Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.Doi Number

Invisible Manipulation: Deep Reinforcement Learning-Enhanced Stealthy Attacks on Battery Energy Management Systems

QI XIAO, (Student Member, IEEE), LIDONG SONG, (Member, IEEE), JONG HA WOO, (Student Member, IEEE), RONGXING HU (Member, IEEE), BEI XU, (Member, IEEE), KAI YE, (Student Member, IEEE), AND NING LU, (Fellow, IEEE)

Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695, USA

Corresponding author: Ning Lu (e-mail: nlu2@ncsu.edu).

This work was supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office, Award Number DE-EE0008770 and NSF ECCS, PD 18-7607 Energy, Power, Control, and Networks (EPCN), Award Number (FAIN): 2329536.

ABSTRACT This paper introduces an innovative cyber-attack scheme, "invisible manipulation," utilizing timed-stealthy false data injection attacks (Timed-SFDIAs). By subtly altering critical measurements ahead of a target period, the attacker covertly steers system operations toward a specific failure state, evading detection while enabling repeated attacks over time. Using Battery Energy Management System (BEMS) as a case study, we demonstrate the scheme's effectiveness in manipulating Battery Energy Storage Systems (BESS), critical for grids with high renewable penetration. Our method employs deep reinforcement learning (DRL) to generate synthetic measurements (e.g., battery voltage, current) that mimic real data, bypassing residual-based bad data detection (BDD) and misleading Extended Kalman-filter (EKF) based State-of-Charge (SoC) estimations. This allows the BEMS to operate the BESS per the attacker's objectives. To minimize real-time computational demands, we transform this online optimization problem into an offline DRL training problem, utilizing high-fidelity simulation data from a digital twin-based microgrid testbed. The testbed incorporates real load and solar generation profiles with BESS models in the electromagnetic transient (EMT) domain at a 100-μs resolution, capturing rapid system dynamics and ensuring robust performance in real-time scenarios. Testing on the same testbed allows real-time evaluation of microgrid responses, where the BEMS, EKF-based SoC estimation algorithms interact dynamically with the injected false measurements. This unique DRL training and testing setup not only showcases the effectiveness of the Timed-SFDIA algorithm in evading detection and achieving diverse attack objectives but also underscores the critical role of high-fidelity, digital-twin based real-time simulation testbeds. Such testbeds are invaluable for training and validating data-driven machine learning algorithms, especially when field tests and real-world validation are challenging to conduct, as they ensure robustness and adaptability under realistic operational conditions.

INDEX TERMS Cyber-physical attacks, deep reinforcement learning, timed stealthy false data injection (SFDIA), invisible manipulation attacks, state-of-charge (SoC) estimation.

I. INTRODUCTION

The secure and reliable operation of the electric grid heavily relies on accurate data for decision-making and control, making false data injection attacks (FDIAs) a significant threat. Over the past decade, research has focused on FDIAs targeting transmission systems by falsifying inputs to state estimation (SE) algorithms and evading detection by bad data detection (BDD) algorithms (e.g., residual-based BDD) [1]. With the rapid integration of distributed energy resources (DERs) [2] and other information and communication technology devices (ICTs), previously passive distribution networks are evolving into smart grids, equipped with numerous remotely accessible automated devices [3]. Consequently, the risk posed by FDIAs to modern active distribution networks (ADNs) has been growing exponentially. This

TABLE 1. Literature review of existing FDIAs targeting BESS

Attack Type	Attack Objectives	BDD (Y/N)	Description	Stealthniess	Key Advantages	Key Limitations
Rule-based	Power control [7]	**No**	Inject bias within the operation range to the active power setpoints of BESS to cause power imbalance in an islanded microgrid.	Easily detectable	Simple, low-cost implementation, feasible in real-time.	Easily detected by BDD or manual inspection.
	Mode control [8]		Falsify the mode command to disrupt the mode conversion from PQ to Vf to fail the microgrid.			
	ON/OFF control [9]		Falsify the ON/OFF command to deteriorate power quality or destabilize the power system.			
	SoC estimation [10]		Different voltage bias is selected within the operation range to disrupt the SoC estimation.			
Optimization-based	SoC estimation [11]	**Yes**	Maximize the SoC estimation error to cause overcharging or over-discharging of batteries with residual-based BDD considered.	Fully stealthy	Highly stealthy, maximizes SoC estimation error while avoiding detection.	High computational cost, requires full system knowledge, limited real-time feasibility.
Machine learning-based	BESS operation status [12],[13]	**No**	ANN based. Replicate the behavior of BESS for enhanced stealth and control the authentic BESS by employing MitM techniques.	Partially stealthy	Moderate computational cost, evades basic detection methods, feasible in real-time.	Requires extensive system data, still vulnerable to residual-based BDD.
	SoC estimation (Proposed)	**Yes**	**Timed-SFDIA. DRL-based.** Gradually injecting battery voltage and current measurement bias to cause a maximum or target SoC error at the desired time.	Fully stealthy	Offline training minimizes real-time computational load, adaptable to various scenarios, feasible in real-time.	Requires some measurement data for model tuning.

has led to an increased emphasis on studying FDIAs in the context of DERs [4].

Battery energy storage system (BESS) plays a critical role in many ADNs by providing grid-following and grid-forming functions. These functions include PV output smoothing, load shifting, and voltage and frequency regulation [5], [6]. However, the versatility and significance of BESS also make it prime target for adversaries seeking to launch FDIAs and exploit them for unlawful purposes.

As shown in Table 1, FDIA objectives targeting BESS fall into two categories: manipulation of inverter control parameters (e.g., real/reactive power, voltage, and frequency) and tampering with battery measurements and status data (e.g., state of charge, or SoC). This paper focuses on the latter, as SoC relies on estimation techniques like Coulomb counting or Kalman filtering, which depend on battery voltage and current measurements from the battery management system (BMS) [14]. This reliance makes SoC estimation vulnerable to cyber-attacks, a risk heightened by the growing use of remote communication between BESS and control centers. Additionally, integrating IoT and cloud-based technologies in BMS [15] further exposes BESS to falsified measurements.

Accurate SoC estimation is crucial for BESS energy scheduling and battery lifespan. Corrupted SoC measurements can mislead the BEMS, causing incorrect decisions, insufficient operational support, or future energy shortages. Moreover, inaccurate SoC values risk overcharging, over-discharging, reduced battery life, and potential hazards like fire or explosion [14]. Therefore, we propose a novel Timed-FDIA scheme that subtly manipulates SoC estimation, deceiving the BEMS into faulty dispatch planning and undermining the BESS's capability to support the grid effectively when the target period approaches.

FDIAs are classified by attack duration into instantaneous and prolonged types. Prolonged attacks include persistent attacks, which apply a constant bias to data, and repetitive attacks, which periodically introduce varying biases [10]. Research on FDIAs targeting BESS has primarily examined their impact on grid stability through persistent and repetitive attacks (see Table 1), often using rule-based approaches. However, the technical implementation of these attacks is rarely detailed. Rule-based FDIAs inject false data randomly within an operational range, creating persistent or repetitive biases in measurements and commands. While these attacks are relatively simple, they are easily detectable through personnel observation and conventional BDD mechanisms within SE. SE is widely used for monitoring system states, particularly in transmission system operations, with residual-based BDD algorithms commonly detecting measurement errors and cyberattacks [16]. As observability in distribution networks has improved, SE algorithms with residual-based BDD are now increasingly applied to ADNs.

In contrast, stealthy FDIAs (SFDIAs) pose a greater threat, as they are designed to bypass BDD systems. Previous studies have introduced SFDIA architectures for BESS that use man-in-the-middle (MitM) attacks to manipulate commands and measurements exchanged between the BESS controller and BMS [14], [15]. These architectures achieve stealthiness at the local controller level by employing artificial neural networks (ANN) to replicate normal BESS behavior. However, these studies do not address evading network-level detection methods like residual-based BDD, making them vulnerable to detection at the broader system level.

Recent research has revealed vulnerabilities in BDD algorithms that enable certain FDIAs, including SFDIAs, to evade detection. For example, studies [17] and [18] show that FDIAs can bypass detection if attackers have access to system parameters. Further, studies [19], [20], and [21] demonstrate that SFDIAs can succeed with only partial or local system parameter information. Additionally, parameter-free SFDIAs have been developed [22]-[24], where attackers estimate system parameters using measurements or apply tensor-shaped SE modeling techniques, effectively bypassing residual-based BDD mechanisms.

Assuming attackers have access to system parameters, [11] proposed an SFDIA scheme for BESS to maximize the absolute SoC estimation error by integrating residual-based BDD with linear optimization methods. However, this approach faced challenges due to the extensive linearization and assumptions needed to simplify the problem and reduce computational costs. These assumptions—such as treating the OCV-SoC curve slope as constant in the Extended Kalman Filter (EKF), using a linearized DC power flow instead of an AC model, and assuming future time-slot data availability—made real-time implementation challenging and increased vulnerability to detection in practical settings [22], [25].

In contrast, deep reinforcement learning (DRL) shows promise for executing real-time attacks by creating an accurate system model through offline training. Widely applied in power systems for decision-making and control [26], [27], DRL has recently gained attention for enhancing cybersecurity [28], [29]. While typically used for attack detection, DRL also serves as a powerful tool for launching cyber-attacks. This study leverages DRL to conduct stealthy attacks on SoC estimation within the BESS, bridging gaps in attack design methodology for integration with BEMS. Unlike persistent and repetitive attacks that apply a constant bias, we propose a timed-attack strategy, which gradually injects a small, varying bias at each time step. This incremental approach allows the bias to accumulate and reach a targeted error at specific times, disrupting BESS operations and potentially affecting supported systems.

The primary contribution of this paper is the development of a DRL-based approach for launching Timed-SFDIAs against SoC estimation. Unlike optimization-based methods that require extensive system data, real-time optimization, and high computing resources—often impractical for real-time applications—our DRL approach enables offline training of an attacker agent. Using results from a high-fidelity real-time simulation testbed, we generate high-quality training datasets that accurately capture microgrid dynamics. This approach allows measurements sent to the SE, BDD, and SoC estimation algorithms to reflect true system behavior without simplifications, thereby improving response accuracy.

A secondary contribution is the introduction of a timed-attack scheme targeting a specific SoC error level. By gradually falsifying SoC values well before the target period, we incrementally build an SoC error between false and actual values. This approach minimizes detection risk, enabling the BEMS to operate the BESS at attacker-defined SoC levels at the specified time. A key advantage of this method is that it remains undetected even post-attack, allowing for repeated attempts without revealing the vulnerability, as demonstrated by the "one-shot kill" scenario that compromises the reliability of BESS-dependent systems.

A third contribution is the training and testing of our algorithm using data from a high-fidelity microgrid digital-twin based testbed, which incorporates real load and solar generation profiles and models the BESS in the electromagnetic transient (EMT) domain at a 100-μs resolution. This DRL training and testing setup highlights the essential role of high-fidelity, real-time digital-twin based testbeds for training and validating data-driven machine learning algorithms, particularly when field tests and real-world validation are difficult to perform. This approach ensures robustness and adaptability under realistic operational conditions.

The remainder of this paper is organized as follows: Section II describes the ADN configuration with BESS and the BDD mechanism. Section III formulates the DRL-based Timed-SFDIA problem and details the design methodology. Section IV presents case study results from implementing the proposed attack scheme on a high-fidelity microgrid real-time simulation testbed. Section V concludes the paper.

II. Modeling Considerations

This section provides an overview of the ADN system configuration, the BDD mechanism at ADN control center, and the proposed Timed-SFDIA scheme.

A. Configuration of the ADN System

As shown in the grey box in Figure 1, the customer-owned BESS comprises a bidirectional three-phase voltage source

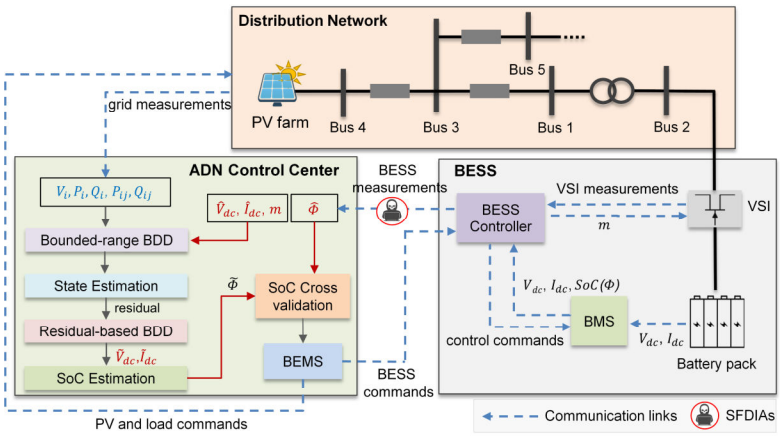


FIGURE 1. **A single-line diagram of an ADN with BESS.**

inverter (VSI), a battery pack, a battery management system (BMS), and a local BESS controller. The BMS uses local battery pack measurements, including DC terminal voltage (V_{dc}) and current (I_{dc}), to estimate the BESS SoC (Φ) and other operational variables. The BESS controller processes data from the VSI and BMS to determine the VSI modulation index (m). It then transmits V_{dc}, I_{dc}, m, and Φ to the ADN control center. This data transmission occurs over the internet or non-proprietary wireless networks, exposing it to the risk of fake data injection. For simplicity, we define the set of BESS measurements sent to the ADN control center as

$$\boldsymbol{z}_{BESS} = [V_{dc}, I_{dc}, m, \Phi] \tag{1}$$

Using the average model introduced in [30] to represent the VSI operation of the BESS unit, we have

$$V_r = mV_{dc}/\sqrt{2} \tag{2}$$

$$P_{ri} + P_{dc} + P_{loss} = 0 \tag{3}$$

$$P_{dc} = V_{dc}I_{dc} \tag{4}$$

$$P_{loss} = I_{ri}^2 R_{ac} + V_{dc}^2/R_{dc} \tag{5}$$

where P_{dc} represents the DC power, P_{ri} and I_{ri} denote AC side power and current, P_{loss} signifies inverter loss, and R_{ac}, R_{dc} are AC and DC bus series resistances, respectively.

As illustrated in the green box in Figure 1, in addition to retrieving BESS operation status from the BESS controller, the ADN control center collects measurements from the distribution network through proprietary Supervisory Control and Data Acquisition (SCADA) systems. These measurements are crucial for SE, a process that processes a set of noisy and redundant measurement data to provide an accurate real-time database for control and monitoring purposes [16]. The SCADA measurement set include vital operation parameters, such as voltage (V_i), phase angle (θ_i), active and reactive power injections at the i^{th} bus (P_i, Q_i), as well as active and reactive power flows between buses i and j (P_{ij}, Q_{ij}). This measurement set from SCADA is defined as

$$z_{SCADA} = [V_i, \theta_i, P_i, Q_i, P_{iJ}, Q_{ij}, P_i] \tag{6}$$

B. BAD DATA DETECTION

To ensure the reliability of measurements and mitigate the impact of potential cyber-attacks, the ADN control center employs a multi-layered BDD strategy. Upon receiving measurements from the BESS and SCADA systems, three distinct BDD mechanisms are sequentially applied: bounded-range BDD, residual-based BDD, and SoC cross-validation.

Figure 2 presents a flowchart that illustrates the sequential operation of these BDD mechanisms. The process operates as follows: 1) Bounded-Range Check: incoming measurements are first validated against predefined upper and lower bounds; 2) State Estimation and Residual-Based Check: if measurements pass the bounded-range check, a state estimation (SE) process is conducted. Measurement consistency is then verified using residual analysis; 3) SoC Cross-Validation: if the battery measurements pass residual-based BDD, the received SoC information is independently validated using EKF-based SoC estimation.

If any validation step fails, an alarm is triggered, and the corresponding measurements are discarded or flagged for further investigation. Only measurements passing all layers of BDD are incorporated into the ADN control center's operation.

Detailed descriptions of each BDD layer are provided below.

1) Bounded-range BDD

Bounded-range BDD applies threshold-based checks to all received measurements upon arrival at the ADN control center. Each measurement is compared against predefined upper and lower bounds, as defined in equations (7) and (8):

$$\boldsymbol{z}_{SCADA}^{min} \leq \boldsymbol{z}_{SCADA} \leq \boldsymbol{z}_{SCADA}^{max} \tag{7}$$

$$\boldsymbol{z}_{BESS}^{min} \leq \boldsymbol{z}_{BESS} \leq \boldsymbol{z}_{BESS}^{max} \tag{8}$$

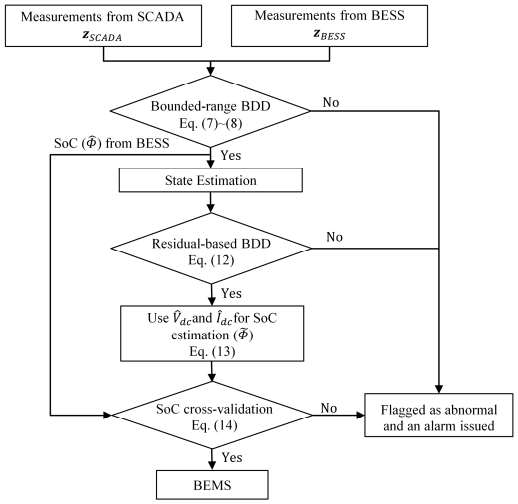


FIGURE 2. Flow chart of bad data detection at the ADN control center.

If any measurement falls outside its respective bounds, an alarm is immediately raised, indicating potential anomalies or cyber-attacks.

2) State Estimation and Residual-based BDD

Residual-based BDD is employed subsequent to bounded-range check. It verifies the validity of measurements by comparing the SE residual ($\bar{r}$) with a predefined threshold value (τ_{SE}). If both BESS and SCADA measurements pass the bounded-range BDD, the SE estimates system states ($\boldsymbol{x}$) that best match the measurements ($\boldsymbol{z}$) via:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e} \tag{9}$$

where $\boldsymbol{h}(\cdot)$ denotes a nonlinear vector function derived from network topology and $\boldsymbol{e}$ is the measurement error vector.

Using the Weighted Least Squares (WLS) method, we minimize weighted measurement residuals and iteratively solve the optimization problem detailed in [31]. Thus, the residual-based BDD mechanism validates measurements based on residual, $\bar{r}$, calculated as

$$\bar{r} = \|\boldsymbol{z} - h(\hat{\boldsymbol{x}})\|_2^2 \tag{10}$$

where $\hat{\boldsymbol{x}}$ represents the estimated value of $\boldsymbol{x}$.

If $\bar{r} \leq \tau_{SE}$, $\boldsymbol{z}$ is considered as normal; otherwise, $\boldsymbol{z}$ is flagged as containing bad data. Thus, a SFDIA that can pass the residual-based BDD, needs to generate a set of measurement attack biases $\boldsymbol{\varepsilon}$ that can satisfy

$$\|\boldsymbol{z} + \boldsymbol{\varepsilon} - h(\hat{\boldsymbol{x}} + \boldsymbol{c})\|_2^2 \leq \tau_{SE} \tag{11}$$

where $\boldsymbol{c}$ represents the malicious error introduced to the original estimation $\hat{\boldsymbol{x}}$.

Many existing methods assume that a large number of measurements in $\boldsymbol{z}$ can be modified. For example, in [11], it is required to alter not only the battery voltage and current but also other SCADA measurements to maintain the residual consistency between attacked and non-attacked cases. This assumption is restrictive, as it demands extensive access to communication links. Consequently, in this paper, we investigated SFDIA under a more realistic scenario where the attacker can only alter the battery measurements by adding an attack vector $\boldsymbol{\varepsilon}_{BESS1}$. Under this assumption, (11) becomes

$$\|\boldsymbol{z}_{SCADA} + \boldsymbol{z}_{BESS1} + \boldsymbol{\varepsilon}_{BESS1} - \boldsymbol{h}(\hat{\boldsymbol{x}} + \boldsymbol{c})\|_2^2 \leq \tau_{SE} \tag{12}$$

where $\boldsymbol{z}_{BESS1} = [V_{dc}, I_{dc}, m]$and $\boldsymbol{\varepsilon}_{BESS1}$is the attack vector containing the biases of battery voltage and current. Please note SoC cannot be directly included for SE. Thus, SoC is excluded in $\boldsymbol{z}_{BESS1}$ compared to $\boldsymbol{z}_{BESS}$.

3) SoC ESTIMATION AND CROSS-VALIDATION

In the ADN control center, the BEMS manages load consumption and power generation, optimizing performance based on the BESS SoC. However, corrupted or inaccurate SoC measurements can mislead the BEMS, leading to incorrect decisions. This may result in insufficient system support, energy shortages, overcharging, over-discharging, reduced battery lifespan, or safety risks such as fires or explosions.

Given the critical role of SoC in system operations, a cross-validation mechanism is assumed for SoC verification. The system estimates SoC ($\tilde{\Phi}$) using battery voltage ($\tilde{V}_{dc}$) and current ($\tilde{I}_{dc}$) measurements that pass residual-based BDD. It cross-validates $\tilde{\Phi}$ against the SoC value ($\hat{\Phi}$) received from the local BMS, as shown in Figure 1. The Extended Kalman filter (EKF), widely used for SoC estimation due to its ability to handle measurement errors, noise, and uncertainties, is employed in this study. The EKF-based SoC estimation process, adopted from [14], is expressed as:

$$\tilde{\Phi} = EKF(\tilde{V}_{dc}, \tilde{I}_{dc}) \tag{13}$$

Assuming the ADN control center and the local BMS use the same battery model for SoC estimation, if the discrepancy between $\tilde{\Phi}$ and $\hat{\Phi}$ is compared against a predefined threshold τ_{SoC}. If the condition:

$$|\tilde{\Phi} - \hat{\Phi}| \leq \tau_{SoC} \tag{14}$$

is satisfied, the received SoC $\hat{\Phi}$ is deemed valid and incorporated into the BEMS for scheduling battery operation. If not, an alarm is triggered.

C. STEALTHY REQUIREMENTS FOR SFDIAS TARGETING SoC ESTIMATION

To successfully execute a SFDIA targeting SoC estimation, while bypassing existing BDD mechanisms, an attacker must falsify three critical measurements: battery voltage, current and SoC. The manipulated battery voltage, current and SoC at time t are denoted as $\hat{V}_{dc}^t$, $\hat{I}_{dc}^t$ and $\hat{\Phi}^t$, respectively. The falsified voltage and current are defined as

$$\hat{V}_{dc}^t = V_{dc}^t + \sum_{t_s}^{t} \Delta V_{dc}^t \tag{15}$$

$$\hat{I}_{dc}^t = I_{dc}^t + \sum_{t_s}^{t} \Delta I_{dc}^t \tag{16}$$

where V_{dc}^t and I_{dc}^t represent the actual battery voltage and current measurements, t_s is the start time of the attack, and ΔV_{dc}^t and ΔI_{dc}^t represent the injected voltage and current bias at each time step.

To bypass detection, the falsified values $\hat{V}_{dc}^t$ and $\hat{I}_{dc}^t$ must satisfy the bounded-range BDD (17) and the residual-based BDD (18) at each time step. Here, $\boldsymbol{\varepsilon}_{BESS1}^t$ represents the cumulative bias vector as (19).

$$[V_{dc}^{min},\ I_{dc}^{min}] \le [\hat{V}_{dc}^t, \hat{I}_{dc}^t] \le [V_{dc}^{max},\ I_{dc}^{max}] \tag{17}$$

$$\|\boldsymbol{z}_{SCADA}^t + \boldsymbol{z}_{BESS1}^t + \boldsymbol{\varepsilon}_{BESS1}^t - \boldsymbol{h}(\hat{\boldsymbol{x}}^t + \boldsymbol{c}^t)\|_2^2 \le \tau_{SE} \tag{18}$$

$$\boldsymbol{\varepsilon}_{BESS1}^t = [\textstyle\sum_{t_s}^t \Delta V_{dc}^t, \sum_{t_s}^t \Delta I_{dc}^t] \tag{19}$$

If $\hat{V}_{dc}^t$ and $\hat{I}_{dc}^t$ bypass both BDDs, they will be accepted as $\tilde{V}_{dc}^t$ and $\tilde{I}_{dc}^t$ for the SoC ($\tilde{\Phi}^t$) estimation by the ADN control center. To prevent detection through SoC cross-validation in (20), the attacker must also alter the SoC value $\hat{\Phi}^t$ reported by the BESS based on the falsified $\hat{V}_{dc}^t$ and $\hat{I}_{dc}^t$, ensuring consistency with the estimated $\tilde{\Phi}^t$ from (21).

$$|\tilde{\Phi}^t - \hat{\Phi}^t| \le \tau_{SoC} \tag{20}$$

$$\tilde{\Phi}^t = EKF(\tilde{V}_{dc}^t, \tilde{I}_{dc}^t) \tag{21}$$

Failure to meet any of these conditions will result in an alarm being raised at the ADN control center.

III. METHODOLOGY

A. PROPOSED TIMED-SFDIA SCHEME TARGETING SoC ESTIMATION

This study presents a novel Timed-SFDIA scheme, termed the "one-shot kill," designed to gradually introduce errors in SoC estimation, ultimately leading to the failure of a BESS-supplied microgrid. The scheme's objective is to cause a midnight blackout by deceiving the BEMS into overestimating SoC levels, thus mismanaging energy reserves.

The attack incrementally falsifies SoC data during the operational period, misleading the BEMS into believing the battery has sufficient capacity to support the microgrid overnight. As actual SoC levels diminish below critical thresholds, the microgrid is forced to shut down. For instance, in the scenario depicted in Figure 3, the BEMS aims to maintain SoC levels of 90% at hour 18 and 45% at hour 24 for stable operation. However, falsified SoC readings may reflect compliance with these targets, while actual SoC levels drop to 70% by hour 18 and 20% by hour 24, leading to system shutdown due to insufficient energy reserves.

Timed attacks offer the strategic advantage of initiating false data injection at an earlier time t_s, well before the intended target time t_e. This approach enables the attacker to subtly alter the data stream over an extended period, reducing the likelihood of detection by the BDD process. Consequently, when the attack reaches its critical phase, the injected changes are more likely to evade detection mechanisms.

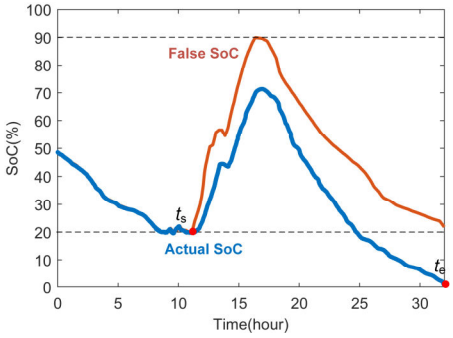


FIGURE 3. Illustration of one-shot kill attack.

To achieve the one-shot kill, we propose two attack methods: *unconstrained* and *constrained* SoC error attack. The unconstrained attack aims to maximize SoC error by t_e, where the exact error is unknown. Conversely, the constrained SoC error attack aims to inject a specific, desired SoC error by the end of attack. The attack process of both attacks is at each timestep from t_s to t_e, a bias $\boldsymbol{\varepsilon}_{BESS1}^t$, as shown in (19), is injected into $\boldsymbol{z}_{BESS1}^t$. Based on these injected biases, a false SoC $\tilde{\Phi}^t$, shown in (21), is estimated to replace the actual SoC. At each timestep, the altered data must satisfy the three BDD constraints, as outlined in (17), (18), and (20).

The inherent complexity and nonlinearity of timed-stealthy attacks, compounded by the EKF-based SoC estimator, pose challenges that require computationally intensive optimization. This optimization may necessitate accurate future system information for optimal performance. Therefore, in the following section, we propose a DRL-based method to address these complexities while maintaining system accuracy.

B. PROPOSED DRL FRAMEWORK FOR TIMED-SFDIAS

Reinforcement learning involves an agent interacting with an environment to learn optimal actions by maximizing cumulative rewards. This process is often modeled as a Markov Decision Process (MDP), where the future depends solely on the present state [32]. The MDP can be represented as a tuple $\{\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R}, \gamma\}$, where $\boldsymbol{S}$ is the environment state space, $\boldsymbol{A}$ is the action space, $\boldsymbol{P}$ denotes the transition probability, $\boldsymbol{R}$ is the reward function and $\gamma \in [0, 1]$ denotes the discount rate for the long-term return.

In this paper, Timed-SFDIAs against SoC estimation are formulated as a MDP, with the attacker modeled as a DRL agent. Figure 4 illustrates the proposed actor-critic-based DRL framework used to train this agent. The framework consists of two neural networks: the actor, which outputs an attack vector $\boldsymbol{a}_t$ based on the current observation $\boldsymbol{o}_t$; and the critic, which evaluates the quality of the selected action by estimating the state-action value function $Q(\boldsymbol{s}_i, \boldsymbol{a}_i)$. Here, the global observation $\boldsymbol{o}_t$, state $\boldsymbol{s}_t$, action set $\boldsymbol{a}_t$ and attack vector $\boldsymbol{\varepsilon}_{BESS}^t$ are defined as follows:

$$\boldsymbol{o}_t = [\boldsymbol{z}_{SCADA}^t, \boldsymbol{z}_{BESS1}^t] \tag{21}$$

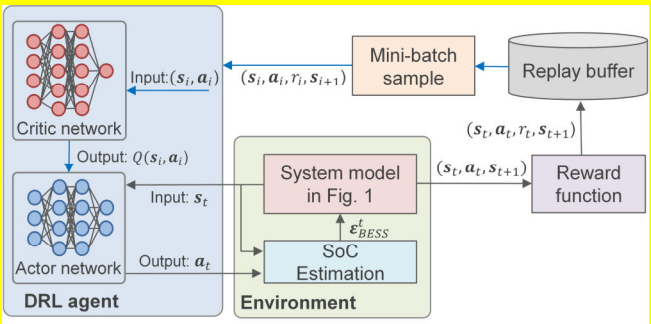


FIGURE 4. Proposed actor-critic-based DRL framework for Timed-SFDIAs against SoC estimation.

$$\boldsymbol{s}_t = \left[\boldsymbol{z}^t_{SCADA}, \boldsymbol{z}^t_{BESS1}, \boldsymbol{\varepsilon}^{t-1}_{BESS}, \frac{t}{T}\right] \tag{22}$$

$$\boldsymbol{a}_t = [\Delta V^t_{dc}, \Delta I^t_{dc}] \tag{23}$$

$$\boldsymbol{\varepsilon}^t_{BESS} = [\boldsymbol{\varepsilon}^t_{BESS1}, \boldsymbol{\varepsilon}^t_{BESS2}] \tag{24}$$

$$\boldsymbol{\varepsilon}^t_{BESS1} = [\textstyle\sum_{t_s}^{t} \Delta V^t_{dc}, \sum_{t_s}^{t} \Delta I^t_{dc}] \tag{25}$$

$$\boldsymbol{\varepsilon}^t_{BESS2} = [\Delta \Phi^t] \tag{26}$$

where $\boldsymbol{o}_t$ includes the measurements from SCADA and BESS at time t; $\boldsymbol{s}_t$ includes these measurements, the BESS attack vector taken at time $t-1$, and the ratio of the current timestep t to the total attack duration T; $\boldsymbol{a}_t$ includes the battery voltage and current bias at time t; $\boldsymbol{\varepsilon}^t_{BESS}$ comprises the accumulated battery voltage and current bias vector $\boldsymbol{\varepsilon}^t_{BESS1}$ by time t, and the SoC error $\Delta\Phi^t$ at the current time step t between the false SoC and actual SoC in $\boldsymbol{\varepsilon}^t_{BESS2}$. Note that $a_{t-1} = 0$ when $t = 1$.

The attacker's actions involve injecting the attack vector $\boldsymbol{\varepsilon}^t_{BESS}$ into the measurement from the BESS. These biases manipulate the battery voltage, current, and SoC data streams sent to the ADN controller (as shown in Figure 1).

As illustrated in Figure 4, at each timestep t, the attacker agent observes the system state $\boldsymbol{s}_t$, and the actor network generates an action $\boldsymbol{a}_t$ based on this state. The attacker can estimate the fake SoC $\widehat{\Phi}^t$ by leveraging different methods based on the false voltage $\hat{V}^t_{dc}$ and current $\hat{I}^t_{dc}$, such as using a deep neural network trained on historical data of actual SoC, voltage, and current [33]. In this scenario, it is assumed that the attackers have access to the parameters of the EKF-based SoC estimator so they can use the EKF to estimate $\widehat{\Phi}^t$ to match the SE estimated $\widetilde{\Phi}^t$, as shown in (27). The SoC bias $\Delta\Phi^t$ in $\boldsymbol{\varepsilon}^t_{BESS2}$ is calculated as shown in (28), with the actual SoC represented by (29), where V^t_{dc} and I^t_{dc} are the actual voltage and current data at time t.

$$\widehat{\Phi}^t = EKF(\hat{V}^t_{dc},\ \hat{I}^t_{dc}) \tag{27}$$

$$\Delta\Phi^t = \widehat{\Phi}^t - \Phi^t \tag{28}$$

$$\Phi^t = EKF(V^t_{dc}, I^t_{dc}) \tag{29}$$

During training, the DRL agent interacts with an environment that includes the closed-loop system model and the SoC estimator. At each timestep, the agent observes the current state $\boldsymbol{s}_t$, selects an action $\boldsymbol{a}_t$, receives a reward r_t, and transitions to the next state $\boldsymbol{s}_{t+1}$. Each transition tuple $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ is stored in a replay buffer. Once sufficient data is collected, mini-batches are randomly sampled to update the actor and critic networks. The reward function is designed to promote stealthy yet effective manipulation of the SoC estimates. Once trained, the actor network can generate real-time attack vectors without requiring future information or detailed system knowledge.

C. REWARD FUNCTION DESIGN

The key challenge of Timed-SFDIAs is to generate a sequence of attack vectors $\boldsymbol{a}_t$ that consistently evades BDD at each time step, ensuring that the desired SoC error is achieved by the end of attack period. This paper presents two reward mechanisms: unconstrained and constrained.

The objective of unconstrained attack is to maximize the SoC error at the end of attack t_e. To achieve this, if there is no BDD violation, the agent receives a reward $r_{u1,t}$ at each time step starting from t_s, based on the current injected SoC error. If, during the attack, any attack vector violates the desired system operation range (as specified in equations (17), (18), and (20)), resulting in a failure to pass any of the bounded-range, residual-based and SoC cross-validation BDDs, a penalty p_u with a large negative value is applied to the reward. At the end of attack t_e, an additional bonus r_{u2} is given.

The reward function (r_t) for the unconstrained SoC error attack are:

$$r_t = r_{u1,t} + r_{u2} + p_u \tag{30}$$

$$r_{u1,t} = \frac{\Delta\Phi^t}{k_{u2}} \times k_{u1},\ \ t \in [t_s, t_e] \tag{31}$$

$$r_{u2} = \begin{cases} 0, & t \in [t_s, t_e) \\ r_{u1,t} \times k_{u3}, & t = t_e \end{cases} \tag{32}$$

$$p_u = k_p \times F_{bdd},\ \ t \in [t_s, t_e] \tag{33}$$

In equation (31), k_{u1} denotes the sign of the desired SoC error: +1 indicates the false SoC is intended to be higher than the actual SoC, and -1 indicates lower. The coefficient k_{u2} is used as a normalization factor for $r_{u1,t}$. Equation (32) offers a bonus based on final SoC error to incentivize larger errors. In (33), k_p specifies the magnitude of the penalty, with F_{bdd} flagging BDD violation (1 for violations, 0 otherwise).

The objective of the constrained SoC error attack is to achieve and maintain a targeted SoC error ($\Delta\Phi^*$) by the end of attack. This means, once the $\Delta\Phi^*$ is reached, there is no need to inject higher SoC errors, risking being detected. Then, the reward function is adjusted as follows:

$$r_t = r_{tar1,t} + r_{tar2} + p_u \tag{34}$$

$$r_{tar1,t} = min\left\{2 - \frac{\Delta\Phi^t}{\Delta\Phi^*}, \frac{\Delta\Phi^t}{\Delta\Phi^*}\right\} \times k_{t1}, t \in [t_s, t_e] \tag{35}$$

$$r_{tar2} = \begin{cases} 0, & t \in [t_s, t_d)\ or\ (t_d, t_e) \\ r_{tar1,t} \times k_{t2} + r_{tar3}, & t = t_d\ or\ t_e \end{cases} \quad (36)$$

$$r_{tar3} = \begin{cases} 0, & |\Delta\Phi^t - \Delta\Phi^*| > 1\% \\ k_{t3}, & |\Delta\Phi^t - \Delta\Phi^*| \le 1\% \end{cases} \quad (37)$$

As shown in equation (35), the reward received at each time step $r_{tar1,t}$ is associated with how close the current SoC error $\Delta\Phi^t$ is towards the target $\Delta\Phi^*$. We use $min\{\}$ to limit its maximum value to 1. Thus, the maximum value is only achieved when the injected SoC error $\Delta\Phi^t$ equals the target $\Delta\Phi^*$. k_{t1} is a coefficient to adjust the value of $r_{tar1,t}$.

The sign of the target $\Delta\Phi^*$ determines whether the false SoC $\widehat{\Phi}^t$ is higher or lower than the actual SoC Φ^t. Specifically, if $\Delta\Phi^* > 0$, $\widehat{\Phi}^t$ is higher than Φ^t by $\Delta\Phi^*$ at t_e; Conversely, if $\Delta\Phi^* < 0$, $\widehat{\Phi}^t$ is lower than Φ^t by $\Delta\Phi^*$ at t_e.

To motivate the agent to achieve the targeted SoC error, an additional bonus r_{tar2} in (36) is given at two specific points: the designated time t_d and the end time t_e. The highest reward is given if the absolute difference between the injected SoC error and the targeted value is within 1% of SoC at these points. k_{t2} and k_{t3} are the coefficients to adjust the bonus value. The dual bonus system ensures that the agent keeps the SoC error until the end of the attack, if the target is reached at t_d.

Balancing penalties and rewards is crucial for optimal agent performance. Excessive penalties may lead to inaction, while high rewards could encourage frequent BDD violations. Additionally, balancing the attack-end bonus with cumulative rewards is necessary to achieve the desired SoC error at specific attack times.

D. Soft Actor-Critic (SAC) Framework

Based on the reward mechanism, the goal of the Timed-SFDIA is to maximize the sum of expected discounted rewards over the attack horizon of T:

$$\max_{\pi} J = \mathop{\boldsymbol{E}}_{(\boldsymbol{s}_t, \boldsymbol{a}_t)\sim\pi}\left[\sum_{t=0}^{T} (\gamma)^t \cdot r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\right] \quad (38)$$

where $\boldsymbol{E}(\cdot)$ represents the mathematical expectation, π is the actor policy that generates action according to state $\boldsymbol{s}_t$, $r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is the reward (equation (30) or (34)) based on current state $\boldsymbol{s}_t$ and action $\boldsymbol{a}_t$. In this paper, we employ soft actor-critics (SAC) algorithm in [34] to find the optimal policy.

To optimize the policy, we employ the soft actor-critic (SAC) algorithm [35], a model-free, off-policy actor-critic method that maximizes both cumulative rewards and policy entropy. This dual-objective approach improves stochastic exploration and optimization efficiency. Using SAC, Equation (38) is reformulated as:

$$\pi^* = arg\max_{\pi} \mathop{\boldsymbol{E}}_{(\boldsymbol{s}_t, \boldsymbol{a}_t)\sim\pi}\left[\sum_{t=0}^{T} (\gamma)^t \left(r_t(\boldsymbol{s}_t, \boldsymbol{a}_t) + \alpha H\big(\pi(\cdot\,|\boldsymbol{s}_t)\big)\right)\right] \quad (39)$$

where π^* represents the optimal policy, $H\big(\pi(\cdot\,|\boldsymbol{s}_t)\big) = -\log\big(\pi(\cdot\,|\boldsymbol{s}_t)\big)$ is the policy entropy, and α is the temperature parameter balancing entropy and reward.

In SAC, policy evaluation and improvement are achieved via training deep neural networks using stochastic gradient descent. SAC employs two networks: the Q network $Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t)$ approximates the state-action value function, and the policy network $\pi_\phi(\boldsymbol{a}_t|\boldsymbol{s}_t)$ approximates the policy function. The Q network parameters θ are trained by minimizing the soft Bellman residual:

$$J_Q(\theta) = \mathop{\boldsymbol{E}}_{(s_t, a_t)\sim D}\left[\frac{1}{2}\left(Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) - \left(r_t(\boldsymbol{s}_t, \boldsymbol{a}_t) + \gamma \mathop{\boldsymbol{E}}_{s_{t+1}\sim p}[V_{\bar{\theta}}(\boldsymbol{s}_{t+1})]\right)\right)^2\right] \quad (40)$$

where $V_{\bar{\theta}}(\boldsymbol{s}_{t+1})$ is the estimated soft state value using a target network updated via moving average.

For continuous action spaces, the policy is modeled as a Gaussian distribution. The policy network outputs the mean and standard deviation of the action distribution. Actor network parameters ϕ are learned by minimizing the expected Kullback-Leibler divergence [34]:

$$J_\pi(\phi) = \mathop{\boldsymbol{E}}_{\boldsymbol{s}_t\sim D, a_t\sim\pi_\phi}\left[\alpha log\left(\pi_\phi(\boldsymbol{a}_t|\boldsymbol{s}_t)\right) - Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t)\right] \quad (41)$$

TABLE 2. Pseudocode of the SAC Algorithm for Timed-SFDIAs

Initialize policy parameters ϕ, double Q-value function parameters θ_1, θ_2 and the target network parameters $\bar{\theta}_1, \bar{\theta}_2$ with θ_1, θ_2.
Initialize experience replay memory D and BDD thresholds.
while not converged
for each episode **do**
Randomly select a start point in the training dataset and obtain the initial state $\boldsymbol{s}_0$.
while not *done*
Select action $\boldsymbol{a}_t$ based on state $\boldsymbol{s}_t$ using the policy.
Input action $\boldsymbol{a}_t$ to environment, acquire *done* signal, reward r_t and next state $\boldsymbol{s}_{t+1}$.
Memorize $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1}, done)$ in experience replay buffer D.
end
end
for each gradient step **do**
Randomly sample a minibatch of transitions from D.
Update the parameters of the Q-function, the policy network, and the target network.
end
end

Two Q network critics are used to prevent value function overestimation. Details on the double Q network, policy updates, and target network mechanisms are in [34] and not discussed here due to space limitations. The SAC algorithm pseudocode for Timed-SFDIAs is presented in Table 2.

E. Timed-SFIDA Training Setup

To train the DRL agent for launching Timed-SFDIAs, the agent learns an effective attack policy by interacting with a simulated environment, as depicted in Figure 1. Since it is unrealistic to assume that attackers possess complete system information to build a direct digital-twin model of the target system, we instead assume that the attacker has access to historical system measurements from the control center of the ADN. This assumption is reasonable, as historical data could be obtained through various methods, such as MitM attacks [36], eavesdropping on communication channels [37], or hacking into the control center's data servers.

Rather than allowing the DRL agent to interact directly with the actual distribution network, BESS, and BEMS, the agent leverages this historical data for offline learning. By doing so, the attacker can train without real-time interaction with the system, mitigating the need for full system access.

To simulate real-world constraints, the BDD mechanism at the control center is incorporated into the training environment. This setup helps assess whether the generated attack vectors can bypass detection mechanisms. The parameters of bounded-range BDD are determined from the historical data and the available battery specifications, which are often accessible to attackers. Since SoC measurements are altered based on falsified battery voltage and current readings, cross-validation of SoC can easily be bypassed in this scenario.

However, the more challenging aspect is bypassing residual-based BDD, which requires the attacker to perform SE and compute residuals. To ensure that the attacks evade residual-based detection, previous studies assume that the attacker has access to system parameters [11], [17], [18]. Extending this assumption, the attacker could either have partial or local system information [19]-[21], or they could use parameter-free SE techniques, such as system topology and parameter estimation [22], [23] or tensor-based SE methods [24], based on the available measurements.

Given that the primary focus of this paper is to demonstrate how a DRL-based approach can be used to launch Timed-SFDIAs, we assume that the attacker has sufficient system information for SE as the worst-case scenario. This assumption could be relaxed in future work, with the DRL agent using parameter estimation techniques if system parameters are not fully accessible. Once trained, the DRL agent can be deployed in the actual system for online attacks.

IV. CASE STUDY

In this paper, we implement the Timed-SFDIA scheme using the BEMS framework from [38]. This setup features a grid-forming BESS with a capacity of 3 MW/12 MWh and a 4.5 MW PV farm powering an islanded microgrid. The SoC of the BESS is critical for microgrid operational planning within the BEMS, with the SoC profile over a day depicted in Figure 2. The BEMS confines the SoC to an operational range of 20% to 90%.

When the SoC approaches 20%, the BEMS initiates non-critical load shedding to maintain power only for critical loads. Conversely, exceeding 90% prompts load engagement or PV power curtailment. Insufficient PV power to recharge the BESS below 20% results in shutdown of the BESS and the microgrid.

A. Simulation Model, Dataset and HIL Testbed

We tested the proposed attack scheme on a centralized distribution model based on the IEEE 123-bus system, as depicted in Figure 1. The model includes a 4.5-MW PV farm connected to Bus 4 and a 3-MW BESS connected to Bus 2. The loads from all feeders in the IEEE-123 bus system in [38] were aggregated and modeled as a centralized load connected to Bus 5. Power consumption data for the load was sourced from actual residential users in Austin, TX. The PV farm and BESS models, developed in [39]-[43], were used, with irradiance data from actual measurements in North Carolina. The BESS is modeled using an RC-branch battery model with parameters summarized in Table 3 [44].

To replicate a practical ADN with communication capabilities, we developed a real-time simulation testbed on the OPAL-RT platform [46]. The distribution network, loads, PV farm, and BESS were modeled on OPAL-RT, while the ADN control center ran on a separate PC. A Python-based script simulated concurrent data transmission across multiple communication channels. This script, deployed on a relay PC, retrieved real-time measurement data from OPAL-RT and relayed it to the control center via TCP/IP. The simulation captured data every minute for state estimation, with control commands issued every 15 minutes to mirror real-world ADN operational cycles.

We conducted a 20-day simulation, generating measurement noises randomly using normal distributions with zero means and predefined standard deviations: 1% for real-time magnitude, 0.5% for phasor measurements, and 2% for power measurements [11]. The SCADA measurements

TABLE 3. Battery model parameters

Parameters	Values
Nominal capacity and power	12 MWh/6667 Ah, 3 MW
Nominal DC voltage and current	1800V, 1667A
DC voltage range	1607 ~ 2100V
DC current range	-1867A ~ 1867 A
R_0 (per cell)	1.3 mΩ
R_1, C_1 (per cell)	4.2 mΩ, 17111 F
Cells (series*parallel)	492*98

$\mathbf{z}_{SCADA}$ included voltage and current phasor at Bus 2, power injection of all nodes, and power flow of the 4 lines in the distribution network. Additionally, the BESS measurement $\mathbf{z}_{BESS}$ included the battery SoC, DC voltage and current, and the modulation index of the BESS inverter. There were collected and sent to the ADN control center along with the network measurements.

B. OFFLINE TRAINING

Assuming the attacker lacks access to the BEMS, we use the first 18 days of historical operation data (1-minute resolution, 25920 data points) for offline training, and the last 2 days for online testing. Both the actor and critic functions in the DRL are modeled using fully connected neural networks. The deep learning framework is implemented in PyTorch, and the algorithm is trained on an NVDIA RTX 3080 GPU. Table 4 summarizes the training hyperparameters. These parameters were selected based on a combination of prior literature [34], default configurations from widely adopted SAC implementations, and empirical tuning tailored to our specific BEMS environment. In particular, the temperature coefficient was tuned by sweeping its value over a range and evaluating the resulting trade-off between exploration and exploitation. The final setting was chosen to ensure stable policy convergence while promoting sufficiently stealthy attack strategies that avoid triggering the BDD constraints. Other parameters were also fine-tuned to enhance convergence speed and robustness across different training runs.

Table 5 details the reward parameters. During training, attacks are initiated at random hours, with the BDD trigger threshold set at 99% of the maximum residual error during normal operations. Each episode lasts 10 hours (600 steps) for both unconstrained and constrained SoC error attacks, with t_d set at the 7.7th hour for the latter. The target error in the constrained attack is randomly selected from 5% to 30% in 5% intervals. The agent is trained to maximize the SoC error or achieve the target SoC error within the attack duration. If falsified data violates any BDD constraints, a penalty is applied, and the episode ends immediately. SoC operation range constraints are removed during training since offline training uses fixed historical data with actual SoC ranging from 20% to 90%, and the BEMS cannot respond to false SoC data.

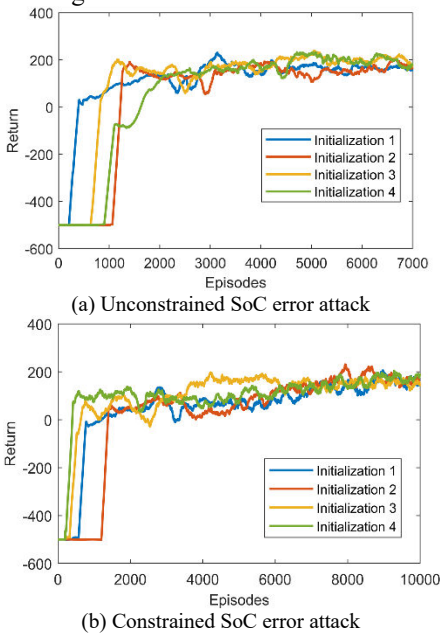
Figure 5 illustrates the mean episode return curves for the unconstrained and constrained SoC error attacks under four different random initializations, using a 200-episode sliding window. As shown in Figure 5(a), for the unconstrained case, all initializations start with low returns (around -500), primarily due to the exploration phase and frequent violation of safety constraints. The model gradually learns to achieve the attack objective while avoiding early termination, with convergence typically observed after approximately 3000 episodes. In the constrained scenario shown in Figure 5(b), convergence occurs more slowly—around 7000 episodes—due to stricter reward design and BDD constraints, which penalize aggressive attack behaviors. Despite different starting conditions, the DRL agent consistently converges across all runs, demonstrating the robustness of the training process. These results confirm that the agent can reliably learn effective attack strategies under both scenarios, with variations in convergence rate influenced by initialization and constraint tightness.

TABLE 4. Hyper parameter for DRL offline training

Hyper parameter	Values
Optimizer	Adam
Learning rate	3e-4
Discount (γ)	0.99
Replay buffer size	10^6
Number of hidden layers (all networks)	3
Number of hidden units per layer	256
Batch size	256
Target smoothing coefficient (τ)	0.005
Temperature coefficient (α)	0.5

TABLE 5. Reward Parameters

Parameters	Values	Parameters	Values
k_{u1}	+ 1	k_{t1}	0.2
k_{u2}	50	k_{t2}	2000
k_{u3}	2500	k_{t3}	100
k_p	-500		



(a) Unconstrained SoC error attack

(b) Constrained SoC error attack

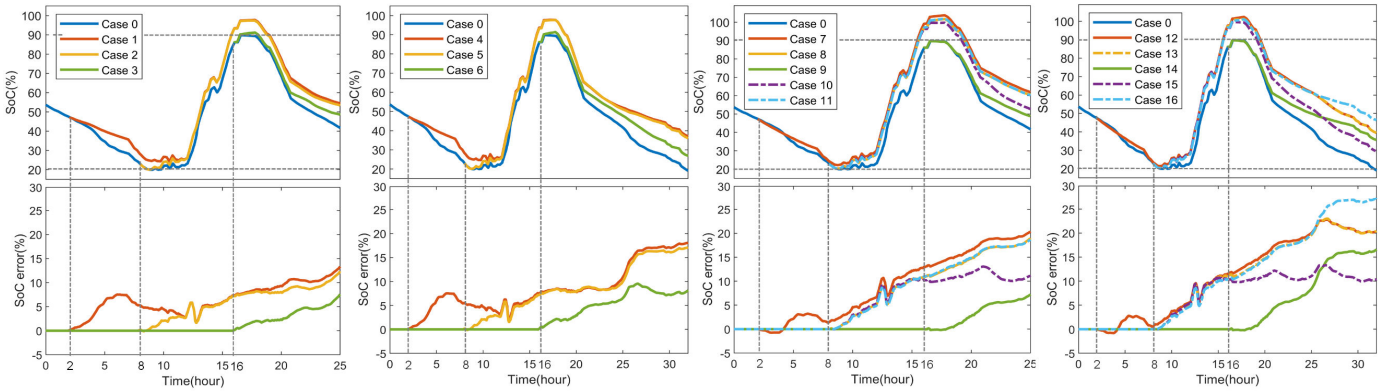
FIGURE 4. Training process of the DRL agent.

C. OFFLINE TEST

Using historical data from the last two days, we conducted an offline test within the BEMS framework [38] using a trained agent. Two attack scenarios were established: one concluding at 1 am with an expected SoC of approximately 40%, and another ending at 8 am with an anticipated SoC near 20%. These attacks were designed to introduce stealthy

TABLE 6. Offline attack test results of unconstrained and constrained SoC error attack

	No attacks	Unconstrained						Constrained									
Cases	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
t_s **(hour)**	-	2nd	8th	16th	2nd	8th	16th	2nd	8th	16th	8th	8th	2nd	8th	16th	8th	8th
t_d **(hour)**	-	-	-	-	-	-	-	21st (21 pm, 1st day)					25th (1 am, 2nd day)				
t_e **(hour)**	-	25th (1 am, 2nd day)			32nd (8 am, 2nd day)			25th (1 am, 2nd day)					32nd (8 am, 2nd day)				
$\Delta\Phi^*(\%)$	-	-	-	-	-	-	-	20			10	30	20			10	30
$\Delta\Phi_{t_d}(\%)$	-	-	-	-	-	-	-	18.3	16.5	4.6	13.1	16.6	19.9	19.6	8.4	10.9	19.7
$\Delta\Phi_{t_e}(\%)$	-	12.8	11.8	6.1	18.1	17.1	11.1	20.4	19	7.2	11.0	18.7	20.4	20.4	16.5	10.4	27.4
$\bar{r}_{med}$	0.062	0.073	0.073	0.081	0.074	0.073	0.073	0.075	0.078	0.078	0.074	0.077	0.075	0.078	0.080	0.074	0.080



(a) Case 0, 1, 2, and 3 (b) Case 0, 4, 5, and 6 (c) Case 0, 7, 8, 9, 10, and 11 (d) Case 0, 12, 13, 14, 15, and 16

FIGURE 5. SoC and SoC error profiles of offline test results with different start, end times, or targeted SoC errors: unconstrained – (a), (b); constrained – (c), (d).

SoC estimation errors that could mislead the BEMS and potentially cause unexpected shutdown of the BESS-supplied system around midnight or between 8 am and 12 pm, with the latter being a critical power-supply period.

Table 6 summarizes 16 offline attack test cases with varying start times, durations, and targeted SoC errors under constrained attack mode. Figure 6 depicts the SoC and SoC error profiles, while Figure 7 shows the residual distribution. In Figure 6, Case 0 (in all subplots) serves as the baseline scenario with no attack applied, where the SoC remains within the safe operational bounds of 20% to 90%. In attack cases, however, the agent successfully manipulates the SoC estimation to induce false higher SoC values, without violating BDD thresholds. Among the constrained cases, Case 4 achieved the largest SoC deviation, reaching an error of 18.1%. In the unconstrained cases, Cases 12, 13, and 15 effectively achieved and sustained the targeted SoC error within a ±1% tolerance at the specified attack intervals. Key observations from the offline tests in Figure 6 include:

- ***Attack durations:*** Longer attack durations or earlier attack start times usually result in larger SoC errors in unconstrained attacks, and higher chances of reaching and maintaining the targeted SoC error at the designated time t_d and the end time t_e for constrained attacks.
- ***SoC error injection:*** SoC error generally increases over time but exhibits fluctuations due to varying system statuses and BDD limits. When the actual SoC near certain plateaus, the SoC error or its injection rate tends to decrease, particularly when the actual SoC approaches the 20% plateaus over four hours. Attacks initiated at 2 am exhibit similar final SoC errors to those started at 8 am, as the accumulated SoC error before 8 am drops near zero during the 20% plateaus. Additionally, attacks concluding at the 32nd hour (8 am of the second day) show little variation in SoC error when the actual SoC is approaching 20%.

It is also observed that some false SoC values exceed 90%. This occurs because the SoC range constraint is removed for more efficient offline training. Moreover, the SoC error injection patterns under different attack scenarios are quite similar for the same system operation points. Part of the SoC error curves overlap or display similar variation trend across various attacks.

- ***Targeted SoC errors***: For constrained attacks, if the targeted SoC error can be reached and maintained at designated times correlates with the target value and attack duration. The goal can only be reached when the targeted SoC error is achievable within the specified attack duration, as shown in cases 10, 12, 13, and 15. Smaller targets are reached sooner and then fluctuate around the target value (cases 10, 15). If the target is not feasible within t_d, it may still be achieved at t_e due to the dual bonus mechanism (cases 7 and 8). The final SoC error will try to get close to

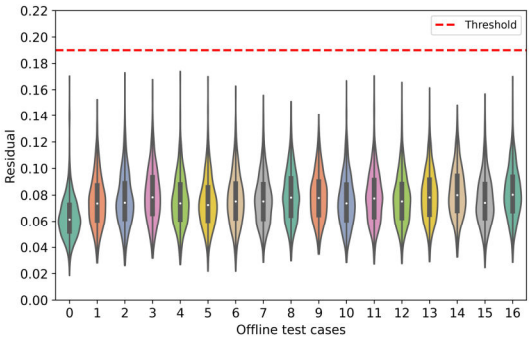


FIGURE 6. Residual distribution of unconstrained and constrained SoC error attack offline test results.

target if the target is too large, comparing to the specified attack duration (cases 9, 11, 14, and 16).

- ***Residual comparison:*** The residuals for all attacked cases are below the threshold and mostly fall within the same range as non-attacked cases. However, the median residual is higher in the attacked cases due to the injected attack vector. Longer attack durations generally result in smaller median residuals for injecting the same SoC error. For unconstrained attacks, case 3 has the highest residual median due to the shortest attack duration. For constrained attacks, both shorter attack durations and larger targeted SoC errors can increase the residual median.

Overall, it can be observed that the proposed method leverages measurement redundancy and BDD threshold margins to inject SoC errors. The residuals in the attack cases remain within the threshold range but exhibit higher median values because only the battery voltage and current are modified. If a sufficient number of measurements can be altered, the residuals could remain consistent. The proposed methodology is adaptable to such scenarios by incorporating these additional measurements into the action space.

D. Online Test

After the offline training and testing, the DRL-trained attack agent is applied to online testing using the same load and PV profile from the offline test. For the online test, the agent is implemented for real-time stealthy attacks by injecting false battery voltage, current, and SoC data to the ADN control center. The attack is done to mislead the BEMS into making inappropriate energy decisions.

The simulation results of all online test cases are summarized in Table 7. By comparing with the offline results in Table 6, it is evident that the online attack result is highly similar to those of the offline attacks. In the unconstrained attack scenario, case 4 shows the highest SoC error due to the longest attack duration. For constrained attacks, a longer attack duration increases the likelihood of achieving the targeted SoC error at two specified times.

Figure 8 presents selected online attack test cases, illustrating both the false and actual SoC profiles, along with the corresponding SoC error trajectories. In each subplot, solid and dashed lines of the same color represent the false and actual SoC, respectively. The primary distinction between offline and online attacks lies in how the actual SoC is affected. During offline attacks, the actual SoC remains within the range of 20% to 90% and is not influenced by the false SoC, as shown in Figure 6. In contrast, during online attacks, the false SoC is treated as the actual SoC by the BEMS and is regulated within the desired range. As shown in Figure 8, all false SoC is within the range of 20% to 90% but with slightly different shapes, even though the bounded-range BDD for SoC is disabled during offline training. Consequently, the actual SoC is impacted by the false SoC or the injected SoC error.

For unconstrained attacks, all cases introduced SoC errors at the desired end times (1 am and 8 am). Case 1 injected the largest SoC error at 1 am, causing the energy deficiency for the system's operation throughout the night. If the attack ends at this time, the BEMS will detect the actual SoC value. This could shut down the system when the actual SoC is below 20% and there is no PV to charge BESS or provide power for loads during the night.

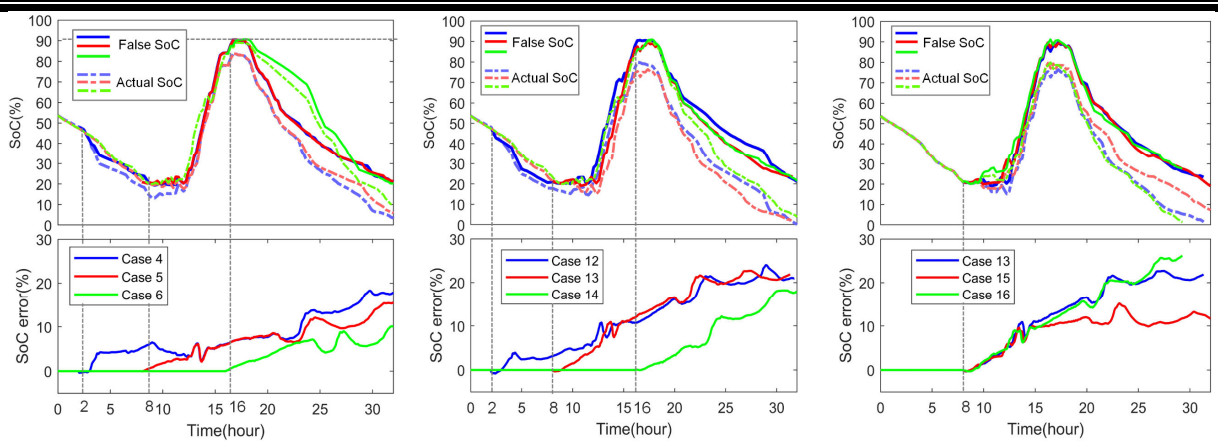
Case 4 injected the largest SoC error of 17.9% when the actual SoC is only 3.2% at 8 am. Since 8 am to 12 am is designated as a critical power supply period in the BEMS, nearly all PV power is used to supply loads while the BESS primarily provides voltage and frequency support, keeping the BESS SoC around 20% during this period. If the attack ends at 8 am, the system will shut down due to the loss of voltage and frequency support, disrupting the power supply during critical periods.

Figure 8 illustrates the real-world danger of such stealthy FDIAs: while the reported SoC remains within safe operational bounds, the actual SoC may silently fall to critical levels. This discrepancy misleads the BEMS and can trigger unexpected shutdowns during periods of high energy need. The DRL-based attack framework successfully orchestrates such scenarios without breaching traditional BDD thresholds, underscoring the need for stronger validation layers within battery energy management systems.

Similarly, for constrained attacks, the same system shutdown will happen due to the injected SoC estimation error. Comparing to the unconstrained attack, the injected SoC error in constrained attacks could be higher and more deliberate, resulting in longer periods of system shutdown.

TABLE 7. Online test results of unconstrained and constrained SoC error attack

	No attacks	Unconstrained						Constrained									
Cases	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
t_s **(hour)**	-	2nd	8th	16th	2nd	8th	16th	2nd	8th	16th	8th	8th	2nd	8th	16th	8th	8th
t_d **(hour)**	-	-	-	-	-	-	-	21st (21 pm, 1st day)					25th (1 am, 2nd day)				
t_e **(hour)**	-	25th (1 am, 2nd day)			32nd (8 am, 2nd day)			25th (1 am, 2nd day)					32nd (8 am, 2nd day)				
$\Delta\Phi^*$**(%)**	-	-	-	-	-	-	-	20			10	30	20			10	30
$\Delta\Phi_{td}$**(%)**	-	-	-	-	-	-	-	19.9	15	6.1	11	15.7	19.8	19.8	11.8	10.6	20.3
$\Delta\Phi_{te}$**(%)**	-	14.7	13.4	8.1	17.9	15.5	9.2	20.4	19.1	11.9	10.4	19.4	21	21.1	17.9	11.6	26.2
$\bar{r}_{med}$	0.062	0.074	0.074	0.081	0.074	0.072	0.076	0.075	0.076	0.081	0.075	0.078	0.072	0.075	0.079	0.074	0.079



(a) Case 4, 5, and 6 (b) Case 12, 13, and 14 (c) Case 13, 15, and 16

FIGURE 8. SoC and SoC error profiles of online test results with different start, end times, or targeted SoC errors: unconstrained – (a); constrained – (b), (c).

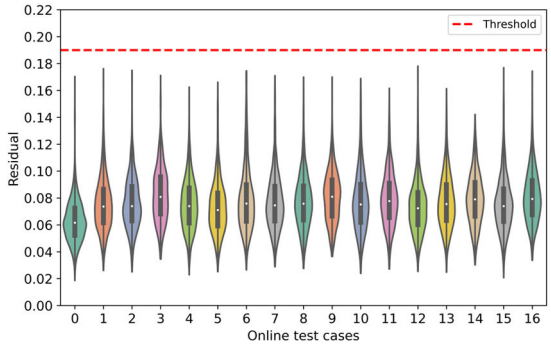


FIGURE 8. Residual distribution of unconstrained and constrained SoC error attack online test results.

Clearly, the actual SoC drops to around 0 before the attack end time in cases 12, 13 and 16.

Figure 9 presents the residual distribution for all online teste cases. All residuals remain below the BDD threshold. Compared to the non-attack case, the median residuals in the attacked cases are higher, with cases 3, 9, 11, 14 and 16 showing the highest median residuals. This is because all cases aim to reach a large SoC error target within a limited attack duration.

E. Discussions

1) PRACTICAL FEASIBILITY AND IMPACT OF TIMED-SFDIAS

This study assumes a stealthy and resourceful adversary with access to representative system data—obtained through passive eavesdropping, device compromise, or insider threats. While such access may not always be realistic in all deployments, it aligns with threat models involving advanced persistent threats frequently observed in critical infrastructure sectors. Furthermore, we assume that the attacker possesses sufficient offline computational resources to train a DRL-based attack policy—an increasingly plausible scenario given the availability of scalable cloud computing platforms. Once trained, the attack strategy can be executed in real time with minimal computational overhead.

Simulation results confirm that undetected Timed-SFDIAs can significantly disrupt BEMS by manipulating SoC estimates, leading to erroneous control decisions such as overcharging or undercharging. These disruptions can compromise system safety, operational continuity, and energy availability. Over time, persistent SoC manipulation could accelerate battery degradation, destabilize load balancing, and reduce system resilience during peak demand or outages.

The risks are even more pronounced in interconnected microgrids, where coordinated control and energy-sharing protocols rely on accurate SoC information. Tampered SoC signals could interfere with distributed decision-making, triggering cascading failures across multiple microgrids or

even wider grid segments. These results highlight the systemic vulnerabilities introduced by stealthy attacks and emphasize the need for proactive anomaly detection, system hardening, and cyber-resilient energy management frameworks.

Future work will expand this framework to explore broader system-level consequences, including the impact on inverter coordination, stability of secondary control loops, and grid service reliability.

2) COMPARISON WITH EXISTING FDIA TECHNIQUES

Existing false data injection attack techniques—including rule-based, machine learning-based, and optimization-based approaches—each have distinct limitations. Rule-based attacks typically manipulate sensor measurements directly but are easily detected, as they often overlook modern BDD mechanisms. Machine learning-based approaches, such as ANN-driven attacks, can fool local controllers but are frequently flagged by residual-based detection schemes at higher control layers.

Optimization-based methods offer theoretical stealth by formulating attacks that bypass BDD constraints. However, they rely on restrictive assumptions, such as a known and static OCV-SoC relationship or access to future measurements—conditions rarely satisfied in real-time deployments.

In contrast, the proposed DRL-based attack offloads the optimization process to the training phase. This enables real-time deployment with greater stealth and adaptability. Notably, the DRL framework supports dynamic and timed attack modes that are difficult to realize using conventional optimization-based techniques. As such, a direct quantitative comparison with traditional methods is not included, as the capabilities and operating assumptions differ fundamentally.

3) SCALABILITY OF THE PROPOSED ATTACK IN LARGER POWER SYSTEMS

Although this study focuses on a single microgrid with one BESS and a PV system, the proposed attack strategy is inherently scalable to larger power systems with multiple interconnected BESS units. The feasibility of such attacks depends less on system size and more on factors like the structure of the state estimation process, measurement redundancy, and the configuration of BDD mechanisms.

As system scale increases, higher observability and measurement redundancy may strengthen BDD performance, potentially reducing attack success rates. However, prior research has demonstrated that attackers can exploit sparsely observed regions, coordinate deviations across nodes, or apply parameter-free strategies to maintain stealth even in complex systems. Moreover, larger systems often rely on distributed control architectures, which, while offering fault-tolerance, may introduce additional attack surfaces and new challenges in coordination and detection.

To rigorously assess the scalability of the proposed framework, future research will extend testing to multi-BESS networks and larger grid configurations. This includes evaluating performance under different state estimation models, investigating cross-node coordination of distributed attacks, and testing the strategy in HIL or real-time simulation environments. Such work will provide deeper insight into the attack's practical implications and inform the development of more resilient cyber-physical defenses for energy storage systems.

4) PREVENTION, DETECTION, AND DEFENSE FOR TIMED-SFDIAS

Timed Timed-SFDIAs represent a serious threat to the integrity and reliability of BESS by gradually introducing subtle measurement errors that evade conventional BDD mechanisms. To counteract these risks, comprehensive strategies encompassing prevention, detection, and defense are essential.

Prevention begins with reinforcing communication security, data integrity, and system-level resilience within BMS and BEMS [46]. Secure communication protocols—such as end-to-end encryption and mutual authentication—must be employed to protect data exchanges between BESS components and central controllers from unauthorized interception or tampering. Cybersecurity hardening at both the hardware and software levels (e.g., secure boot, intrusion detection systems, and firmware integrity checks) enhances resistance to potential intrusions. Additionally, incorporating system-level redundancy and fail-safes can help safeguard critical measurements—particularly SoC data—from compromise or corruption.

Early detection is crucial for minimizing operational and safety impacts. Advanced anomaly detection algorithms, leveraging statistical methods or machine learning techniques, can be deployed to identify irregularities in SoC estimation that may signal ongoing attacks. Cross-validation of measurements using multiple sensors or independent estimation models increases robustness and detection sensitivity. Furthermore, time-series analysis can reveal subtle, temporally aligned deviations in SoC behavior, consistent with gradual manipulation strategies employed in Timed-SFDIAs.

Defense and mitigation strategies focus on reducing the impact of successful attacks and enabling rapid system recovery. Real-time monitoring and response mechanisms can trigger automated protective actions, such as dynamically adjusting charging/discharging profiles, isolating compromised components, or notifying operators for manual intervention. Enhancing the robustness of SoC estimation through hybrid or ensemble modeling—such as combining model-based approaches with data-driven techniques—can further improve resilience to adversarial manipulation. A resilient BMS architecture that supports fault tolerance and graceful degradation ensures continued safe operation under uncertainty or partial system compromise.

By integrating secure communication frameworks, advanced anomaly detection techniques, and real-time mitigation protocols, the overall cyber-physical resilience of BESS can be significantly strengthened. These measures are critical to safeguarding modern energy storage infrastructures against sophisticated threats such as Timed-SFDIAs and ensuring the reliable and secure operation of power systems in adversarial environments.

V. Conclusion

To address the substantial computational demands of nonlinear Timed-SFDIAs while ensuring real-time online deployment, we proposed a DRL-based Timed-SFDIA algorithm specifically designed to disrupt BESS operations during targeted time periods. This innovative algorithm gradually degrades BESS SoC estimation by strategically altering battery voltage and current, resulting in significant SoC deviations over time. Our method exploits measurement redundancy and BDD threshold margins to effectively inject potential SoC errors.

The RL agent, through interaction with the ADN environment incorporating three distinct BDD algorithms, learns to generate a sequence of attack vectors for Timed-SFDIA attacks. These vectors are capable of evading BDD detection, successfully introducing the desired SoC error by the end of the attack period. We introduced two distinct attack modes: unconstrained and constrained SoC error attacks. The constrained mode allows for precise control of the injected SoC error, maintaining it within a targeted range, while the unconstrained mode aims to generate the largest possible errors.

Our proposed attack methodology and strategies were rigorously tested using a HIL platform. The results demonstrated the effectiveness of our approach in injecting the desired SoC error without triggering any BDD mechanisms. This injected error can cause severe consequences, including power shortages and system shutdowns during critical periods in BESS-supported microgrids. A distinct advantage of this method is its low detectability—even after a shutdown due to insufficient SoC—since operators may attribute the low SoC to natural battery degradation, as no SE alarms are triggered throughout the event. This allows the attack to be relaunched at any future time. This confirms the potential of our DRL-based Timed-SFDIA algorithm to meet real-time deployment requirements while effectively compromising BESS operations. The success of this method underscores the need for enhanced BDD mechanisms to counteract sophisticated SFDIA threats and safeguard BESS integrity.

APPENDIX A
LIST OF ABBREVIATIONS

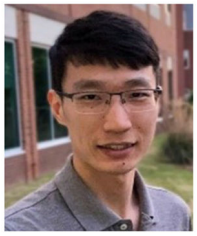
ADN: Active Distribution Network
ANN: Artificial Neural Network
BDD: Bad Data Detection
BEMS: Battery Energy Management System
BESS: Battery Energy Storage System
BMS: Battery Management System
DER: Distributed Energy Resource
DRL: Deep Reinforcement Learning
SFDIAs: Stealthy False Data Injection Attacks
EKF: Extended Kalman-filter
EMT: Electromagnetic Transient
ICT: Information and Communication Technology
IoT: Internet of Things
MDP: Markov Decision Process
MitM: Man-in-the-middle
OCV: Open Circuit Voltage
SAC: Soft Actor-Critic
SE: State Estimation
SoC: State of Charge
SCADA: Supervisory Control and Data Acquisition
VSI: Voltage Source Inverter
WLS: Weighted Least Squares

REFERENCES

[1] A. S. Musleh, G. Chen, and Z. Y. Dong, "A survey on the detection algorithms for false data injection attacks in smart grids," IEEE Transactions on Smart Grid, vol. 11, no. 3, pp. 2218-2234, 2019.
[2] Q. Xiao, P. Mattavelli, A. Khodamoradi, P. Loh. "Modelling and analysis of equivalent SISO dq impedance of grid-connected converters," Mathematics and Computers in Simulation, 184:5-20, 2021.
[3] H. He, et al., "Cyber physical attacks and defenses in the smart grid: a survey," IET Cyber Physical Systems: Theory & Applications,13-27, 2016.
[4] P. Zhuang, R. Deng and H. Liang, "False Data Injection Attacks Against State Estimation in Multiphase and Unbalanced Smart Distribution Systems," IEEE Trans. on Smart Grid, vol. 10, no. 6, 2019.
[5] Q. Xiao et al., "Assessment of Transmission-level Fault Impacts on 3-phase and 1-phase Distribution IBR Operation," 2024 IEEE Power & Energy Society General Meeting (PESGM), Seattle, WA, USA, 2024, pp. 1-5, doi: 10.1109/PESGM51994.2024.10688676.
[6] B. Xu, V. Daldegan Paduani, Q. Xiao, L. Song, D. Lubkeman and N. Lu, "Under-Frequency Load Shedding for Power Reserve Management in Islanded Microgrids," in IEEE Transactions on Smart Grid, vol. 15, no. 5, pp. 4662-4673, Sept. 2024, doi: 10.1109/TSG.2024.3393426.
[7] M. Chlela, et al., "Real-time testing platform for microgrid controllers against false data injection cybersecurity attacks," 2016 IEEE Power and Energy Society General Meeting (PESGM), Boston, USA, 2016, pp. 1-5.
[8] X. Liu, M. Shahidehpour, et al., "Microgrid risk analysis considering the impact of cyber attacks on solar PV and ESS control systems, " IEEE Trans. Smart Grid, vol. 8, no. 3, pp. 1330–1339, May 2017.
[9] S. Ghosh, M. H. Ali and D. Dasgupta, "Effects of Cyber-Attacks on the Energy Storage in a Hybrid Power System," 2018 IEEE Power & Energy Society General Meeting (PESGM), Portland, OR, USA, 2018, pp. 1-5.
[10] V. Obrien, R. D. Trevizan and V. S. Rao, "Detecting False Data Injection Attacks to Battery State Estimation Using Cumulative Sum Algorithm," 2021 NAPS, College Station, TX, USA, 2021, pp. 01-06.
[11] P. Zhuang, H. Liang, "False data injection attacks against state-of-charge estimation of battery energy storage systems in smart distribution networks," IEEE Trans. on Smart Grid, vol. 12, no. 3, pp. 2566-2577, 2020.
[12] M. Pasetti et al., "Artificial Neural Network-Based Stealthy Attack on Battery Energy Storage Systems," in IEEE Trans. on Smart Grid, vol. 12, no. 6, pp. 5310-5321, Nov. 2021.

[13] A. O. De Sá, et al., "ANN-Based Stealthy Attack to Battery Energy Storage Systems by Using a Low-Cost Device," 2022 IEEE International Workshop on Metrology for Industry 4.0 & IoT, Trento, Italy, 2022, pp. 201-206.
[14] Z. Chen, Y. Fu, and C. Mi, "State of charge estimation of lithium-ion batteries in electric drive vehicles using extended Kalman filtering," IEEE Transactions on Vehicular Technology, vol. 62, no. 3, pp. 1020-1030, 2012.
[15] S. Kumbhar, T. Faika, D. Makwana, T. Kim, and Y. Lee, "Cybersecurity for battery management systems in cyber-physical environments," in Proc. IEEE Trans p. Electrific. Conf. Expo., Jun. 13-15, 2018, pp. 934–938.
[16] R. Deng, G. Xiao, R. Lu, et al., "False Data Injection on State Estimation in Power Systems—Attacks, Impacts, and Defense: A Survey," in IEEE Trans. on Industrial Informatics, vol. 13, no. 2, pp. 411-423, April 2017.
[17] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," ACM Transactions on Information and System Security, vol. 14, no. 1, pp. 1–33, May. 2011.
[18] C. Liu, H. Liang, and T. Chen, "Network parameter coordinated false data injection attacks against power system ac state estimation," IEEE Transactions on Smart Grid, vol. 12, no. 2, pp. 1626–1639, Mar. 2021.
[19] J. Zhang, Z. Chu, L. Sankar, and O. Kosut, "Can attackers with limited information exploit historical data to mount successful false data injection attacks on power systems?" IEEE Transactions on Power Systems, vol. 33, no. 5, pp. 4775–4786, Sep. 2018.
[20] J. Liang, L. Sankar, and O. Kosut, "Vulnerability analysis and consequences of false data injection attack on power system state estimation," IEEE Transactions on Power Systems, vol. 31, no. 5, pp. 3864–3872, Sep. 2016.
[21] X. Liu and Z. Li, "Local topology attacks in smart grids," IEEE Transactions on Smart Grid, vol. 8, no. 6, pp. 2617–2626, Nov. 2017.
[22] M. Du, G. Pierrou, X. Wang, and M. Kassouf, "Targeted false data injection attacks against ac state estimation without network parameters," IEEE Transactions on Smart Grid, vol. 12, no. 6, pp. 5349–5361, Aug. 2021.
[23] Y. Guo, Y. Yuan and Z. Wang, "Distribution Grid Modeling Using Smart Meter Data," in IEEE Transactions on Power Systems, vol. 37, no. 3, pp. 1995-2004, May 2022, doi: 10.1109/TPWRS.2021.3118004.
[24] H. Yang, W. Zhang, Z. Liang, Z. Wang, C. Y. Chung and Q. Wang, "Parameter-Free False Data Injection Attack Against AC State Estimation: A Canonical Polyadic Decomposition Based Approach," in IEEE Transactions on Power Systems.
[25] R. Jiao, G. Xun, X. Liu and G. Yan, "A New AC False Data Injection Attack Method Without Network Information," in IEEE Transactions on Smart Grid, vol. 12, no. 6, pp. 5280-5289, Nov. 2021.
[26] S. Zhang, M. Zhang, R. Hu, D. Lubkeman, Y. Liu and N. Lu, "Reinforcement Learning for Volt- Var Control: A Novel Two-stage Progressive Training Strategy," 2022 IEEE Power & Energy Society General Meeting (PESGM), Denver, CO, USA, 2022, pp. 1-5.
[27] J. H. Woo, L. Wu, S. M. Lee, J. -B. Park and J. H. Roh, "D-STATCOM d-q Axis Current Reference Control Applying DDPG Algorithm in the Distribution System," in IEEE Access, vol. 9, pp. 145840-145851, 2021.
[28] D. An, Q. Yang, W. Liu and Y. Zhang, "Defending Against Data Integrity Attacks in Smart Grid: A Deep Reinforcement Learning-Based Approach," in IEEE Access, vol. 7, pp. 110835-110845, 2019.
[29] Y. Wang, B. C. Pal, "Destabilizing Attack and Robust Defense for Inverter-Based Microgrids by Adversarial Deep Reinforcement Learning," in IEEE Trans. on Smart Grid, vol. 14, no. 6, pp. 4839-4850, Nov. 2023.
[30] A. de la Villa Jaén, E. Acha, and A. G. Expósito, "Voltage source converter modeling for power system state estimation: STATCOM and VSC-HVDC," IEEE Trans. on power systems, vol.23, no.4, pp. 1552-1559, 2008.
[31] Z. Wang, H. He, Z. Wan and Y. Sun, "Detection of False Data Injection Attacks in AC State Estimation Using Phasor Measurements," in IEEE Trans. on Smart Grid, doi: 10.1109/TSG.2020.2972781.
[32] Z. Zhang, D. Zhang, and R. C. Qiu, "Deep reinforcement learning for power system applications: An overview," CSEE Journal of Power and Energy Systems, vol. 6, no. 1, pp. 213-225, 2019.
[33] X. Song, F. Yang, D. Wang, K. Tsui, "Combined CNN-LSTM network for state-of-charge estimation of lithium-ion batteries," IEEE Access, 7:88894-902, Jul. 2019.
[34] T. Haarnoia et al., "Soft actor-critic algorithms and applications," 2018, arXiv: 1812.05905.
[35] L. Yan, et al., "Deep Reinforcement Learning for Continuous Electric Vehicles Charging Control With Dynamic User Behaviors," in IEEE Trans. on Smart Grid, vol. 12, no. 6, pp. 5124-5134, Nov. 2021.
[36] P. Wlazlo, A. Sahu, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz. "Man‐in‐the‐middle attacks and defence in a power system cyber‐physical testbed." IET Cyber‐Physical Systems: Theory & Applications 6, no. 3 (2021): 164-177.
[37] M. Z. Gunduz, and D. Resul, "Cyber-security on smart grid: Threats and potential solutions," Computer networks, 169: 107094, 2020.
[38] R. Hu, A. Shirsat, V. Muthukaruppan, et al, "Adaptive cold-load pickup considerations in 2-stage microgrid unit commitment for enhancing microgrid resilience, " in Applied Energy, 2024, 356: 122424.
[39] V. D. Paduani, H. Yu, B. Xu and N. Lu, "A Unified Power-Setpoint Tracking Algorithm for Utility-Scale PV Systems With Power Reserves and Fast Frequency Response Capabilities," in IEEE Trans. on Sustainable Energy, vol. 13, no. 1, pp. 479-490, Jan. 2022.
[40] B. Xu, et al., "A novel grid-forming voltage control strategy for supplying unbalanced microgrid loads using inverter-based resources," in 2022 IEEE Power & Energy Society General Meeting (PESGM), 2022, pp. 1-5.
[41] Q. Xiao, F. Tang, Z. Xin, J. Zhou, P. Chen, and Loh, P. Loh, "Large time-delay decoupling and correction in synchronous complex-vector frame," IET Power Electronics, 12: 254-266.
[42] Q. Xiao, P. Mattavelli, A. Khodamoradi and F. Tang, "Analysis of transforming dq impedances of different converters to a common reference frame in complex converter networks," in CES Transactions on Electrical Machines and Systems, vol. 3, no. 4, pp. 342-350, Dec. 2019, doi: 10.30941/CESTEMS.2019.00046.
[43] J. Woo, Q. Xiao, V. Paduani, N. Lu, "A Two-Stage Optimization Method for Real-Time Parameterization of PV-Farm Digital Twin," arXiv preprint arXiv:2410.04244, 2024 Oct 5.
[44] V. Paduani, Q. Xiao, B. Xu, D. Lubkeman, and N. Lu, "Optimal Control Design for Operating a Hybrid PV Plant with Robust Power Reserves for Fast Frequency Regulation Services", arXiv preprint arXiv:2212.03803.
[45] Q. Xiao, J. Woo, L. Song, N. Lu, and V. Paduani, "Design and Implementation of Scalable Communication Interfaces for Reliable and Stable Real-time Co-Simulation of Power Systems", arXiv preprint arXiv:2502.07866.
[46] A. Sahu, K. Davis, H. Huang, A. Umunnakwe, S. Zonouz, and A. Goulart. "Design of next-generation cyber-physical energy management systems: Monitoring to mitigation." IEEE Open Access Journal of Power and Energy 10 (2023): 151-163.

Qi Xiao (Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Beijing Jiaotong University, Beijing, China, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree with North Carolina State University, Raleigh, NC, USA. From 2018 to 2019, he was a Visiting Scholar with the University of Padova, Padova, Italy. His research focuses on the modeling and control of inverter-based resources, specializing in battery management systems, battery energy storage systems, and the application of machine learning.



Lidong Song (Member, IEEE) received the B.S. degree in electrical engineering from the China University of Mining and Technology, Beijing, in 2016, and the M.S. degree in electrical engineering from Xi'an Jiaotong University, in 2019. He is currently pursuing the Ph.D. degree in electrical engineering with North Carolina State University, Raleigh, USA. His research interests include deep learning application and data-driven model development in distribution systems. He is currently working on demand side data analysis, including load profile super-resolution, non-intrusive load monitoring, and baseline load estimation.

Jong Ha Woo received his B.S. and M.S. degrees in Electrical Engineering from Konkuk University, Seoul, South Korea, in 2020 and 2022, respectively. He is currently pursuing a Ph.D. in Electrical Engineering at North Carolina State University, Raleigh, NC, United States, under the supervision of Prof. Ning Lu. His research interests include power system optimization and electromagnetic transient (EMT) simulation using OPAL-RT.

Rongxing Hu received the B.S. and the M.S. degrees in electrical engineering from the South China University of Technology, Guangzhou, China, in 2013 and 2016, respectively, and Ph.D. degree in electrical engineering with North Carolina State University, Raleigh, NC, USA, in 2024. From 2016 to 2019, he was an Electrical Engineer at the China Southern Power Grid Co., Ltd., Foshan, China. His research interests include distribution system operation, microgrid energy management, demand response, and power system resilience.

Bei Xu (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Beijing Jiaotong University, Beijing, China, in 2016 and 2019, respectively, and the Ph.D. degree in electrical engineering from North Carolina State University, Raleigh, NC, USA, in 2023. She is currently a Control and Algorithm Engineer with Element Energy, Inc., Menlo Park, CA, USA. Her research interests include microgrid hardware-in-the-loop testbed development, battery energy storage systems modeling and control, and battery management system design.

Kai Ye (Student Member, IEEE) received the B.S. degree in new energy science and engineering from the Chinese University of Hong Kong, Shenzhen, China, in 2019, and the M.S. degree in electrical and computer engineering from University of Minnesota, Minneapolis, MN, USA in 2020. Currently he is a Ph.D. candidate in electrical and computer engineering with the Future Renewable Electric Energy Delivery and Management (FREEDM) Systems Center, North Carolina State University, Raleigh, NC, USA. His research interests include integration of distributed energy resources and data-driven load modeling in distribution systems.

Ning Lu (Fellow, IEEE) received the Ph.D. degree in electric power engineering from Rensselaer Polytechnic Institute in 2002. She is a Professor with the Department of Electrical and Computer Engineering, North Carolina State University. She served as a Senior Research Engineer with Pacific Northwest National Laboratory from 2003 to 2012. Her research focuses on power system load modeling, microgrid modeling and control, and the development of real-time and faster-than-real-time large-scale co-simulation systems. Additionally, she explores the application of machine learning in power system data analysis, modeling, and control.