\mathbf{x}_t + \mathbf{w}_t),$$
where $\mathbf{D}_t = \mathrm{diag}(d_{1,t}, \ldots, d_{K,t})$ is a diagonal matrix consisting of 0s and 1s. Particularly, if $y_{k,t}$ is available, then $d_{k,t} = 1$, otherwise $d_{k,t} = 0$. Note that $\mathbf{D}_t = \mathbf{I}_K$ for $t < \tau$,
5) in case of a network topology attack, the measurement matrix changes. Denoting the measurement matrix under topology attack at time $t \geq \tau$ by $\bar{\mathbf{H}}_t$, we have
$$\mathbf{y}_t = \begin{cases} \mathbf{H}\mathbf{x}_t + \mathbf{w}_t, & \text{if } t < \tau \\ \bar{\mathbf{H}}_t\mathbf{x}_t + \mathbf{w}_t, & \text{if } t \geq \tau, \end{cases}$$
6) in case of a mixed topology and hybrid FDI/jamming attack, the measurement model can be written as follows:
$$\mathbf{y}_t = \begin{cases} \mathbf{H}\mathbf{x}_t + \mathbf{w}_t, & \text{if } t < \tau \\ \bar{\mathbf{H}}_t\mathbf{x}_t + \mathbf{w}_t + \mathbf{b}_t + \mathbf{u}_t, & \text{if } t \geq \tau. \end{cases}$$

B. State Estimation

Since the smart grid is regulated based on estimated system states, state estimation is a fundamental task in the smart grid, that is conventionally performed using the static least squares (LS) estimators [5], [6], [35]. However, in practice, the smart grid is a highly dynamic system due to time-varying load and power generation [36]. Furthermore, time-varying cyber-attacks can be designed and performed by the adversaries. Hence, dynamic system modeling as in (1) and (2) and correspondingly using a dynamic state estimator can be quite useful for real-time operation and security of the smart grid [8], [9].

For a discrete-time linear dynamic system, if the noise terms are Gaussian, the Kalman filter is the optimal linear estimator in minimizing the mean squared state estimation error [37]. Note that for the Kalman filter to work correctly, the system needs to be observable. The Kalman filter is an online estimator consisting of prediction and measurement update steps at each iteration. Denoting the state estimates at time t with $\hat{\mathbf{x}}_{t|t'}$ where $t' = t-1$ and $t' = t$ for the prediction and measurement update steps, respectively, the Kalman filter equations at time t can be written as follows:

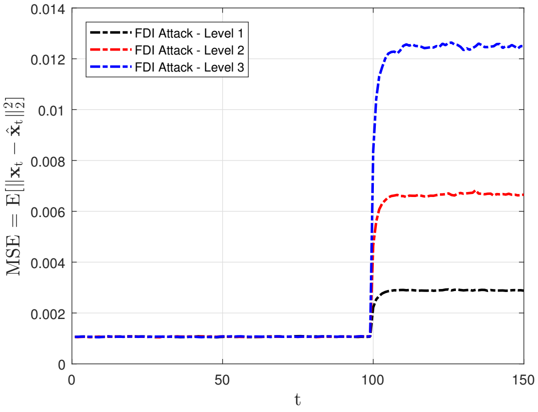


Fig. 1: Mean squared state estimation error vs. time where random FDI attacks with various magnitude levels are launched at time $\tau = 100$.

Prediction:
$$\begin{aligned} \hat{\mathbf{x}}_{t|t-1} &= \mathbf{A}\hat{\mathbf{x}}_{t-1|t-1}, \\ \mathbf{F}_{t|t-1} &= \mathbf{A}\mathbf{F}_{t-1|t-1}\mathbf{A}^{\mathrm{T}} + \sigma_v^2 \mathbf{I}_N, \end{aligned} \tag{3}$$
Measurement update:
$$\begin{aligned} \mathbf{G}_t &= \mathbf{F}_{t|t-1}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{F}_{t|t-1}\mathbf{H}^{\mathrm{T}} + \sigma_w^2 \mathbf{I}_K)^{-1}, \\ \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{G}_t(\mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}), \\ \mathbf{F}_{t|t} &= \mathbf{F}_{t|t-1} - \mathbf{G}_t\mathbf{H}\mathbf{F}_{t|t-1}, \end{aligned} \tag{4}$$
where $\mathbf{F}_{t|t-1}$ and $\mathbf{F}_{t|t}$ denote the estimates of the state covariance matrix based on the measurements up to $t-1$ and t, respectively. Moreover, $\mathbf{G}_t$ is the Kalman gain matrix at time t.

We next demonstrate the effect of cyber-attacks on the state estimation mechanism via an illustrative example. In IEEE-14 bus power system with system parameters chosen as $\mathbf{A} = \mathbf{I}_N$, $\sigma_v^2 = 10^{-4}$, and $\sigma_w^2 = 2 \times 10^{-4}$, we consider a random FDI attack with various magnitude/intensity levels and show how the mean squared state estimation error of the Kalman filter changes when FDI attacks are launched to the system. We assume that the attacks are launched at $\tau = 100$, i.e., the system is operated under normal (non-anomalous) conditions up to time 100 and under attacking conditions afterwards. Three attack magnitude levels are considered:

- Level 1: $b_{k,t} \sim \mathcal{U}[-0.04, 0.04]$, $\forall k \in \{1, \ldots, K\}$ and $\forall t \geq \tau$,
- Level 2: $b_{k,t} \sim \mathcal{U}[-0.07, 0.07]$, $\forall k \in \{1, \ldots, K\}$ and $\forall t \geq \tau$,
- Level 3: $b_{k,t} \sim \mathcal{U}[-0.1, 0.1]$, $\forall k \in \{1, \ldots, K\}$ and $\forall t \geq \tau$,

where $\mathcal{U}[\zeta_1, \zeta_2]$ denotes a uniform random variable in the range of $[\zeta_1, \zeta_2]$. The corresponding mean squared error (MSE) versus time curves are presented in Fig. 1. We observe that in case of cyber-attacks, the state estimates are deviated from the actual system states where the amount of deviation increases with the attack magnitude.

III. Problem Formulation

Before we introduce our problem formulation, we briefly explain a POMDP setting as follows. Given an agent and an environment, a discrete-time POMDP is defined by the seven-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \mathcal{G}, \gamma)$ where $\mathcal{S}$ denotes the set of (hidden) states of the environment, $\mathcal{A}$ denotes the set of actions of the agent, $\mathcal{T}$ denotes the set of conditional transition probabilities between the states, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ denotes the reward function that maps the state-action pairs to rewards, $\mathcal{O}$ denotes the set of observations of the agent, $\mathcal{G}$ denotes the set of conditional observation probabilities, and $\gamma \in [0, 1]$ denotes a discount factor that indicates how much present rewards are preferred over future rewards.

At each time t, the environment is in a particular hidden state $s_t \in \mathcal{S}$. Obtaining an observation $o_t \in \mathcal{O}$ depending on the current state of the environment with the probability $\mathcal{G}(o_t|s_t)$, the agent takes an action $a_t \in \mathcal{A}$ and receives a reward $r_t = \mathcal{R}(s_t, a_t)$ from the environment based on its action and the current state of the environment. At the same time, the environment makes a transition to the next state s_{t+1} with the probability $\mathcal{T}(s_{t+1}|s_t, a_t)$. The process is repeated until a terminal state is reached. In this process, the goal of the agent is to determine an optimal policy $\pi : \mathcal{O} \rightarrow \mathcal{A}$ that maps observations to actions and maximizes the agent's expected total discounted reward, i.e., $\mathrm{E}\big[\sum_{t=0}^{\infty} \gamma^t r_t\big]$. Equivalently, if an agent receives costs instead of rewards from the environment, then the goal is to minimize the expected total discounted cost. Considering the latter, the POMDP problem can be written as follows:

$$\min_{\pi:\, \mathcal{O} \rightarrow \mathcal{A}} \; \mathrm{E}\Big[\sum_{t=0}^{\infty} \gamma^t r_t\Big]. \tag{5}$$

Next, we explain the online attack detection problem in a POMDP setting. We assume that at an unknown time τ, a cyber-attack is launched to the system and our aim is to detect the attack as quickly as possible after it occurs, where the attacker's capabilities/strategies are completely unknown. This defines a quickest change detection problem where the aim is to minimize the average detection delay as well as the false alarm rate. This problem can, in fact, be expressed as a POMDP problem (see Fig. 2). In particular, due to the unknown attack launch time τ, there are two hidden states: *pre-attack* and *post-attack*. At each time t, after obtaining the measurement vector $\mathbf{y}_t$, two actions are available for the agent (defender): *stop* and declare an attack or *continue* to have further measurements. We assume that whenever the action *stop* is chosen, the system moves into a *terminal* state, and always stays there afterwards.

Furthermore, although the conditional observation probability for the *pre-attack* state can be inferred based on the system model under normal operating conditions, since the attacking strategies are unknown, the conditional observation probability for the *post-attack* state is assumed to be totally unknown. Moreover, due to the unknown attack launch time τ, state transition probability between the *pre-attack* and the *post-attack* states is unknown.

Since our aim is to minimize the detection delays and the false alarm rate, both the false alarm and the detection delay events should be associated with some costs. Let the relative cost of a detection delay compared to a false alarm event be $c > 0$. Then, if the true underlying state is *pre-attack* and the action *stop* is chosen, a false alarm occurs and the defender receives a cost of 1. On the other hand, if the underlying state is *post-attack* and the action *continue* is chosen, then the defender receives a cost of c due to the detection delay. For all other (hidden) state-action pairs, the cost is assumed to be zero. Also, once the action *stop* is chosen, the defender does not receive any further costs while staying in the *terminal* state. The objective of the defender is to minimize its expected total cost by properly choosing its actions. Particularly, based on its observations, the defender needs to determine the stopping time at which an attack is declared.

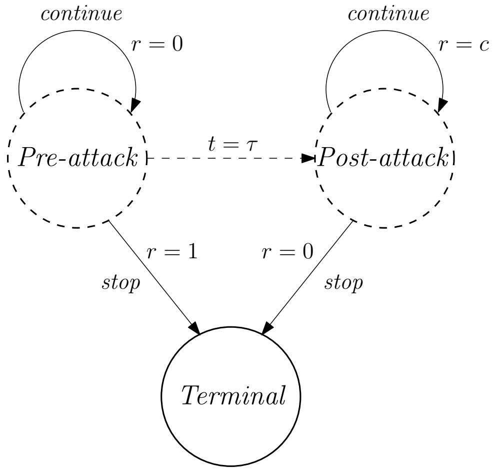


Fig. 2: State-machine diagram for the considered POMDP setting. The hidden states and the (hidden) transition between them happening at time $t = \tau$ are illustrated with the dashed circles and the dashed line, respectively. The defender receives costs (r) depending on its actions and the underlying state of the environment. Whenever the defender chooses the action *stop*, the system moves into a *terminal* state and the defender receives no further cost.

Let Γ denote the stopping time chosen by the defender. Moreover, let P_k denote the probability measure if the attack is launched at time k, i.e., $\tau = k$, and let E_k denote the corresponding expectation. Note that since the attacking strategies are unknown, P_k is assumed to be unknown. For the considered online attack detection problem, we can derive the expected total discounted cost as follows:

$$\begin{aligned}\mathrm{E}\Big[\sum_{t=0}^{\infty} \gamma^t r_t\Big] &= \mathrm{E}_\tau\Big[\mathbb{1}\{\Gamma < \tau\} + \sum_{t=\tau}^{\Gamma} c\Big] \\ &= \mathrm{E}_\tau\big[\mathbb{1}\{\Gamma < \tau\} + c\,(\Gamma - \tau)^+\big] \\ &= \mathrm{P}_\tau(\{\Gamma < \tau\}) + c\,\mathrm{E}_\tau\big[(\Gamma - \tau)^+\big], \end{aligned} \tag{6}$$

where $\gamma = 1$ is chosen since the present and future costs are equally weighted in our problem, $\{\Gamma < \tau\}$ is a false alarm event that is penalized with a cost of 1, and $\mathrm{E}_\tau\big[(\Gamma - \tau)^+\big]$ is the average detection delay where each detection delay is penalized with a cost of c and $(\cdot)^+ = \max(\cdot, 0)$.

Based on (5) and (6), the online attack detection problem can be written as follows:

$$\min_{\Gamma} \; \mathrm{P}_\tau(\{\Gamma < \tau\}) + c\,\mathrm{E}_\tau\big[(\Gamma - \tau)^+\big]. \tag{7}$$

Since c corresponds to the relative cost between the false alarm and the detection delay events, by varying c and solving the corresponding problem in (7), a tradeoff curve between average detection delay and false alarm rate can be obtained. Moreover, $c < 1$ can be chosen to prevent frequent false alarms.

Since the exact POMDP model is unknown due to unknown attack launch time τ and the unknown attacking strategies

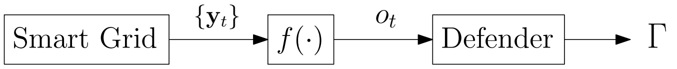


Fig. 3: A graphical description of the online attack detection problem in the smart grid. The measurements $\{\mathbf{y}_t\}$ are collected through smart meters and processed to obtain $o_t = f(\{\mathbf{y}_t\})$. The defender observes $f(\{\mathbf{y}_t\})$ at each time t and decides on the attack declaration time Γ.

and since the RL algorithms are known to be effective over uncertain environments, we follow a model-free RL approach to obtain a solution to (7). Then, a direct mapping from observations to the actions, i.e., the stopping time Γ, needs to be learned. Note that the optimal action is *continue* if the underlying state is *pre-attack* and *stop* if the underlying state is *post-attack*. Then, to determine the optimal actions, the underlying state needs to be inferred using observations and the observation signal should be well informative to reduce the uncertainty about the underlying state. As described in Sec. II, the defender observes the measurements $\mathbf{y}_t$ at each time t. The simplest approach can be forming the observation space directly with the measurement vector $\mathbf{y}_t$ but we would like to process the measurements and form the observation space with a signal related to the deviation of system from its normal operation.

Furthermore, it is, in general, possible to obtain identical observations in the *pre-attack* and the *post-attack* states. This is called perceptual aliasing and prevent us to make a good inference about the underlying state by only looking at the observation at a single time. We further note that in our problem, deciding on an attack solely based on a single observation corresponds to an outlier detection scheme for which more practical detectors are available not requiring a learning phase, see e.g., [25], [26]. However, we are particularly interested in detecting sudden and persistent attacks/anomalies that more likely happen due to an unfriendly intervention to the system rather than random disturbances due to high-level system noise realizations.

Since different states require different optimal actions, the ambiguity on the underlying state should be further reduced with additional information derived from the history of observations. In fact, there may be cases where the entire history of observations is needed to determine the optimal solution in a POMDP problem [38]. However, due to computational limitations, only a finite memory can be used in practice and an approximately optimal solution can be obtained. A simple approach is to use a finite-size sliding window of observations as a memory and map the most recent history window to an action, as described in [22]. This approach is particularly suitable for our problem as well since we assume persistent attacks/anomalies that happen at an unknown point of time and continue thereafter. That is, only the observations obtained after an attack are significant from the attack detection perspective.

Let the function that processes a finite history of measurements and produces the observation signal be denoted with $f(\cdot)$ so that the observation signal at time t is $o_t = f(\{\mathbf{y}_t\})$. Then, at each time, the defender observes $f(\{\mathbf{y}_t\})$ and decides on the stopping time Γ, as illustrated in Fig. 3. The aim of the defender is to obtain a solution to (7) by using an RL algorithm, as detailed in the subsequent section.

IV. Solution Approach

Firstly, we explain our methodology to obtain the observation signal $o_t = f(\{\mathbf{y}_t\})$. Note that the pdf of meter measurements in the *pre-attack* state can be inferred using the baseline measurement model in (2) and the state estimates provided by the Kalman filter. In particular, the pdf of the measurements under normal operating conditions can be estimated as follows:

$$\mathbf{y}_t \sim \mathcal{N}(\mathbf{H}\hat{\mathbf{x}}_{t|t}, \sigma_w^2\,\mathbf{I}_K).$$

The likelihood of measurements based on the baseline density estimate, denoted with $L(\mathbf{y}_t)$, can then be computed as follows:

$$\begin{aligned} L(\mathbf{y}_t) &= (2\pi\sigma_w^2)^{-\frac{K}{2}} \exp\Big(\frac{-1}{2\sigma_w^2}(\mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t})^{\mathrm{T}}(\mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t})\Big) \\ &= (2\pi\sigma_w^2)^{-\frac{K}{2}} \exp\Big(\frac{-1}{2\sigma_w^2}\eta_t\Big), \end{aligned}$$

where

$$\eta_t \triangleq (\mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t})^{\mathrm{T}}(\mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t}) \tag{8}$$

is the estimate of the negative log-scaled likelihood.

In case the system is operated under normal conditions, the likelihood $L(\mathbf{y}_t)$ is expected to be high. Equivalently, small (close to zero) values of η_t may indicate the normal system operation. On the other hand, in case of an attack/anomaly, the system deviates from normal operating conditions and hence the likelihood $L(\mathbf{y}_t)$ is expected to decrease in such cases. Then, persistent high values of η_t over a time period may indicate an attack/anomaly. Hence, η_t may help to reduce the uncertainty about the underlying state to some extent.

However, since η_t can take any nonnegative value, the observation space is continuous and hence learning a mapping from each possible observation to an action is computationally infeasible. To reduce the computational complexity in such continuous spaces, we can quantize the observations. We then partition the observation space into I mutually exclusive and disjoint intervals using the quantization thresholds $\beta_0 = 0 < \beta_1 < \cdots < \beta_{I-1} < \beta_I = \infty$ so that if $\beta_{i-1} \leq \eta_t < \beta_i, i \in 1, \ldots, I$, the observation at time t is represented with θ_i. Then, possible observations at any given time are $\theta_1, \ldots, \theta_I$. Since θ_i's are representations of the quantization levels, each θ_i needs to be assigned to a different value.

Furthermore, as explained before, although η_t may be useful to infer the underlying state at time t, it is possible to obtain identical observations in the *pre-attack* and *post-attack* states. For this reason, we propose to use a finite history of observations. Let the size of the sliding observation window be M so that there are I^M possible observation windows and the sliding window at time t consists of the quantized versions of $\{\eta_j : t - M + 1 \leq j \leq t\}$. Henceforth, by an observation o, we refer to an observation window so that the observation space $\mathcal{O}$ consists of all possible observation windows. For instance, if $I = M = 2$, then $\mathcal{O} = \{[\theta_1, \theta_1], [\theta_1, \theta_2], [\theta_2, \theta_1], [\theta_2, \theta_2]\}$.

For each possible observation-action pair (o, a), we propose to learn a $Q(o, a)$ value, i.e., the expected future cost, using an RL algorithm where all $Q(o, a)$ values are stored in a Q-table of size $I^M \times 2$. After learning the Q-table, the policy of the defender will be choosing the action a with the minimum $Q(o, a)$ for each observation o. In general, increasing I and M improves the learning performance but at the same time results

Algorithm 1 Learning Phase – SARSA Algorithm

```
1:  Initialize $Q(o,a)$ arbitrarily, $\forall o \in \mathcal{O}$ and $\forall a \in \mathcal{A}$.
2:  for $e = 1 : E$ do
3:      $t \leftarrow 0$
4:      $s \leftarrow$ pre-attack
5:      Choose an initial $o$ based on the pre-attack state and choose the initial $a =$ continue.
6:      while $s \neq$ terminal and $t < T$ do
7:          $t \leftarrow t + 1$
8:          if $a =$ stop then
9:              $s \leftarrow$ terminal
10:             $r \leftarrow \mathbb{1}\{t < \tau\}$
11:             $Q(o,a) \leftarrow Q(o,a) + \alpha\,(r - Q(o,a))$
12:         else if $a =$ continue then
13:             if $t >= \tau$ then
14:                 $r \leftarrow c$
15:                 $s \leftarrow$ post-attack
16:             else
17:                 $r \leftarrow 0$
18:             end if
19:             Collect the measurements $\mathbf{y}_t$.
20:             Employ the Kalman filter using (3) and (4).
21:             Compute $\eta_t$ using (8) and quantize it to obtain $\theta_i$ if $\beta_{i-1} \leq \eta_t < \beta_i, i \in 1, \ldots, I$.
22:             Update the sliding observation window $o$ with the most recent entry $\theta_i$ and obtain $o'$.
23:             Choose action $a'$ from $o'$ using the $\epsilon$-greedy policy based on the $Q$-table (that is being learned).
24:             $Q(o,a) \leftarrow Q(o,a) + \alpha\,(r + Q(o',a') - Q(o,a))$
25:             $o \leftarrow o'$, $a \leftarrow a'$
26:         end if
27:     end while
28: end for
29: Output: $Q$-table, i.e., $Q(o,a)$, $\forall o \in \mathcal{O}$ and $\forall a \in \mathcal{A}$.
```

Algorithm 2 Online Attack Detection

```
Input: $Q$-table learned in Algorithm 1.
Choose an initial $o$ based on the pre-attack state and choose the initial $a =$ continue.
$t \leftarrow 0$
while $a \neq$ stop do
    $t \leftarrow t + 1$
    Collect the measurements $\mathbf{y}_t$.
    Determine the new $o$ as in the lines 20–22 of Algorithm 1.
    $a \leftarrow \arg\min_a Q(o,a)$.
end while
Declare an attack and terminate the procedure.
```

in a larger Q table, that would require to increase the number of training episodes and hence the computational complexity of the learning phase. Hence, I and M should be chosen considering the expected tradeoff between performance and computational complexity.

The considered RL-based detection scheme consists of learning and online detection phases. In the literature, SARSA, a model-free RL control algorithm [39], was numerically shown to perform well over the model-free POMDP settings [24]. Hence, in the learning phase, the defender is trained with many episodes of experience using the SARSA algorithm and a Q-table is learned by the defender. For training, a simulation environment is created and during the training procedure, at each time, the defender takes an action based on its observation and receives a cost in return of its action from the simulation environment, as illustrated in Fig. 4. Based on this experience, the defender updates and learns a Q-table. Then, in the online detection phase, based on the observations, the action with the lowest expected future cost (Q value) is chosen at each time using the previously learned Q-table. The online detection phase continues until the action *stop* is chosen by the defender. Whenever *stop* is chosen, an attack is declared and the process is terminated.

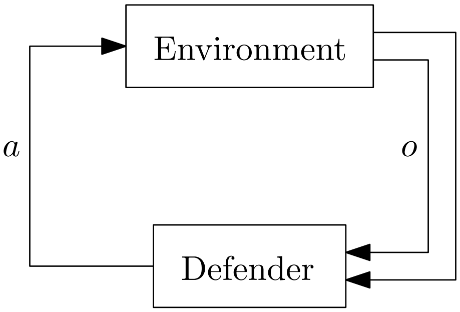


Fig. 4: An illustration of the interaction between defender and the simulation environment during the learning procedure. The environment provides an observation o based on its internal state s, the agent chooses an action a based on its observation and receives a cost r from the environment in return of its action. Based on this experience, the defender updates $Q(o,a)$. This process is repeated many times during the learning procedure.

Note that after declaring an attack, whenever the system is recovered and returned back to the normal operating conditions, the online detection phase can be restarted. That is, once a defender is trained, no further training is needed. We summarize the learning and the online detection stages in Algorithms 1 and 2, respectively. In Algorithm 1, E denotes the number of learning episodes, T denotes the maximum length of a learning episode, α is the learning rate, and ϵ is the exploration rate, where the ϵ-greedy policy chooses the action with the minimum Q value with probability $1 - \epsilon$ and the other action (for exploration purposes during the learning process) with probability ϵ.

Since RL is an iterative procedure, same actions are repeated at each iteration (learning episode). The time complexity of an RL algorithm can then be considered as the time complexity of a single iteration [40]. As the SARSA algorithm performs one update on the Q-table at a time and the maximum time limit for a learning episode is T, the time complexity of Algorithm 1 is $O(T)$. Moreover, the overall complexity of the learning procedure is $O(TE)$, as E is the number of learning episodes. Notice that the time complexity does not depend on the size of the action and observation spaces. On the other hand, as I and/or M increase, a larger Q-table needs to be learned, that requires to increase E for a better learning. Furthermore, the space complexity (memory cost) of Algorithm 1 is $M + 2\,I^M$ due to the sliding observation window of size M and the Q-table of size $I^M \times 2$. Note that the space complexity is fixed over time. During the learning procedure, based on the smart grid model and some attack models (that are used to obtain low-magnitude attacks that correspond to small deviations from the normal system operation), we can obtain the measurement data online and the defender is trained with the observed data stream. Hence, the learning phase does not require storage of large amount of training data as only a sliding observation window of size M needs to be stored at each time.

In Algorithm 2, at each time, the observation o is determined and using the Q-table (learned in Algorithm 1), the corresponding action a with the minimum cost is chosen. Hence, the complexity at a time is $O(1)$. This process is repeated until the action *stop* is chosen at the stopping time Γ. Furthermore, similarly to Algorithm 1, the space complexity of Algorithm 2 is $M + 2\,I^M$.

Remark 1: Since our solution approach is model-free, i.e., it is not particularly designed for specific types of attacks, the proposed detector does not distinguish between an attack and other types of persistent anomalies such as network topology faults. In fact, the proposed algorithm can detect any attack/anomaly as long as effect of such attack/anomaly on the system is at a distinguishable level, i.e., the estimated system states are at least slightly deviated from the actual system states. On the other hand, since we train the agent (defender) with low-magnitude attacks with some known attack types (to create the effect of small deviations from actual system operation in the *post-attack* state) and we test the proposed detector against various cyber-attacks in the numerical section (see Sec. V), we lay the main emphasis on online attack detection in this study. In general, the proposed detector can be considered as an online anomaly detection algorithm.

Remark 2: The proposed solution scheme can be applied in a distributed smart grid system, where the learning and detection tasks are still performed at a single center but the meter measurements are obtained in a distributed manner. We briefly explain this setup as follows:

- In the wide-area monitoring model of smart grids, there are several local control centers and a global control center. Each local center collects and processes measurements of a set of smart meters in its neighborhood, and communicates with the global center and with the neighboring local centers.
- The system state is estimated in a distributed manner, e.g., using the distributed Kalman filter designed for wide-area smart grids in [8].
- Let $\mathbf{h}_k^T \in \mathbb{R}^N$ be the kth row of the measurement matrix, i.e., $\mathbf{H}^T = [\mathbf{h}_1, \dots, \mathbf{h}_K]$. Then, estimate of the negative log-scaled likelihood, η_t, can be written as follows (see (8)):
$$\eta_t = \sum_{k=1}^{K} (y_{k,t} - \mathbf{h}_k^T \hat{\mathbf{x}}_{t|t})^2. \tag{9}$$
By employing the distributed Kalman filter, the local centers can estimate the system state at each time t. Then, they can compute the term $(y_{k,t} - \mathbf{h}_k^T \hat{\mathbf{x}}_{t|t})^2$ for the meters in their neighborhood. Let the number of local centers be R and the set of meters in the neighborhood of the rth local center be denoted with $\mathcal{S}_r$. Then, η_t in (9) can be rewritten as follows:
$$\eta_t = \sum_{r=1}^{R} \underbrace{\sum_{k \in \mathcal{S}_r} (y_{k,t} - \mathbf{h}_k^T \hat{\mathbf{x}}_{t|t})^2}_{\eta_{t,r}}$$
$$= \sum_{r=1}^{R} \eta_{t,r}.$$
- In the distributed implementation, each local center can compute $\eta_{t,r}$ and report it to the global center, which then sums $\{\eta_{t,r}, r = 1, 2, \dots, R\}$ and obtain η_t.
- The learning and detection tasks (Algorithms 1 and 2) are performed at the global center in the same way as explained above.

V. Simulation Results

A. Simulation Setup and Parameters

Simulations are performed on an IEEE-14 bus power system that consists of $N + 1 = 14$ buses and $K = 23$ smart meters. The initial state variables (phase angles) are determined using the DC optimal power flow algorithm for case-14 in MATPOWER [41]. The system matrix $\mathbf{A}$ is chosen to be an identity matrix and the measurement matrix $\mathbf{H}$ is determined based on the IEEE-14 power system. The noise variances for the normal system operation are chosen as $\sigma_v^2 = 10^{-4}$ and $\sigma_w^2 = 2 \times 10^{-4}$. Simulations are performed on MATLAB using a PC with Intel Xeon CPU E5-1607 v3 @ 3.10 GHz processor and 16 GB RAM.

For the proposed RL-based online attack detection scheme, the number of quantization levels is chosen as $I = 4$ and the quantization thresholds are chosen as $\beta_1 = 0.95 \times 10^{-2}$, $\beta_2 = 1.05 \times 10^{-2}$, and $\beta_3 = 1.15 \times 10^{-2}$ via an offline simulation by monitoring $\{\eta_t\}$ during the normal system operation. Further, $M = 4$ is chosen, i.e., sliding observation window consists of 4 entries. Moreover, the learning parameters are chosen as $\alpha = 0.1$ and $\epsilon = 0.1$, and the episode length is chosen to be $T = 200$. In the learning phase, the defender is firstly trained over 4×10^5 episodes where the attack launch time is $\tau = 100$ and then trained further over 4×10^5 episodes where $\tau = 1$ to ensure that the defender sufficiently explores the observation space under normal operating conditions as well as the attacking conditions. More specifically, since a learning episode is terminated whenever the action *stop* is chosen and observations under an attack become available to the defender only for $t \geq \tau$, we choose $\tau = 1$ in the half of the learning episodes to make sure that the defender is sufficiently trained under the post-attack regime.

To illustrate the tradeoff between the average detection delay and the false alarm probability, the proposed algorithm is trained for both $c = 0.02$ and $c = 0.2$. Moreover, to obtain a detector that is robust and effective against small deviations of measurements from the normal system operation, the defender needs to be trained with very low-magnitude attacks that correspond to slight deviations from the baseline. For this purpose, some known attack types with low magnitudes are used. In particular, in one half of the learning episodes, random FDI attacks are used with attack magnitudes being realizations of the uniform random variable $\pm\mathcal{U}[0.02, 0.06]$, i.e., $b_{k,t} \sim \mathcal{U}[0.02, 0.06]$ is the injected false datum to the kth meter at time $t \geq \tau$, $\forall k \in \{1, \dots, K\}$. In the other half of the learning episodes, random hybrid FDI/jamming attacks are used where $b_{k,t} \sim \mathcal{U}[0.02, 0.06]$, $u_{k,t} \sim \mathcal{N}(0, \sigma_{k,t})$, and $\sigma_{k,t} \sim \mathcal{U}[2 \times 10^{-4}, 4 \times 10^{-4}]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \geq \tau$. The total training time costs are recorded approximately as 5018 seconds and 5106 seconds for $c = 0.2$ and $c = 0.02$, respectively.

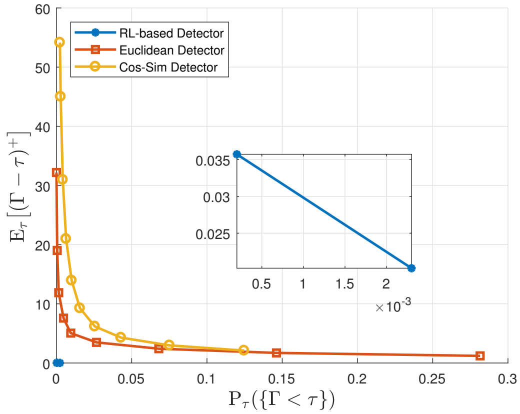


Fig. 5: Average detection delay vs. probability of false alarm curves for the proposed algorithm and the benchmark tests in case of a random FDI attack.

B. Performance Evaluation

In this section, performance of the proposed RL-based attack detection scheme is evaluated and compared with some existing detectors in the literature. Firstly, we report the average false alarm period, $\mathrm{E}_\infty[\Gamma]$, of the proposed detection scheme, i.e., the first time on the average the proposed detector gives an alarm although no attack/anomaly happens at all ($\tau = \infty$). The average false alarm periods are obtained approximately as $\mathrm{E}_\infty[\Gamma] = 9.4696 \times 10^5$ for $c = 0.2$ and $\mathrm{E}_\infty[\Gamma] = 7.9210 \times 10^6$ for $c = 0.02$. As expected, false alarm rate of the proposed detector reduces as the relative cost of the false alarm event, $1/c$, increases.

Based on the optimization problem in (7), our performance metrics are the probability of false alarm, i.e., $\mathrm{P}_\tau(\{\Gamma < \tau\})$, and the average detection delay, i.e., $\mathrm{E}_\tau\left[(\Gamma-\tau)^+\right]$. Notice that both performance metrics depend on the unknown attack launch time τ. Hence, in general, the performance metrics need to be computed for each possible τ. For a representative performance illustration, we choose τ as a geometric random variable with parameter ρ such that $P(\tau = k) = \rho(1-\rho)^{k-1}, k = 1, 2, 3, \dots$ where $\rho \sim \mathcal{U}[10^{-4}, 10^{-3}]$ is a uniform random variable.

With Monte Carlo simulations over 10000 trials, we compute the probability of false alarm and the average detection delay of the proposed detector, the Euclidean detector [25], and the cosine-similarity metric based detector [26]. To obtain the performance curves, we vary the thresholds of the benchmark tests and vary c for the proposed algorithm. To evaluate the proposed algorithm, we use Algorithm 2 that makes use of the Q-tables learned in Algorithm 1 for $c = 0.02$ and $c = 0.2$. Furthermore, we report the precision, recall, and F-score for all simulation cases. As the computation of these measures requires computing the number of detected and missed trials, we define an upper bound on the detection delay (that corresponds to the maximum acceptable detection delay) such that if the attack is detected within this bound we assume the attack is detected, otherwise missed. As an example, we choose this bound as 10 time units. Then, we compute the precision, recall, and F-score out of 10000 trials as follows:

$$\text{Precision} = \frac{\text{\# trials } (\tau \le \Gamma \le \tau+10)}{\text{\# trials } (\tau \le \Gamma \le \tau+10) + \text{\# trials } (\Gamma < \tau)},$$

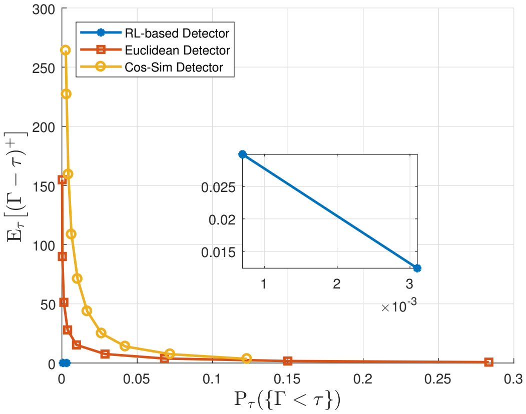


Fig. 6: Performance curves for the proposed algorithm and the benchmark tests in case of a structured FDI attack.

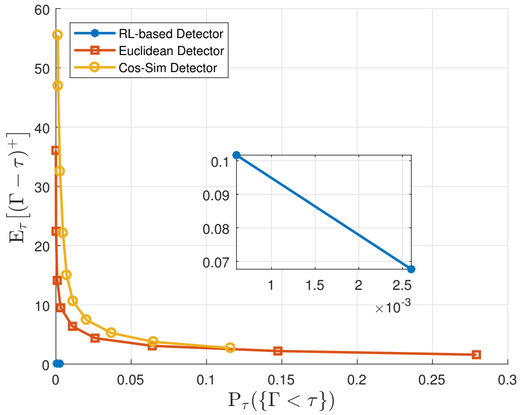


Fig. 7: Performance curves for the proposed algorithm and the benchmark tests in case of a jamming attack with AWGN.

$$\text{Recall} = \frac{\text{\# trials } (\tau \le \Gamma \le \tau+10)}{\text{\# trials } (\tau \le \Gamma \le \tau+10) + \text{\# trials } (\Gamma > \tau+10)},$$

and

$$\text{F-score} = 2\,\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

where "# trials" means "the number of trials with".

We evaluate the proposed and the benchmark detectors under the following attack scenarios:

1) Firstly, we evaluate the detectors against a random FDI attack where $b_{k,t} \sim \mathcal{U}[-0.07, 0.07]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \ge \tau$. The corresponding tradeoff curves are presented in Fig. 5.
2) We then evaluate the detectors against a structured FDI attack [5], where the injected data $\mathbf{b}_t$ lies on the column space of the measurement matrix $\mathbf{H}$. We choose $\mathbf{b}_t = \mathbf{H}\mathbf{g}_t$ where $\mathbf{g}_t \triangleq [g_{1,t}, \dots, g_{N,t}]^{\mathrm{T}}$ and $g_{n,t} \sim \mathcal{U}[0.08, 0.12]$, $\forall n \in \{1, \dots, N\}$ and $\forall t \ge \tau$. The corresponding performance curves are illustrated in Fig. 6.
3) Then, we evaluate the detectors in case of a jamming attack with zero-mean AWGN where $u_{k,t} \sim \mathcal{N}(0, \sigma_{k,t})$ and $\sigma_{k,t} \sim \mathcal{U}[10^{-3}, 2 \times 10^{-3}]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \ge \tau$. The corresponding tradeoff curves are presented in Fig. 7.

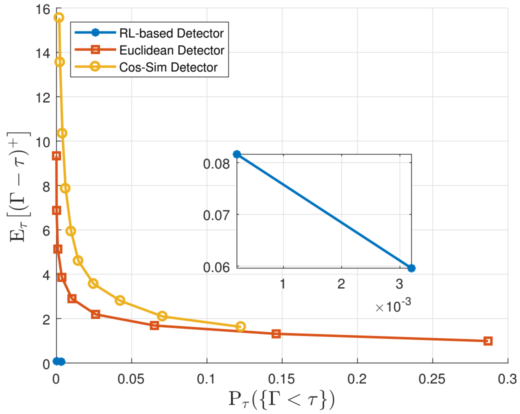


Fig. 8: Performance curves for the proposed algorithm and the benchmark tests in case of a jamming attack with jamming noise correlated over the space.

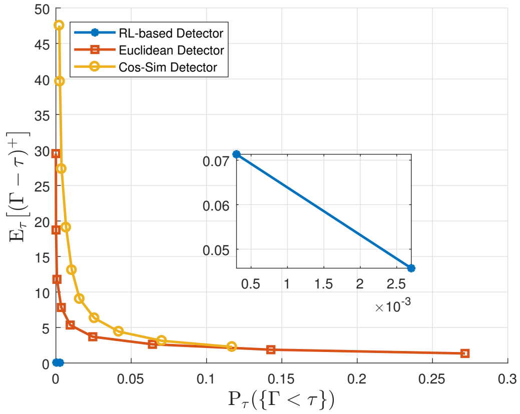


Fig. 9: Performance curves for the proposed algorithm and the benchmark tests in case of a hybrid FDI/jamming attack.

4) Next, we evaluate the detectors in case of a jamming attack with jamming noise correlated over the meters where $\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{U}_t)$, $\mathbf{U}_t = \mathbf{\Sigma}_t \mathbf{\Sigma}_t^{\mathrm{T}}$, and $\mathbf{\Sigma}_t$ is a random Gaussian matrix with its entry at the ith row and the jth column is $\mathbf{\Sigma}_{t,i,j} \sim \mathcal{N}(0, 8 \times 10^{-5})$. The corresponding performance curves are given in Fig. 8.
5) Moreover, we evaluate the detectors under a hybrid FDI/jamming attack where $b_{k,t} \sim \mathcal{U}[-0.05, 0.05]$, $u_{k,t} \sim \mathcal{N}(0, \sigma_{k,t})$, and $\sigma_{k,t} \sim \mathcal{U}[5 \times 10^{-4}, 10^{-3}]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \geq \tau$. The corresponding tradeoff curves are presented in Fig. 9.
6) Then, we evaluate the detectors in case of a random DoS attack where the measurement of each smart meter become unavailable to the system controller at each time with probability 0.2. That is, for each meter k, $d_{k,t}$ is 0 with probability 0.2 and 1 with probability 0.8 at each time $t \geq \tau$. The performance curves against the DoS attack are presented in Fig. 10.
7) Further, we consider a network topology attack where the lines between the buses 9-10 and 12-13 break down. The measurement matrix $\bar{\mathbf{H}}_t$ for $t \geq \tau$ is determined accordingly. The corresponding tradeoff curves are given in Fig. 11.
8) Finally, we consider a mixed topology and hybrid FDI/jamming attack, where the lines between buses 9-10 and 12-13 break down for $t \geq \tau$ and further, we have $b_{k,t} \sim \mathcal{U}[-0.05, 0.05]$, $u_{k,t} \sim \mathcal{N}(0, \sigma_{k,t})$, and $\sigma_{k,t} \sim \mathcal{U}[5 \times 10^{-4}, 10^{-3}]$, $\forall k \in \{1, \dots, K\}$ and $\forall t \geq \tau$. The corresponding performance curves are presented in Fig. 12.

Table II and Table III summarize the precision, recall, and F-score for the proposed RL-based detector for $c = 0.2$ and $c = 0.02$, respectively against all the considered simulation cases above. Moreover, for the random FDI attack case, Fig. 13 illustrates the precision versus recall curves for the proposed and benchmark detectors. Since we obtain similar results for the other attack cases, we report the results for the random FDI attack case as a representative.

For almost all cases, we observe that the proposed RL-based detection scheme significantly outperforms the benchmark tests. This is because through the training process, the defender learns to differentiate the instantaneous high-level system noise from persistent attacks launched to the system. Then, the trained defender is able to significantly reduce its false alarm rate. Moreover, since the defender is trained with low attack magnitudes, it becomes sensitive to detect small deviations of the system from its normal operation. On the other hand, the benchmark tests are essentially outlier detection schemes making sample-by-sample decisions and hence they are unable to distinguish high-level noise realizations from real attacks that makes such schemes more vulnerable to false alarms. Furthermore, in case of DoS attacks, since the meter measurements become partially unavailable so that the system greatly deviates from its normal operation, all detectors are able to detect the DoS attacks with almost zero average detection delays (see Fig. 10).

Finally, we evaluate the effect of the window size, M, on the performance of the RL-based detector (trained for $c = 0.2$) against random FDI attacks with varying magnitudes. Table IV summarizes the results for $M = 1$, $M = 2$, $M = 4$, and $M = 6$ where $b_{k,t} \sim \mathcal{U}[-\varphi, \varphi]$, $\forall k \in \{1, \dots, K\}$, $\forall t \geq \tau$ and φ takes the values of 0.03, 0.04, and 0.05. We observe that in terms of false alarm rate, $M = 1$ case is significantly worse than the other window size cases. This is because $M = 1$ case corresponds to an outlier detection scheme that is more vulnerable to false alarms. Moreover, in terms of detection delays, as we compare the cases $M = 2$, $M = 4$, and $M = 6$, advantage of having a larger window size appears as the attack magnitudes get smaller, i.e., as the detection becomes more difficult. This is because, as M increases, the detector can better evaluate small deviations from the baseline since a larger group of observations may collectively indicate an anomaly. On the other hand, as the attack magnitudes increase, the detector performs more similarly for the cases $M = 2$, $M = 4$, and $M = 6$. We note that sensitivity and the performance of the detector can be further improved by increasing the number of quantization levels I.

VI. Concluding Remarks

In this paper, an online cyber-attack detection problem is formulated as a POMDP problem and a solution based on the model-free RL for POMDPs is proposed. The numerical studies illustrate the advantages of the proposed detection scheme in fast and reliable detection of cyber-attacks targeting the smart grid. The results also demonstrate the high potential of RL algorithms in solving complex cyber-security problems.

Measure	FDI	Jamming	Corr. Jamm.	Hybrid	DoS	Structured FDI	Topology	Mixed
Precision	0.9977	0.9974	0.9968	0.9973	0.9977	0.9968	0.9972	0.9973
Recall	1	1	1	1	1	0.9756	0.9808	1
F-score	0.9988	0.9987	0.9984	0.9986	0.9988	0.9861	0.9890	0.9986

TABLE II: Precision, recall, and F-score for the proposed detector ($c = 0.2$) in detection of various cyber-attacks.

Measure	FDI	Jamming	Corr. Jamm.	Hybrid	DoS	Structured FDI	Topology	Mixed
Precision	0.9998	0.9994	0.9998	0.9997	0.9995	0.9993	0.9999	0.9995
Recall	1	1	1	1	1	0.9449	0.9785	1
F-score	0.9999	0.9997	0.9999	0.9998	0.9997	0.9713	0.9891	0.9997

TABLE III: Precision, recall, and F-score for the proposed detector ($c = 0.02$) in detection of various cyber-attacks.

Window Size	$\varphi = 0.03$	$\varphi = 0.04$	$\varphi = 0.05$
$M = 1$	0.0187/ 8.4208/ 0.9739/ 0.7119/ 0.8226	0.0187/ 1.2219/ 0.9813/ 0.9995/ 0.9903	0.0187/ 0.2606/ 0.9813/ 1/ 0.9906
$M = 2$	0.0021/ 15.9532/ 0.9957/ 0.4872/ 0.6543	0.0021/ 1.9046/ 0.9979/ 0.9923/ 0.9951	0.0021/ 0.4049/ 0.9979/ 1/ 0.9989
$M = 4$	0.0021/ 14.8548/ 0.9959/ 0.5077/ 0.6725	0.0021/ 1.8437/ 0.9979/ 0.9943/ 0.9961	0.0021/ 0.4016/ 0.9979/ 1/ 0.9989
$M = 6$	0.0021/ 14.2506/ 0.9960/ 0.5210/ 0.6841	0.0021/ 1.8207/ 0.9979/ 0.9955/ 0.9967	0.0021/ 0.4016/ 0.9979/ 1/ 0.9989

TABLE IV: Effect of the window size M on the performance of the RL-based detector for $c = 0.2$ under random FDI attacks with different magnitude levels. Each entry in the table shows the results based on the following format: $\mathrm{P}_\tau(\{\Gamma < \tau\})$/ $\mathrm{E}_\tau\left[(\Gamma - \tau)^+\right]$/ Precision/ Recall/ F-score.

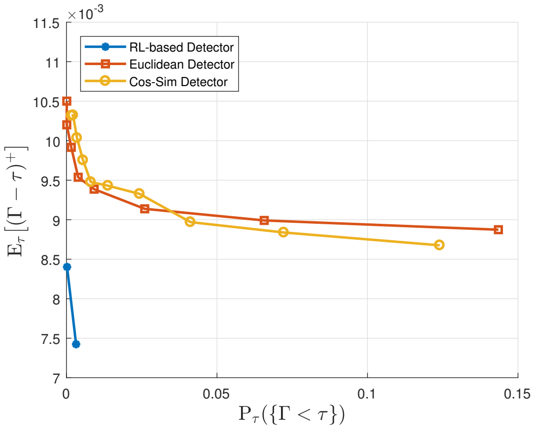


Fig. 10: Performance curves for the proposed algorithm and the benchmark tests in case of a DoS attack.

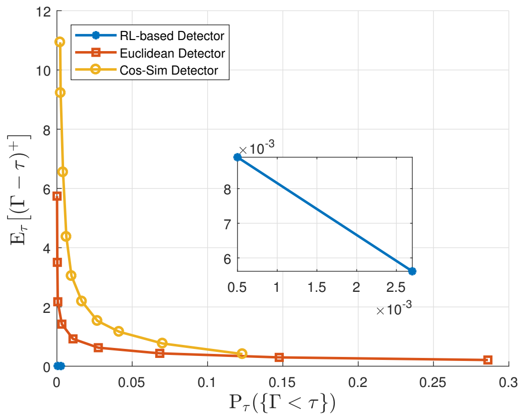


Fig. 12: Performance curves for the proposed algorithm and the benchmark tests in case of a mixed network topology and hybrid FDI/jamming attack.

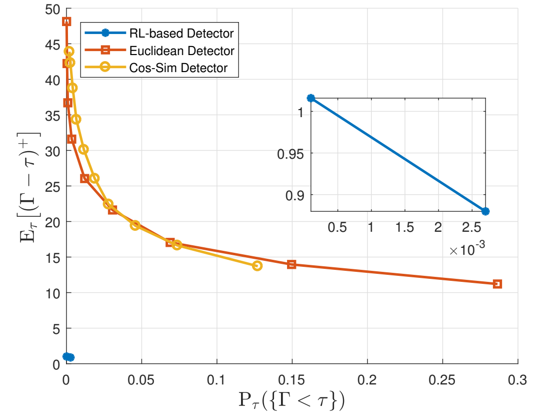


Fig. 11: Performance curves for the proposed algorithm and the benchmark tests in case of a network topology attack.

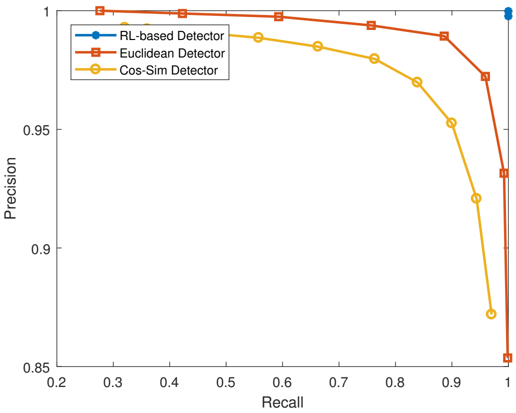


Fig. 13: Precision vs. recall for the proposed and the benchmark detectors against a random FDI attack.

In fact, the algorithm proposed in this paper can be further improved using more advanced methods. Particularly, the following directions can be considered as future works:

- compared to the finite-size sliding window approach, more sophisticated memory techniques can be developed,
- compared to discretizing the continuous observation space and using a tabular approach to compute the Q values, linear/nonlinear function approximation techniques, e.g., neural networks, can be used to compute the Q values,
- and deep RL algorithms can be useful to improve the performance.

We further note that the proposed online detection method is widely applicable to any quickest change detection problem where the pre-change model can be derived with some accuracy but the post-change model is unknown. This is, in fact, commonly encountered in many practical applications where the normal system operation can be modeled sufficiently accurately and the objective is the online detection of anomalies/attacks that are difficult to model in general. Moreover, depending on specific applications, if real post-change, e.g., attack/anomaly, data can be obtained, the real data can be further enhanced with simulated data and the training can be performed accordingly, that would potentially improve the detection performance.

Finally, in this study, we have considered a single-agent RL setting with the aim of optimizing the policy of the defender, where the attacking strategies, such as attack types, magnitudes, set of attacked meters, etc., do not alter the best policy of the defender. That is, once an attack is launched, the defender's best policy is *stop* and declare an attack. Moreover, in the current setup, the attacker does not learn. As a future work, the current setup can be extended to a multi-agent RL setting where there is an underlying partially observable stochastic game with multiple states in which the payoffs and state transitions depend on joint actions taken by the attacker and the defender. In this setting, both agents aim to learn and adapt to the environment and also to each other's policies in order to maximize their payoffs.

References

[1] G. Liang, J. Zhao, F. Luo, S. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2016.

[2] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344–1371, 2013.

[3] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys & Tutorials*, 2012.

[4] L. Xie, Y. Mo, and B. Sinopoli, "False data injection attacks in electricity markets," in *2010 First IEEE International Conference on Smart Grid Communications*, Oct 2010, pp. 226–231.

[5] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 21–32.

[6] R. B. Bobba, K. M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. J. Overbye, "Detecting false data injection attacks on dc state estimation," in *Preprints of the First Workshop on Secure Control Systems, CPSWEEK 2010*, 2010.

[7] S. Li, Y. Yilmaz, and X. Wang, "Quickest detection of false data injection attack in wide-area smart grids," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2725–2735, Nov 2015.

[8] M. N. Kurt, Y. Yilmaz, and X. Wang, "Distributed quickest detection of cyber-attacks in smart grid," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2015–2030, Aug 2018.

[9] ——, "Real-time detection of hybrid and stealthy cyber-attacks in smart grid," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 498–513, Feb 2019.

[10] S. Asri and B. Pranggono, "Impact of distributed denial-of-service attack on advanced metering infrastructure," *Wireless Personal Communications*, vol. 83, no. 3, pp. 2211–2223, 2015.

[11] Y. Zhang, L. Wang, W. Sun, R. Green, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *Smart Grid, IEEE Transactions on*, vol. 2, no. 4, pp. 796–808, 2011.

[12] H. V. Poor and O. Hadjiliadis, *Quickest Detection*. Cambridge University Press, 2008.

[13] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[14] V. V. Veeravalli and T. Banerjee, "Chapter 6 - Quickest Change Detection," in *Academic Press Library in Signal Processing: Volume 3 Array and Statistical Signal Processing*, ser. Academic Press Library in Signal Processing, R. C. Abdelhak M. Zoubir, Mats Viberg and S. Theodoridis, Eds. Elsevier, 2014, vol. 3, pp. 209 – 255.

[15] A. S. Polunchenko and A. G. Tartakovsky, "State-of-the-art in sequential change-point detection," *Methodology and Computing in Applied Probability*, vol. 14, no. 3, pp. 649–684, Sep 2012.

[16] G. V. Moustakides, "Optimal stopping times for detecting changes in distributions," *Ann. Statist.*, vol. 14, no. 4, pp. 1379–1387, 1986.

[17] G. Lorden, "Procedures for reacting to a change in distribution," *Ann. Math. Statist.*, vol. 42, no. 6, pp. 1897–1908, 1971.

[18] S. Ross, J. Pineau, B. Chaib-draa, and P. Kreitmann, "A bayesian approach for learning and planning in partially observable markov decision processes," *J. Mach. Learn. Res.*, vol. 12, pp. 1729–1770, Jul. 2011.

[19] F. Doshi-Velez, D. Wingate, N. Roy, and J. Tenenbaum, "Nonparametric bayesian policy priors for reinforcement learning," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'10. USA: Curran Associates Inc., 2010, pp. 532–540.

[20] T. Jaakkola, S. P. Singh, and M. I. Jordan, "Reinforcement learning algorithm for partially observable markov decision problems," in *Proceedings of the 7th International Conference on Neural Information Processing Systems*, ser. NIPS'94. Cambridge, MA, USA: MIT Press, 1994, pp. 345–352.

[21] T. J. Perkins, "Reinforcement learning for pomdps based on action values and stochastic optimization," in *Eighteenth National Conference on Artificial Intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 2002, pp. 199–204.

[22] J. Loch and S. P. Singh, "Using eligibility traces to find the best memoryless policy in partially observable markov decision processes," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ser. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 323–331.

[23] P. L. Lanzi, "Adaptive agents with reinforcement learning and internal memory," in *In*. MIT Press, 2000, pp. 333–342.

[24] L. Peshkin, N. Meuleau, and L. P. Kaelbling, "Learning policies with external memory," *CoRR*, vol. cs.LG/0103003, 2001. [Online]. Available: http://arxiv.org/abs/cs.LG/0103003

[25] K. Manandhar, X. Cao, F. Hu, and Y. Liu, "Detection of faults and attacks including false data injection attack in smart grid using kalman filter," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 370–379, Dec 2014.

[26] D. B. Rawat and C. Bajracharya, "Detection of false data injection attacks in smart grid communication systems," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1652–1656, Oct 2015.

[27] 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 Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2018.

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

[29] A. Nowé, P. Vrancx, and Y.-M. De Hauwere, "Game theory and multi-agent reinforcement learning," in *Reinforcement Learning*. Springer, 2012, pp. 441–470.

[30] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 157–163.

[31] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," *AAAI/IAAI*, vol. 1998, pp. 746–752, 1998.

[32] J. Hu and M. P. Wellman, “Nash q-learning for general-sum stochastic games,” *Journal of machine learning research*, vol. 4, no. Nov, pp. 1039–1069, 2003.

[33] M. Weinberg and J. S. Rosenschein, “Best-response multiagent learning in non-stationary environments,” in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*. IEEE Computer Society, 2004, pp. 506–513.

[34] A. Abur and A. Gomez-Exposito, *Power System State Estimation: Theory and Implementation*. New York, NY: Marcel Dekker, 2004.

[35] M. Esmalifalak, H. Nguyen, R. Zheng, and Z. Han, “Stealth false data injection using independent component analysis in smart grid,” in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Oct 2011, pp. 244–248.

[36] S. Tan, D. De, W. Z. Song, J. Yang, and S. K. Das, “Survey of security advances in smart grid: A data driven approach,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 397–422, Firstquarter 2017.

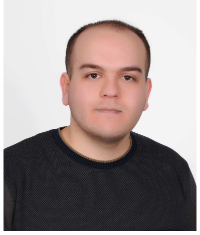
[37] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[38] N. Meuleau, L. Peshkin, K.-E. Kim, and L. P. Kaelbling, “Learning finite-state controllers for partially observable environments,” in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI’99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 427–436.

[39] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[40] M. M. Kokar and S. A. Reveliotis, “Reinforcement learning: Architectures and algorithms,” *International journal of intelligent systems*, vol. 8, no. 8, pp. 875–894, 1993.

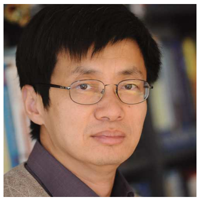
[41] R. Zimmerman, C. Murillo-SĞnchez, and R. Thomas, “Matpower: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb 2011.



Mehmet Necip Kurt received the B.S. and M.S. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with Columbia University, New York, NY, USA. His current research interests include statistical signal processing and machine learning with applications to cybersecurity and cyber-physical systems.

Oyetunji Ogundijo (S’18) received the Bachelor’s degree in Electronic and Electrical Engineering from Obafemi Awolowo University, Nigeria in 2011 and the Master’s degree in Electrical Engineering from Columbia University, New York in 2014. He has been a doctoral student in the Department of Electrical Engineering, Columbia University since 2015. His research interests lie in the areas of Machine Learning, Computational Biology, Bioinformatics, and Genomic Signal Processing.

Chong Li is a co-founder of Nakamoto & Turing Labs, Inc. He is Chief Architect and Head of Research at Canonchain Network. He is also an adjunct assistant professor at Columbia University (in the City of New York). Prior to that, he was a staff engineer at Qualcomm Research. Dr. Li received a Ph.D degree from Iowa State University and a B.E degree from Harbin Institute of Technology, both in Electrical Engineering. He is a holder of 200+ U.S. patents (granted and pending). Dr. Li has published many technical papers on top-ranking journals, including *Proceedings of the IEEE*, *IEEE Transactions on Information Theory*, *IEEE Communications Magazine*, *Automatica*, etc. He has also served as reviewer and technical program committee for most prestigious journals and conferences in communications and control society. Dr. Li has broad research interests including information theory, machine learning, networked control and communication, PYH/MAC system design for advanced telecommunication technologies (5G and beyond).



Xiaodong Wang (S’98-M’98-SM’04-F’08) received the Ph.D degree in electrical engineering from Princeton University. He is currently a Professor of electrical engineering with Columbia University, New York, NY, USA. His research interests fall in the general areas of computing, signal processing and communications, and has published extensively in these areas. Among his publications is a book entitled “Wireless Communication Systems: Advanced Techniques for Signal Reception”, published by Prentice Hall in 2003. His current research interests include wireless communications, statistical signal processing, and genomic signal processing. Dr. Wang received the 1999 NSF CAREER Award, the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award, and the 2011 IEEE Communication Society Award for Outstanding Paper on New Communication Topics. He has served as an Associate Editor for the *IEEE Transactions on Communications*, the *IEEE Transactions on Wireless Communications*, the *IEEE Transactions on Signal Processing*, and the *IEEE Transactions on Information Theory*. He is a Fellow of the IEEE and listed as an ISI Highly-cited Author.