


Detecting Attacks and Estimating States of Power Grids from Partial Observations with Machine Learning

Zheng-Meng Zhai¹,¹ Mohammadamin Moradi¹,¹ and Ying-Cheng Lai^{1,2,*}

¹*School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, Arizona 85287, USA*

²*Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*

 (Received 3 July 2024; revised 1 December 2024; accepted 15 January 2025; published 4 February 2025)

The ever-increasing complexity of modern power grids makes them vulnerable to cyber and/or physical attacks. To protect them, accurate attack detection is essential. A challenging scenario is that a localized attack has occurred on a specific transmission line but only a small number of transmission lines elsewhere can be monitored. That is, full state observation of the whole power grid is not feasible, so attack detection and state estimation need to be done with only limited, partial state observations. We articulate a machine-learning framework to address this problem, where the necessity to deal with sequential time-series data with dynamical memories and to avoid a vanishing gradient has led us to choose the long short-term memory (LSTM) architecture. Leveraging the inherent capabilities of LSTM to handle sequential data and capture temporal dependencies, we demonstrate, using three benchmark power-grid networks, that the complete dynamical state of the whole power grid can be faithfully reconstructed and the attack can be accurately localized from limited, partial state observations even in the presence of noise. The performance improves as more observations become available. Further justification for using the LSTM is provided by our comparing its performance with that of alternative machine-learning architectures such as feedforward neural networks and random forest. Despite the gigantic existing literature on applications of LSTM to power grids, to our knowledge, the problem of locating an attack and estimating the state from limited observations had not been addressed before our work. The method developed can potentially be generalized to broad complex cyber-physical systems.

DOI: [10.1103/PRXEnergy.4.013003](https://doi.org/10.1103/PRXEnergy.4.013003)

I. INTRODUCTION

In complex physical systems consisting of many interconnected components, localized disturbances disrupting or even disabling the system functioning such as random perturbations or intentional attacks are expected to occur from time to time. An example is the modern power grids, a class of cyber-physical systems that contain a physical component with transformers and generators as well as a cyber component with sensors, control systems, and communication networks [1,2]. In a power grid, random disturbances can occur but they are typically local, so are intentional attacks that often target some particular transmission lines or generating stations. Because of the scale of the system, a full state observation is often not feasible because it is practically impossible to observe or monitor

all the dynamical variables. What is possible is limited, often quite limited, partial state observations. When a random disturbance or an attack occurs at a location or on a part of the system that does not contain any dynamical variables under direct observation, how can the disturbance or attack be accurately detected and located from partial state observations of a small number of dynamical variables elsewhere?

The problem of detecting and locating the source of disturbance based on partial state observations elsewhere is challenging, even when the governing equations of the system are available. Suppose the system is described by a set of nonlinear differential equations of the form $d\mathbf{X}/dt = \mathbf{F}(\bar{\mathbf{X}}, \mathbf{x}_D, \mathbf{x}_O)$, where \mathbf{X} is the full state vector, \mathbf{x}_D is the set of disturbed variables, \mathbf{x}_O is the set of variables under observation, and $\bar{\mathbf{X}}$ denotes the set of variables in \mathbf{X} excluding \mathbf{x}_D and \mathbf{x}_O . We assume that the vectors \mathbf{x}_D and \mathbf{x}_O are distinct and do not overlap with each other, and that their dimensions are much smaller than that of \mathbf{X} , as the disturbance is assumed to be localized and the observation is partial and limited. The problem of locating \mathbf{x}_D from partial state observations can be stated as follows. Suppose a change in \mathbf{x}_O has been observed: $\mathbf{x}_O \rightarrow \mathbf{x}_O + \Delta\mathbf{x}_O$. Can

*Contact author: ying-cheng.lai@asu.edu

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

\mathbf{x}_D be inferred from $\Delta\mathbf{x}_O$? If the vector field $\mathbf{F}(\bar{\mathbf{X}}, \mathbf{x}_D, \mathbf{x}_O)$ is nonlinear, this is, in general, not possible in a traditional sense. One may attempt to build up a disturbance-of-attack “library” that lists all possible $\Delta\mathbf{x}_O$ when \mathbf{x}_D is disturbed, but this is not feasible either, especially when the system is large or high-dimensional. In real-world applications, the governing equations are usually not available. In this case, inferring \mathbf{x}_D from $\Delta\mathbf{x}_O$ needs to be done in a fully data-driven manner: on the basis of only on the time series \mathbf{x}_O , how can \mathbf{x}_D be inferred? To our knowledge, this is a challenging problem with no conventional solution.

In recent years, data-driven approaches to reconstructing the unknown topology of complex networks have been developed [3–10]. However, the problem to be addressed in this paper is not reconstructing complex-network topology, but rather, is detecting and locating attacks on power-grid networks using limited partial state observations. Here we present a machine learning–based approach to detecting and locating the source of disturbance or attack as well as state estimation from partial state observations. A basic question is what neural-network architecture is suitable for this problem. Our choice is long short-term memory (LSTM) [11] neural networks, a class of recurrent neural networks (RNNs) [12–14]. There are two reasons for this choice. First, we assume that observation of some dynamical variables of the physical system of interest, e.g., a power grid, will generate sequential time-series data. RNNs are fundamentally designed to handle such data. Differing from feedforward neural networks (FNNs), RNNs have loops in their architecture that allow information to be passed from one step in the sequence to the next, allowing them to capture the temporal dependencies in the data. This virtue makes RNNs well suited for tasks such as speech recognition [15,16], natural language processing [17–19], time-series forecasting [20–29], and signal processing and filtering [30–32]. Second, a common difficulty with RNNs is vanishing gradients. When this occurs, the network can no longer learn the time dependencies in the data, presenting a difficulty for our problem of detecting and locating disturbances. The requirement is then that the neural-network architecture maintain its ability to capture the time dependencies in time. LSTM neural networks are designed to mitigate the issue of vanishing gradients. This is achieved by a complex cell structure that allows the neural networks to selectively forget or remember previous inputs, allowing them to capture the long-term dependencies without encountering the problem of vanishing gradients. Overall, LSTMs can be particularly effective in capturing the long-term dependencies in the network data, allowing them to detect subtle changes in the network behavior that may be indicative of a disturbance or an attack. We emphasize that the aim of this work is the development of a machine-learning framework for detecting attacks and estimating states from partial

observations, rather than outperforming the state-of-the-art machine-learning methods for tasks such as regression, classification, and network reconstruction.

To describe our work in a concrete setting, we focus on a major class of cyber-physical systems—power grids. In the modern world, the ability to better monitor and control the power generating, transmission, and distribution systems is important. To prevent failures, especially cascading grid failures caused by attacks [33], continuous tracking of the physical health of the grid components such as transformers [34,35] and energy usage [36] is necessary. Power grids are vulnerable to both cyber and physical attacks. An example of cyber attacks on the power grid is malicious actors attempting to penetrate and disrupt the digital systems that control the flow of electricity [37], which can be done through various means, such as phishing [38,39], malware [40,41], or denial-of-service attacks [42–44]. (A known incident was the 2015 Ukrainian power grid attack, in which hackers were able to infiltrate the digital systems of several energy companies and shut down power to 225 000 people [41,45,46].) Physical attacks on the power grid involve the destruction or damage of power lines, transformers, and substations [47,48]. (A previous incident of physical attacks was the 2013 Metcalf sniper attack, where unknown assailants damaged 17 transformers at a California substation, causing \$15 million in damage and nearly resulting in a power outage [49–51].) With the rise of digital technology and the increasing interconnectedness of the power grids, the threat of cyber and physical attacks has been increasing, making the problem of detecting and locating attacks of critical importance.

In recent years, there has been growing interest in using machine learning to improve attack detection for power grids [52,53], which enables large volumes of data from physical and digital sources to be analyzed to identify patterns and anomalies that are indicative of an attack. For example, power usage data can be analyzed to identify unusual spikes or drops in demand that may be due to a cyber attack or physical disruption, and network traffic data can be monitored to identify unusual behavior patterns, such as a sudden influx of traffic from some unexpected sources [54–56]. Machine learning can also be used to automate responses to potential attacks, such as isolating compromised systems or shutting down critical infrastructure to prevent damage. In addition, machine-learning tools can be used to predict the occurrence of faults, thereby helping utility companies to address problems such as inefficient electricity inspection and irregular power consumption [57,58]. Of particular relevance to our work is the use of RNNs and LSTMs in applications such as power demand forecasting, anomaly detection, and attack detection. For example, RNNs and LSTMs were used to analyze historical power usage data and forecast future load demand, allowing utility companies to better manage their resources and avoid blackouts [59–61]. The

LSTM-based frameworks can be used to detect anomalies in power usage data that are indicative of a fault or an attack [62–65]. For attack detection, RNNs and LSTMs were also used to analyze network traffic and system logs for signs of cyber attacks on the power grid, based on identifying patterns such as unusual data transfers or attempts to access restricted areas of the network [66].

Furthermore, previous studies exploited machine learning to detect the exact locations of attacks on power grids. For instance, convolutional neural network–based frameworks were proposed for localizing false data injection attacks [67–70]. Alternative machine-learning frameworks such as graph neural networks [71] and traditional support vector machines and random forest [72] with demonstrated performance of localizing attacks were also investigated. In these studies, full state observation was required. Our work relaxes this constraint by demonstrating that accurate attack detection and localization can be achieved even with quite limited state observation, i.e., partial state observation. This brings machine learning–based attack detection and localization a step closer to real-world implementations.

We train the LSTM networks on historical power-grid data so as to learn the underlying dynamical patterns and trends in the data. This allows us to reconstruct the full state from partial observations and identify the source of disturbances. We perform a robustness analysis by evaluating the performance under different levels of partial observations, demonstrating the ability to detect attacks accurately even when the observed data are quite limited. This is particularly important as power-grid data may be incomplete for various reasons, including technical limitations and deliberate efforts by attackers to hide their activities. Three benchmark systems of distinct scales are used in our study: one RTE 14-bus system and two IEEE 118-bus systems. The results suggest that our LSTM framework is capable of accurately detecting and locating attacks, with the potential to enhance the security and resilience of power grids against attacks.

In Sec. II, we describe in detail our LSTM method for detecting and locating disturbances. As our benchmark systems are power grids, a real power grid simulation platform is needed. We use the Grid2Op (“grid to operate”) platform, which is also described in Sec. II. Section III presents the simulation scenarios, data preprocessing, and detection results. A discussion and future perspectives are provided in Sec. IV.

II. METHODS

The working principle of our machine learning–based attack detector is outlined as follows. Consider an attack on a power grid, as indicated by the dashed black line (line 13) in Fig. 1. The goal is to ascertain the presence of an attack and identify the specific line attacked by monitoring the

capacity indicator ρ of a few randomly selected lines, e.g., lines 3, 5, 15, and 17, as indicated by the light blue–shaded line segments, where ρ is defined as the observed current flow divided by the thermal limit of the line. The input to the neural-network architecture includes current and historical information from these lines. The goal of training is for the neural network to produce the attack information. In addition to attack detection, the machine can be trained to perform full state estimation of the power-grid system based on partial observation by generating, for all the transmission lines in the power grid, the capacity indicator ρ or the power flow indicator p_{or} (the active power flow at the origin end of each power line).

To illustrate our machine-learning framework, we use three benchmark power grids: “l2rpn_case14_sandbox,” “l2rpn_wcci_2022,” and “l2rpn_idf_2023.” The first comprises 14 substations and 20 transmission lines [shown in Fig. 1(a)], while the second and third have 118 substations and 186 lines (described in Appendix A). We focus our analysis on the first in the main text, and present the results for the other two in Appendixes A, D, and E. Figure 1(a) presents a snapshot of an attack. Partial state observations consist of the monitoring of the current flows in four specific lines (lines 3, 5, 15, and 17), which are randomly selected. Figure 1(b) exemplifies an input configuration for the machine-learning framework, where each power line has a sequence length of 5, encompassing the current and historical observations. The machine-learning framework combines LSTM with fully connected neural networks, as shown in Fig. 1(c). Assume that an attack has occurred on line 13, which is not a line under observation. The task of attack detection is illustrated in Fig. 1(d). In addition, the framework can also perform full state estimation, as indicated in Fig. 1(e). It is worth noting that attack detection is essentially a classification-type task, while state estimation is a regression-type task. By modification of the activation function and the number of nodes in the last layer of the LSTM architecture as well as the use of different loss functions during training, the framework can perform both classification and regression tasks.

To provide a comprehensive picture of our articulated machine-learning framework for detecting attacks on a power grid, we present detailed descriptions of the following three components: (1) Grid2Op, a tool used for simulating realistic power-grid dynamics; (2) the proposed machine-learning architecture, and (3) the data-analysis method.

A. Power-grid simulation

Figure 1(a) displays a power grid, a complex system consisting of various interconnected components, including power transmission lines, loads (e.g., cities or industries), and generators (e.g., power plants), which are represented by lines, yellow triangles, and green

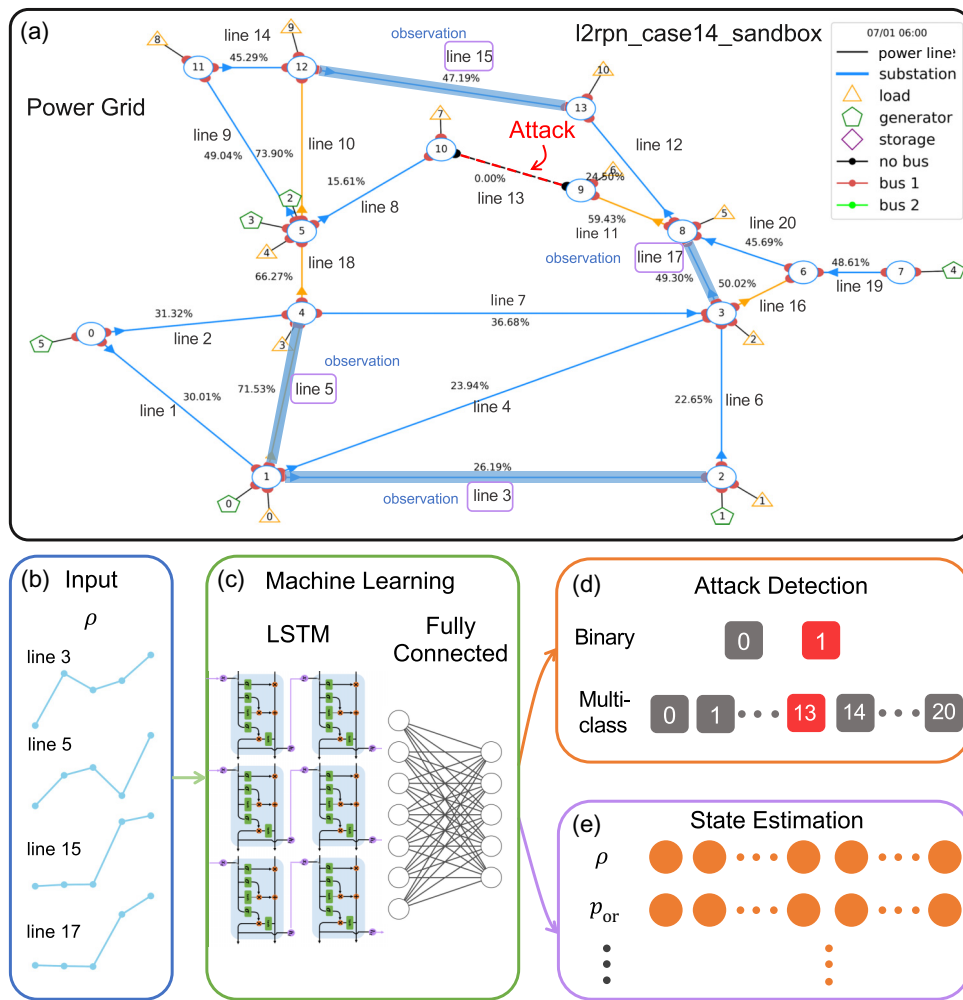


FIG. 1. Working principle of the proposed machine learning-based attack-detection and state-estimation framework. (a) A snapshot of a benchmark power grid. Partial state observations consist of monitoring the current flows in four specific transmission lines: 3, 5, 15, and 17, as indicated by the light blue-shaded line segments. An attack has occurred on line 13. The objective is to detect the attack and identify the specific line attacked on the basis solely of the partial state observations. (b) The input configuration of the machine learning-based attack detector, where the input from each transmission line under observation consists of the current capacity ρ at the current time step and the previous four time steps. (c) The structure of the machine-learning framework, which combines LSTM layers and fully connected neural networks. By adjustment of the activation function or the target variables, the framework can be adapted to the tasks of attack detection and full state estimation, as demonstrated in (d),(e), respectively. (d) An example of attack detection, where the machine-learning framework accurately detects the occurrence and location of the attack. (e) An example of state estimation, where the framework outputs the full scope of the current capacity ρ and the power flow indicator of the entire power grid.

pentagons, respectively. Substations serving as the connection points between these objects are represented by blue circles. Each substation features several switches (depicted as red balls) that enable the interconnections.

Grid2Op is a state-of-the-art platform [73] for simulating the power-grid dynamics. It is an open-source, PYTHON-based platform designed to simulate and optimize power-grid operations, and offers capabilities such as simulating attacks on a power grid, modifying generator set points, performing maintenance operations, and addressing security issues by modifying the power grid's

topology. Grid2Op is built upon an object-oriented framework, which encapsulates the underlying dynamics and constraints of the power grid within distinct modules. In particular, Grid2Op comprises four primary modules: Environment, Agent, Runner, and Backend. The Environment module encapsulates the state and dynamics of the power grid, adhering to the OpenAI Gym [74] interface to ensure compatibility with a wide range of reinforcement-learning algorithms. It provides extensive customization options, enabling key parameters such as the grid topology, load profiles, generation profiles, and

contingency scenarios to be specified. The Agent module includes the decision-making process used by the reinforcement-learning algorithm. Grid2Op offers several preimplemented agents, such as DoNothingAgent, RandomAgent, and RecoPowerlineAgent. In our work, we choose RecoPowerlineAgent, which enables power lines to be reconnected or disconnected immediately after an attack. The Runner module manages the interaction between Agent and Environment, overseeing the execution of the simulation and collecting performance metrics. Finally, the Backend module serves as the foundation for power flow computations, ensuring the accuracy and reliability of the simulation results. It also provides flexibility to integrate alternative power-system analysis frameworks.

The Grid2Op framework is derived from the “Learning to run a power network” (L2RPN) challenge—a series of competitions aimed at modeling and developing realistic power-network environments [75]. The primary objective of the L2RPN challenge is to control the power-grid network and ensure a stable electricity supply to consumers, while avoiding blackouts. In reality, blackouts caused by cascading failures of overloaded lines can result in power loss for consumers and potentially lead to secondary disasters in cities. During the L2RPN challenge, participating agents have access to complete information about the power network’s state at each step, including power line flows, electricity consumption and production at each location, power line status, and other relevant parameters.

The focus of our work is attack detection and state estimation in the realistic scenario where only partial information about the power lines’ capacity indicators is available with the system states including the electricity consumption and production associated with each power line. To achieve these goals, we simulate the Grid2Op framework as follows. Each benchmark power grid contains multiple “chronics”—time-series datasets simulating real-world power-grid conditions, which include active load consumption and generator voltage set points. The chronics make it feasible to simulate real-world power-grid flows and introduce disturbances to mimic attack scenarios. Specifically, each chronic provides data to modify the input parameters of the power flow over an extended period. The length of the time-series data differs across different power grids. In our study, the numbers of chronics for the small and two large benchmark power grids are 1004, 1662, and 832, respectively. Before the simulation, we randomly divide the thousands of chronics into three sets: training, validation, and testing, with proportions of 60%, 20%, and 20%, respectively. In the simulation, the time resolution is set to 5 min, and the simulation continues until either the chronics are run out or a game-over condition is triggered. The game-over condition is reached if the total electricity demand cannot be met, indicating that some consumption is lost at a substation. Within the simulation environment, both the opponent (RandomLineOpponent) and the agent

(RecoPowerlineAgent) operate. The opponent randomly receives a budget at each time step, which it can use to conduct attacks (disconnect power lines) for a certain number of time steps. After an attack, the opponent enters a cooldown period, during which it cannot conduct further attacks. In the meantime, the agent attempts to reconnect any disconnected power lines immediately after an attack [76]. To generate power-grid simulation data, we conduct 50 runs of each training chronics for training purposes, and 20 runs of the validation and testing chronics, respectively. We collect observations during these simulation runs to obtain comprehensive datasets for training, validating, and evaluating our proposed machine learning–based framework.

B. Machine-learning frameworks

We briefly describe the three machine-learning methods used in our study, while leaving certain details to Appendix B.

1. Long short-term memory

LSTM is a specialized type of RNN, which is effective in capturing the long-term dependencies in sequential data. Initially introduced in 1997 [11], LSTM addresses the challenges faced by traditional RNNs in capturing long-range dependencies due to issues such as vanishing or exploding gradients. Since then, LSTMs have demonstrated remarkable success in time-series forecasting, natural language processing, speech recognition, and image segmentation, among other applications [77,78]. An LSTM network consists of interconnected LSTM cells that process the sequential input data, with each cell designed to selectively retain or update information on the basis of the temporal patterns in the input, as shown in Fig. 2(a).

The LSTM component learns to represent the current and previous information, which is then passed through a decoder. In our work, the decoder is a feedforward neural network. Modification of the configuration of the last layer and selection of appropriate loss functions are necessary so that the LSTM framework is capable of dealing with different tasks.

2. Random forest

Random forest is an ensemble learning technique for classification and regression tasks with robustness, scalability, and high predictive accuracy. It was introduced as an extension of the decision-tree method to overcome the limitations associated with individual decision trees, such as overfitting and sensitivity to minor changes in training data [79]. The random forest algorithm operates by constructing multiple decision trees during the training phase and aggregating their predictions to produce the final output. As illustrated in Fig. 2(b), the input of random forest is denoted as x_i . During the training phase, a total of N

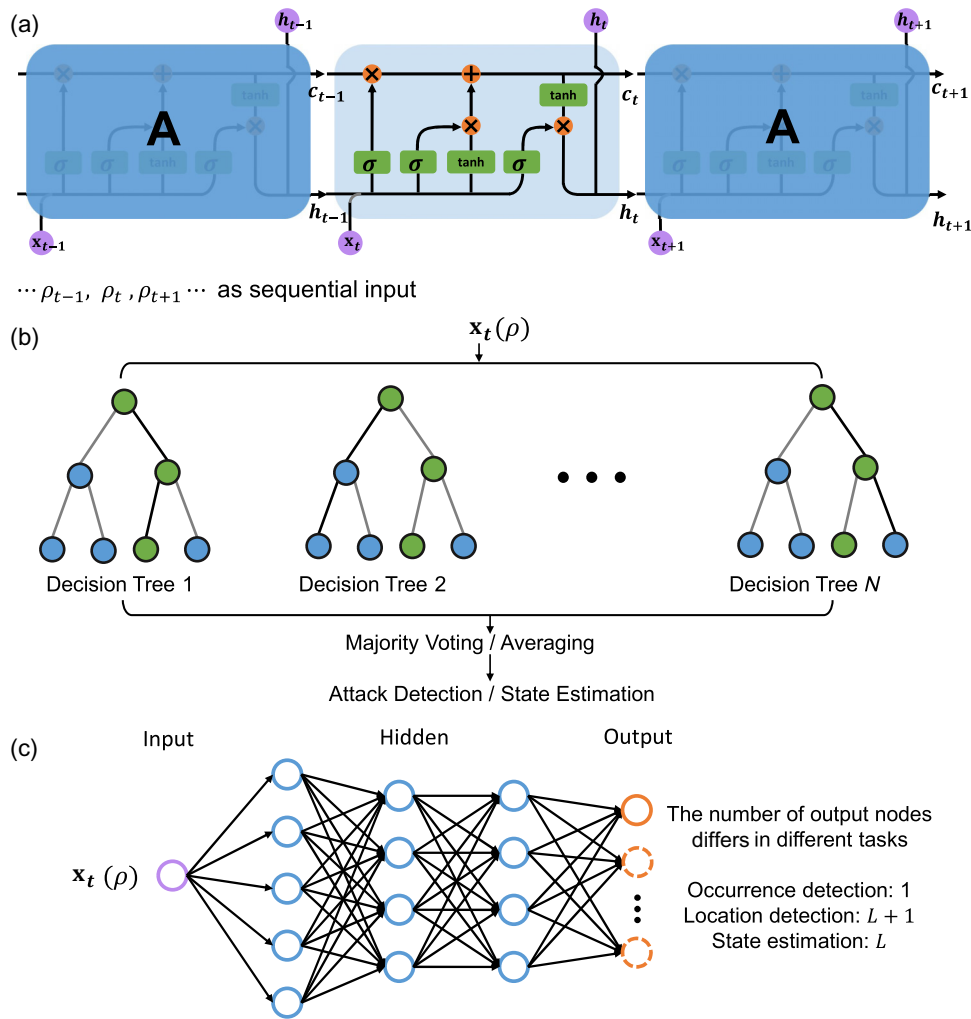


FIG. 2. Machine-learning frameworks tested in our study: (a) LSTM, (b) random forest, and (c) a feedforward neural network.

decision trees are constructed, with each tree trained on a different bootstrap sample obtained by random sampling of the training data with replacement.

While random forest offers advantages such as feature importance evaluation, noise resistance, and handling of missing data [80], it may not be suitable for all types of problems. In our work, we observe that LSTM outperforms random forest in certain scenarios, primarily due to LSTM’s ability to capture the long-range temporal dependencies in sequential data. Unlike random forest, LSTM takes into account historical information, making it more effective in modeling complex, high-dimensional, and non-linear relationships among the features. Particularly for high-dimensional datasets, random forest may suffer from overfitting or failure to converge. Overall, while random forest is a feasible machine-learning algorithm for attack detection and state estimate, its suitability depends on the specific characteristics of the problem and the nature of the data. Our results (presented below) emphasize the importance of selecting the most appropriate machine-learning

framework on the basis of the given problem’s characteristics and data properties.

3. Feedforward neural networks

FNNs are widely used for machine-learning problems such as classification and regression. An FNN consists of interconnected layers of nodes, each receiving input from the previous layer, processing the information, and passing it forward to the subsequent layer. Figure 2(c) illustrates a typical FNN architecture: input layers (purple nodes), output layers (orange nodes), and hidden layers (blue nodes) that learn to extract and represent essential features from the input data. The input to the FNN is denoted as x_t , and the number of output nodes varies depending on the specific task. The architecture in Fig. 2(c) applies to both classification and regression tasks.

We compare the performances of the LSTM and FNN frameworks. To ensure a fair comparison, we construct the FNN framework with the same number of nodes and

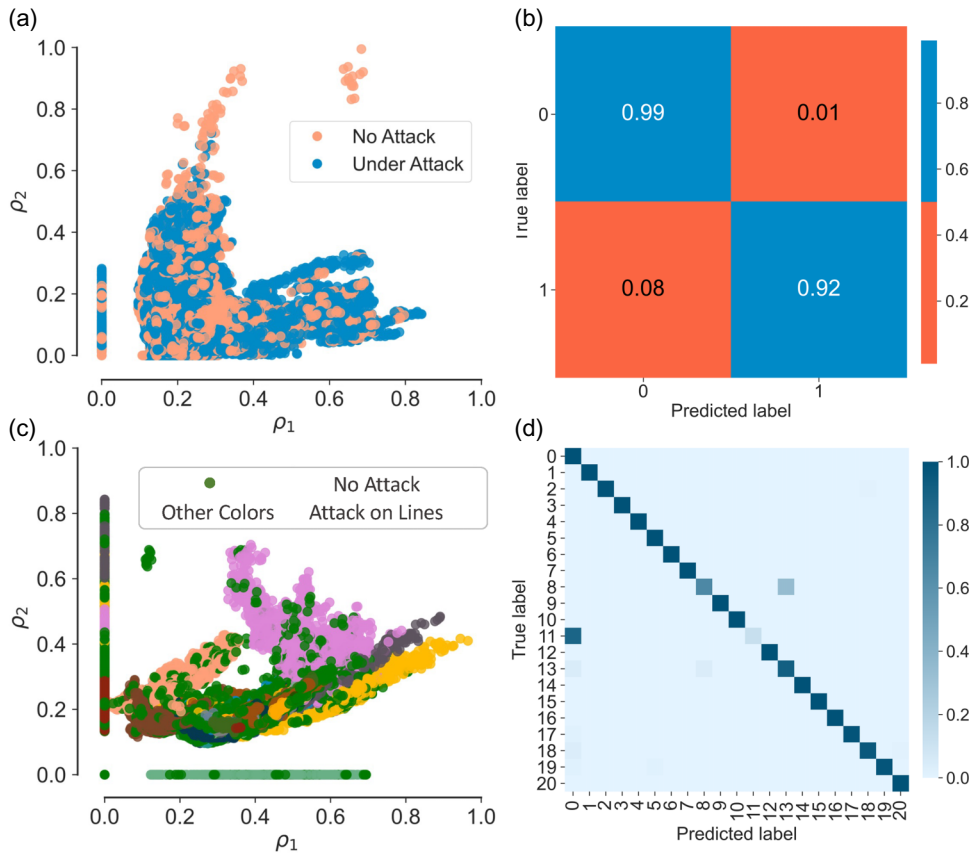


FIG. 3. Attack detection for the small power grid `l2rpn_case14_sandbox`. (a),(b) Attack occurrence detection—a binary classification problem in which the machine-learning framework determines if there is an attack on the power grid. (c),(d) Attack location detection—a multiclass classification problem in which the machine determines if an attack has occurred and, if so, identifies the specific transmission line under attack. The degree of partial state observations (the fraction of the number of transmission lines monitored), denoted as P_o , is set to 0.3, so the dimension of the input vector is $20 \times 0.3 = 6$. The relationship between the output and two input features ρ_1 and ρ_2 , among the six total inputs, is displayed in (a),(c). The machine needs to find the complex nonlinear relationship in the high-dimensional space to make satisfactory classifications. The confusion matrices in (b),(d) compare the true labels with the predicted labels obtained.

layers as in the LSTM framework by replacing several feedforward layers in the FNN with LSTM layers. Despite the similarities in the architectures and other settings, our experiments show that the LSTM framework consistently outperforms the FNN framework. This superiority can be attributed to the LSTM's ability to capture temporal dependencies in sequential data, whereas FNNs lack this capability as they do not inherently model the temporal relationships among the data points.

III. RESULTS

A. Experimental setup

Our data are from power-grid simulations on the Grid2Op platform. In particular, to obtain sufficient datasets for training, validation, and testing, we divide the thousands of chronics into the respective datasets and restart the simulations multiple times, with each simulation randomly selecting a chronic. The opponent in the

simulation is `RandomLineOpponent`, which attempts to disconnect power lines using its allocated budgets. The opponent's initial budget is set to 0 and increases by 0.8 for each step. A higher budget makes the opponent more likely to attack. In each simulation, the attack duration and the cooldown time, which represent the duration of each attack and the minimum time between two attacks, respectively, are generated independently and uniformly within the range $[1, 6]$. We assume that all power lines in the grid are susceptible to attacks and, after an attack, the power lines are reconnected immediately by `RecoPowerlineAgent`.

We conduct the simulation of the machine-learning frameworks on three computers equipped with a GeForce RTX 4090 GPU and a 13th generation Intel Core i9-13900KS CPU using PYTHON. The simulation parameters and data preprocessing are as follows. Within a "length" (the total length of temporal evolution), attacks may or may not occur. For the benchmark grid `l2rpn_case14_sandbox`, the total lengths of the training,

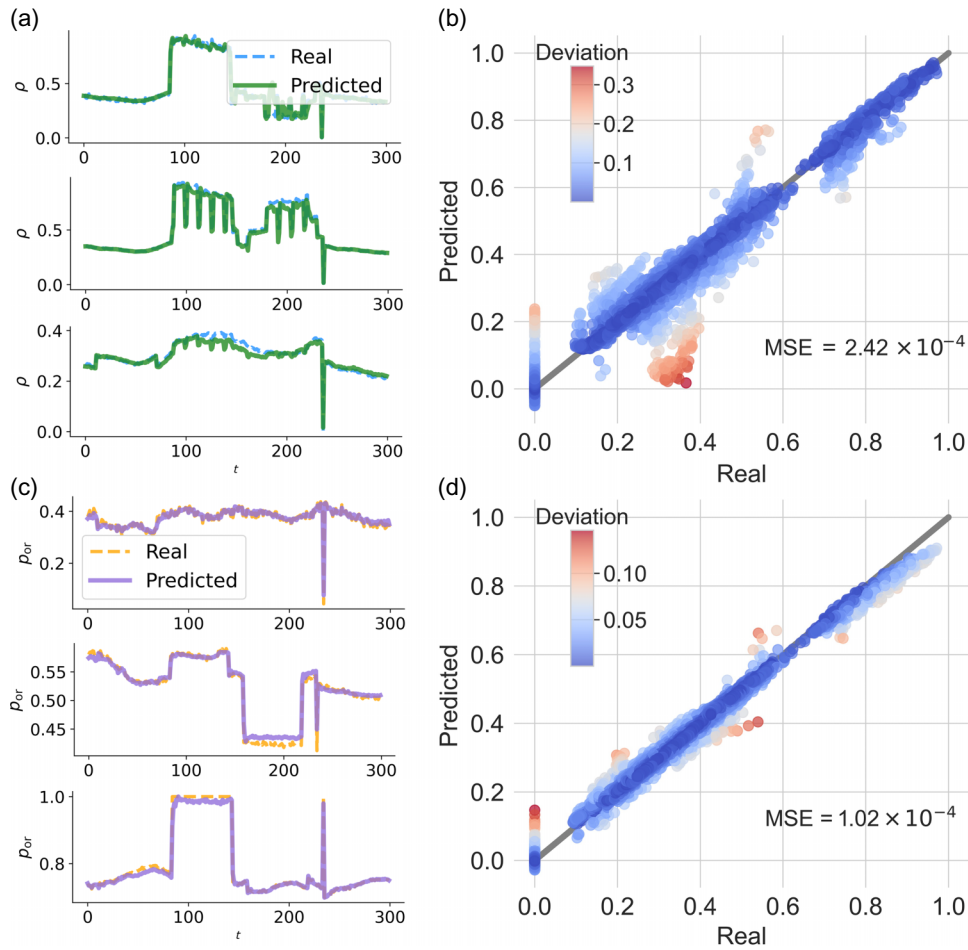


FIG. 4. State estimation for the small power grid l2rpn_case14_sandbox. (a),(b) Estimation of ρ : the machine-learning method predicts the ρ values of all the power lines on the basis of partial observation of ρ . (c),(d) State estimation for p_{or} , where the machine predicts the p_{or} values of all power lines on the basis of partial observations of ρ . In (a),(c) segment examples of the real and predicted values are presented, while in (b),(d) regression results by comparison of the true and predicted values of a specific transmission line are shown. The degree of partial state observations $P_o = 0.3$. The dimension of the input vector is $20 \times 0.3 = 6$.

validation, and testing datasets are 556 208, 76 167, and 78 251 time steps, respectively. Within time steps 311 157, 38 089, and 38 801 for training, validation, and testing, respectively, the power grid is under attack. For l2rpn_wcci_2022, the total lengths of the respective datasets are 588 472, 97 201, and 99 660 for training, validation, and testing, respectively. Within time steps 323 823, 51 745, and 57 366, the power grid is under attack. For l2rpn_idf_2023, the total lengths of the respective datasets are 686 530, 145 684, and 138 621 for training, validation, and testing, respectively. Within time steps 386 568, 84 003, and 86 082, the power grid is under attack. Each dataset encompasses all the necessary information, such as whether an attack has occurred, which line was attacked, and the capacity of each power line. We preprocess the time series using min-max normalization [81] to ensure that their amplitude falls in the unit interval. Specifically, for the capacity time series x_ρ of a power line in the

training phase, the data are normalized as

$$x'_\rho = (x_\rho - \min(x_\rho)) / (\max(x_\rho) - \min(x_\rho)),$$

providing consistent scaling of all the data.

To demonstrate the performance of the machine-learning frameworks, we use two widely used evaluation metrics—the F_1 score and the mean squared error (MSE, denoted as E)—to characterize the performance of attack detection and state estimation, respectively. The F_1 score is commonly used for classification tasks, particularly for imbalanced datasets. It combines precision (P) and recall (R) into a single measure, providing a balanced assessment of the performance of the machine-learning framework. Precision is the fraction of true positive predictions among all positive predictions, while recall is the fraction of true positive predictions among all actual positive instances. Mathematically, precision and recall are defined

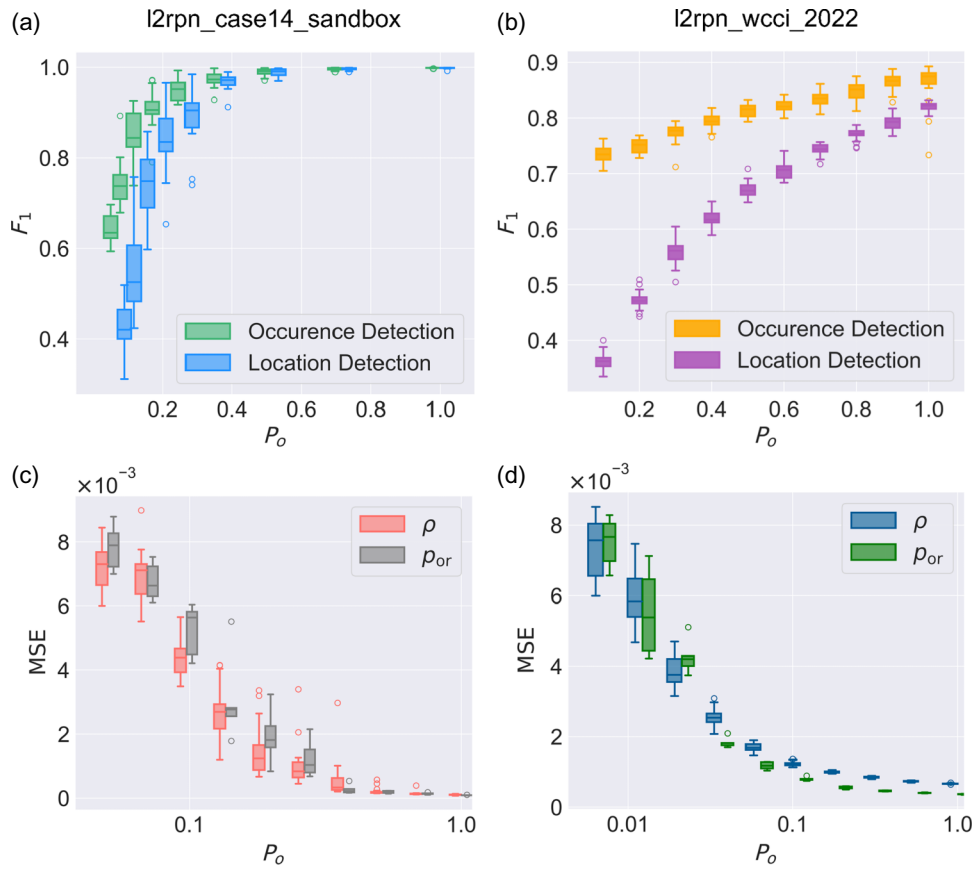


FIG. 5. Performance of the LSTM-based framework under differing extent of partial state observations. (a),(b) Results from attack-detection tasks in the small and large power grids, respectively, with the F_1 score as the evaluation metric. (c),(d) Results from state-estimation tasks in the small and large power grids, respectively, with the MSE as the evaluation metric. The box plots are obtained from 20 experiments conducted for each value of the observation extent P_o . Both sets of results indicate that as P_o increases, the performance of the LSTM-based framework improves.

as follows:

$$P = \frac{T_P}{T_P + F_P}, \quad (1)$$

$$R = \frac{T_P}{T_P + F_N}, \quad (2)$$

where T_P represents the number of true positives, an indication that the framework correctly predicts the positive labels, F_P represents the number of false positives, meaning that the framework incorrectly predicts the positive labels, and F_N represents the number of false negatives, indicating that the framework incorrectly predicts the negative labels. The F_1 score is the harmonic mean of precision and recall as

$$F_1 = 2 \times \frac{P \times R}{P + R}, \quad (3)$$

whose values lie in the unit interval, with a higher value indicating better classifier performance. For regression

tasks, the MSE measures the average squared difference between the predicted and true values, which is indicative of the framework's accuracy in predicting continuous variables. Specifically, the MSE (E) is calculated as follows:

$$E = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (4)$$

where y_i and \hat{y}_i represent the true and predicted values, respectively, and n is the number of data points. A lower MSE value signifies better performance.

To evaluate the robustness of the machine-learning frameworks regarding random disturbances, we introduce Gaussian white noise to the input data during the training phase. The measurement noise is added as follows:

$$\tilde{x}_i = x_i + \xi_n, \quad (5)$$

where the stochastic process ξ_n follows a normal distribution with zero mean and standard deviation σ_n . Unless otherwise specified, we set $\sigma_n = 0.02$.

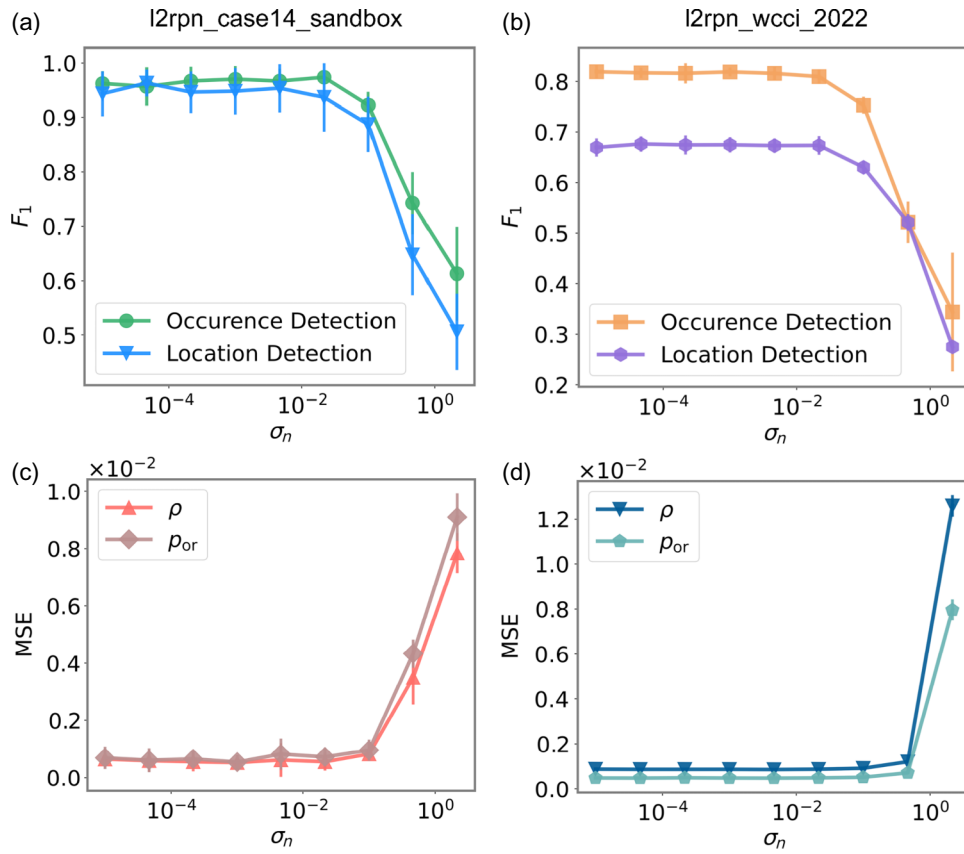


FIG. 6. Robustness of the LSTM-based framework regarding noise. (a),(b) Results from attack-detection tasks for the small and large power grids, respectively, with the F_1 score as the evaluation metric. (c),(d) Results from state-estimation tasks for the small and large power grids, respectively, with the MSE as evaluation metric. Each error bar represents the variability observed across 20 experiments. As the noise level σ_n increases, the performance of the LSTM framework initially remains relatively stable and then starts to decline.

Our proposed framework combines an LSTM component, which captures the temporal features and dependencies from input sequences, with an FNN decoder that maps these features to the desired outputs, for attack detection and state estimation. In particular, the first task aims to detect the occurrence of an attack and identify the specific transmission line targeted. The output layer is adjusted accordingly, with one node and a “sigmoid” activation function to ascertain whether an attack has occurred (binary classification), and $L + 1$ nodes and a “softmax” activation function to locate the attack (multiclass classification), where L is the number of power lines of the grid. For the binary and multiclass classification tasks, the training loss functions are binary cross-entropy and categorical cross-entropy [82], respectively. For the state estimation task (a regression-type task), the objective is to reconstruct the complete power-grid state on the basis of partial observations, where the output layer consists of L nodes with a “linear” activation function and the training loss function is the mean squared error. For both attack detection and state estimation, historical information from the measurements is important. By our incorporating historical data

into our LSTM framework, it can capture the temporal dependencies and increase the accuracy for both tasks.

B. Demonstration of attack detection and state estimation

Our machine-learning framework has two LSTM layers followed by two feedforward layers. To prevent overfitting and increase generalizability, the operation of Dropout is applied to all the layers of the network with the dropout rate 0.2. The number of nodes in the LSTM layers is set to 128 and 64, respectively. The number of nodes in the feedforward layers varies depending on the specific task: 16 and 1 for attack occurrence detection, 64 and $L + 1$ for attack location detection, and 64 and L for state estimation. To meet the requirements of the LSTM framework, we reorganize the input data. In particular, we use a sequence length of 5 to determine the number of consecutive time steps included in the input data. This ensures that the framework captures the temporal dependencies within a specified time window. More specifically, at each time step t , the input data consist of the current measurement \mathbf{x}_t and

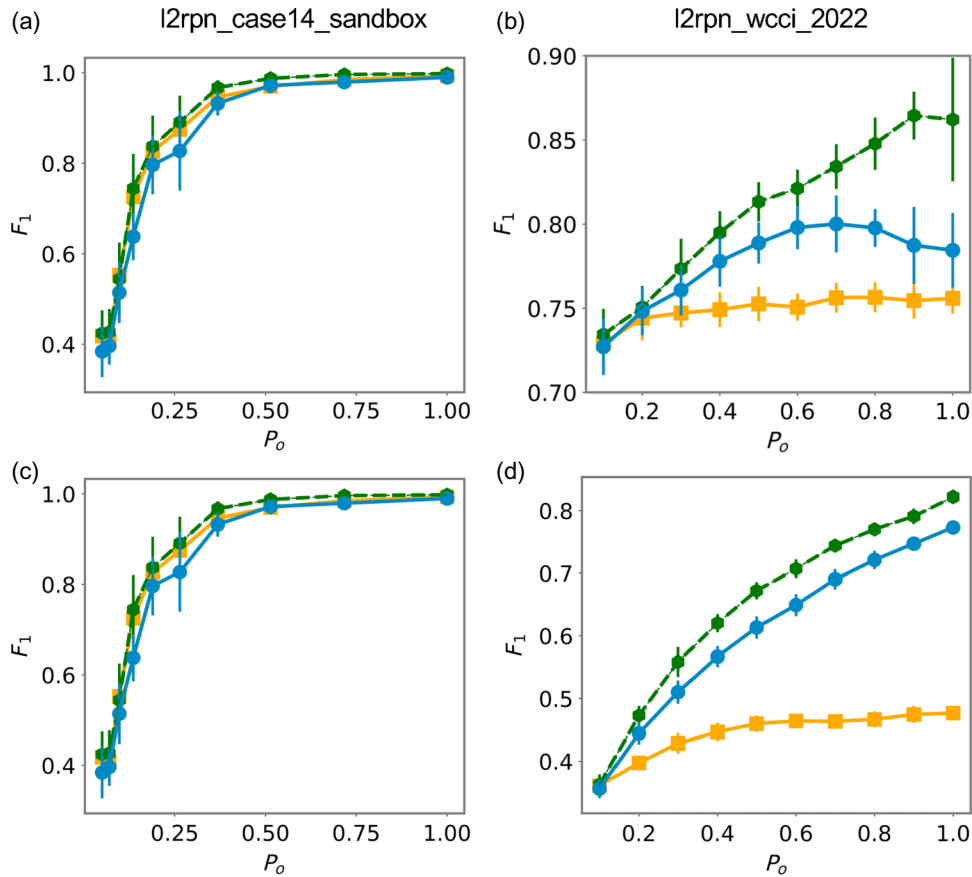


FIG. 7. Performance comparison of attack-detection tasks among three machine-learning frameworks. Results for the small [(a),(c)] and large [(b),(d)] power grids for (a),(b) attack occurrence detection and (c),(d) attack location detection. Yellow, green and blue lines denote the performance of the random forest, LSTM, and FNN frameworks, respectively. Each error bar is calculated from 20 experiments. The superiority of the LSTM-based framework becomes more pronounced for the large power grid.

the measurements at the previous $s - 1$ time steps, denoted as $\{\mathbf{x}_i\}_{i=t-s}^t$. The output corresponds to the target value at time t . This process is repeated for the entire dataset, resulting in a reorganized data structure with input-output pairs suitable for training the LSTM-based framework. The reorganized dataset can be represented as “[samples, sequence length, features],” where “samples” is referred to as the length of the dataset.

We first present representative results of attack detection for $l2rpn_case14_sandbox$ —a relatively small power grid. The extent of partial state observations P_o denotes the fraction of the number of transmission lines monitored among all the lines in the power grid, which is set to 0.3. That is, of the 20 transmission lines in this power network, six lines are monitored continuously in time, generating the input data for the machine-learning framework. For attack occurrence detection, Fig. 3(a) shows the relationship between the input features and the attack occurrence. Ascertaining the attack occurrence on the basis of partial state observations of one or two lines, e.g., ρ_1 and ρ_2 , is difficult. When six lines are observed, the machine-learning

framework achieves high accuracy, as can be seen from the confusion matrix in Fig. 3(b) demonstrating that when there is no attack (label 0), the framework predicts it correctly with probability 0.99. When there is an attack, the framework predicts it accurately with probability 0.92. For attack location detection, overall the framework performs well, correctly classifying most of the labels. However, there are a few instances where the framework fails to give the correct location of the attack. For example, when line 11 is under attack, the framework gives that this line is not under any attack. Figure 3(c) shows the attack on different lines (via different colors), and the corresponding confusion matrix is shown in Fig. 3(d).

We next present results from the regression problem for state estimation. Figure 4 presents an example of state estimation for the $l2rpn_case14_sandbox$ power grid. For $P_o = 0.3$, the framework predicts the values of ρ and p_{or} of all the transmission lines. Figures 4(a) and 4(c) show three examples of the true and predicted values for ρ and p_{or} , respectively, while Figs. 4(b) and 4(d) compare the predicted and true values for a specific line that is not under

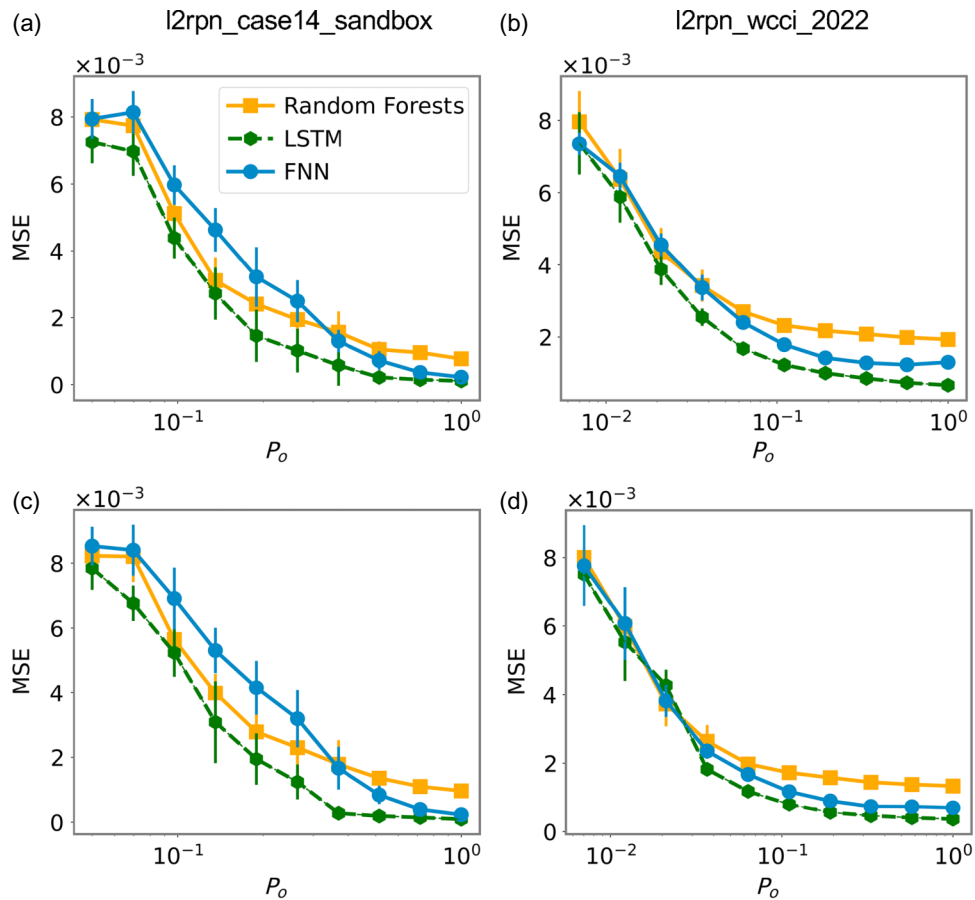


FIG. 8. Performance comparison of state estimation tasks among three machine-learning frameworks. Results for the small [(a),(c)] and large [(b),(d)] power grids for (a),(b) state estimation for ρ and (c),(d) state estimation for p_{or} . Each error bar is from 20 experiments. Overall, the LSTM-based method outperforms the FNN and random forest frameworks.

observation, where the deviations are illustrated with different colors (red indicating larger deviations). The overall MSE for this line is shown in the lower-right corner in Figs. 4(b) and 4(d). Overall, the results demonstrate that the machine-learning method is capable of accurate full state estimation when 30% or more of the transmission lines are under observation.

For the small power grid `l2rpn_case14_sandbox`, additional state-estimation results can be found in Appendix C. For the two large power grids `l2rpn_wcci_2022` and `l2rpn_idf_2023`, the attack-detection and state-estimation results are presented in Appendixes D and E, respectively. Performances similar to those in Figs. 3 and 4 are achieved. The performance of the LSTM under different class weights is demonstrated in Appendix F.

C. Effect of partial state observations

The extent of partial state observations, denoted as P_o , characterizes the scope of the information input to the machine-learning framework. A small value of P_o corresponds to a low-dimensional input where the framework can access only quite limited information about the

power grid, while a high value of P_o indicates that more comprehensive information about the power grid is input to the machine-learning framework at each time step. The three power grid networks `l2rpn_case14_sandbox`, `l2rpn_wcci_2022`, and `l2rpn_idf_2023` have 20, 186, and 186 power lines, respectively. If the observation is from a single transmission line, the values of P_o for the three power grids are 0.05, 0.007, and 0.007, respectively.

To investigate the impact of the P_o value on machine-learning performance, we conducted a systematic analysis by varying this parameter across its entire range and evaluating the performance for each configuration. It is useful to note that the model needs to be retrained for different configurations of partial observations P_o . To reduce the computational complexity and the inherent fluctuations in the training process, we use half of the available training, validation, and testing data segments. To obtain reliable results, we train the framework 20 times for each value of P_o and analyze the results to assess how the value of P_o influences the overall performance of the framework. Figure 5 shows the results from the attack-detection and state-estimation tasks, where Figs. 5(a) and 5(c) are for

l2rpn_case14_sandbox, and Figs. 5(b) and 5(d) are for l2rpn_wcci_2022. The results are visualized by box plots, which provide a compact and intuitive representation of the data distribution. In a box plot, the rectangular box represents the interquartile range, which contains the central 50% of the data, with the lower and upper quartiles forming the boundaries of the box. The median value is indicated by a horizontal line inside the box. Whiskers extend from the box to the minimum and maximum data points within 1.5 times the interquartile range. Data points outside this range are considered outliers and are plotted individually.

Figure 5(a) presents the performance for the attack occurrence and attack location detection tasks on the small power grid. For both tasks, when P_o is larger than approximately 0.2, reasonable performance can be achieved. For $P_o > 0.4$, the classification scores approach 1.0. The corresponding results for the large power grid are shown in Fig. 5(b). Due to the significantly high complexity of this power grid as compared with the small one, achieving acceptable performance requires the observation of more than half of the transmission lines. Even for $P_o = 1.0$ so that the capacities ρ of all power lines are observed, the classification results are still not significantly better. The reason is attributed to the assumption in our work that noise is always present. Specifically, in a large network, certain power lines may be in a low-burden or a high-burden state even under normal operation, which can resemble scenarios of complete disconnection or an attack. When noise is added to the observations, the attack and normal cases may become mixed, leading to a degradation in the machine-learning performance. Our experiment with $P_o = 1$ in a noise-free scenario shows that the F_1 score exceeds 0.99. The corresponding results for state estimation are presented in Figs. 5(c) and 5(d) for the small and large power grids, respectively. As P_o increases, the MSE decreases rapidly, so we use a semilogarithmic scale to illustrate the performance. The results for both power grids exhibit a similar trend: as P_o increases, the performance in estimating the full scope of ρ and p_{or} improves. However, for the large power grid, achieving good performance requires relatively smaller values of P_o , as more power lines are under observation with the same value of P_o .

D. Robustness regarding noise

Evaluation of the robustness of machine-learning frameworks regarding noise [25] is necessary for applications. We add Gaussian noise of varied amplitude to the normalized training data. The simulation results are presented in Fig. 6, where Figs. 6(a) and 6(c) and Figs. 6(b) and 6(d) show the performance for the small and large power grids, respectively, for four tasks: attack occurrence detection, attack location detection, ρ state estimation, and p_{or} state

estimation. Overall, the results indicate that the performance of the LSTM framework remains relatively stable as the noise level σ_n increases initially, suggesting that the framework is robust regarding moderate noise levels. However, as the noise level increases further, the performance deteriorates rapidly. Such behavior appears to be common in applications of machine learning in nonlinear and complex dynamical systems [25]. The robustness of random forest and the FNN regarding noise is also tested, and they show similar performance to the LSTM-based framework, i.e., the models remain robust under small amounts of noise but as the noise level becomes relatively large, the performance decreases.

E. Comparison among different machine-learning methods

To justify our choice of the LSTM framework for attack detection and state estimation, we compare the performances of three machine-learning methods: LSTM, random forest, and an FNN.

Figure 7 presents the results of performance comparison for the attack-detection tasks, where Figs. 7(a) and 7(c) and Figs. 7(b) and 7(d) are for the small and large power grids, respectively, and Figs. 7(a) and 7(b) and Figs. 7(c) and 7(d) display the results for attack occurrence detection and attack location detection, respectively, based on partial observations of ρ , where the error bars are calculated from 20 independent simulation runs. While Figs. 7(a) and 7(c) show that the three machine-learning frameworks all exhibit reasonably good performance, with no apparent significant differences, Figs. 7(b) and 7(d) demonstrate the advantage and superiority of the LSTM-based framework over the FNN and random forest for the large power grid. Figure 8 shows a performance comparison for the state-estimation tasks, where Figs. 8(a) and 8(c) and Figs. 8(b) and 8(d) are for the small and large power grids, respectively, and Figs. 8(a) and 8(b) and Figs. 8(c) and 8(d) are for estimating ρ and p_{or} , respectively, for all the transmission lines. A behavior similar to that of the attack-detection tasks emerges: the LSTM-based framework consistently outperforms the FNN and random forest. These results of performance comparison thus highlight the inherent advantage of LSTM in capturing temporal dependencies and handling complex feature relationships required for attack detection and state estimation of complex dynamical systems such as power grids.

IV. DISCUSSION

In complex cyber-physical networked systems of interconnected components, a challenging problem is to detect and locate an attack on some component on the basis of observing the dynamical behaviors of some other components in the system that may not be adjacent to the component under attack. This is the problem of attack

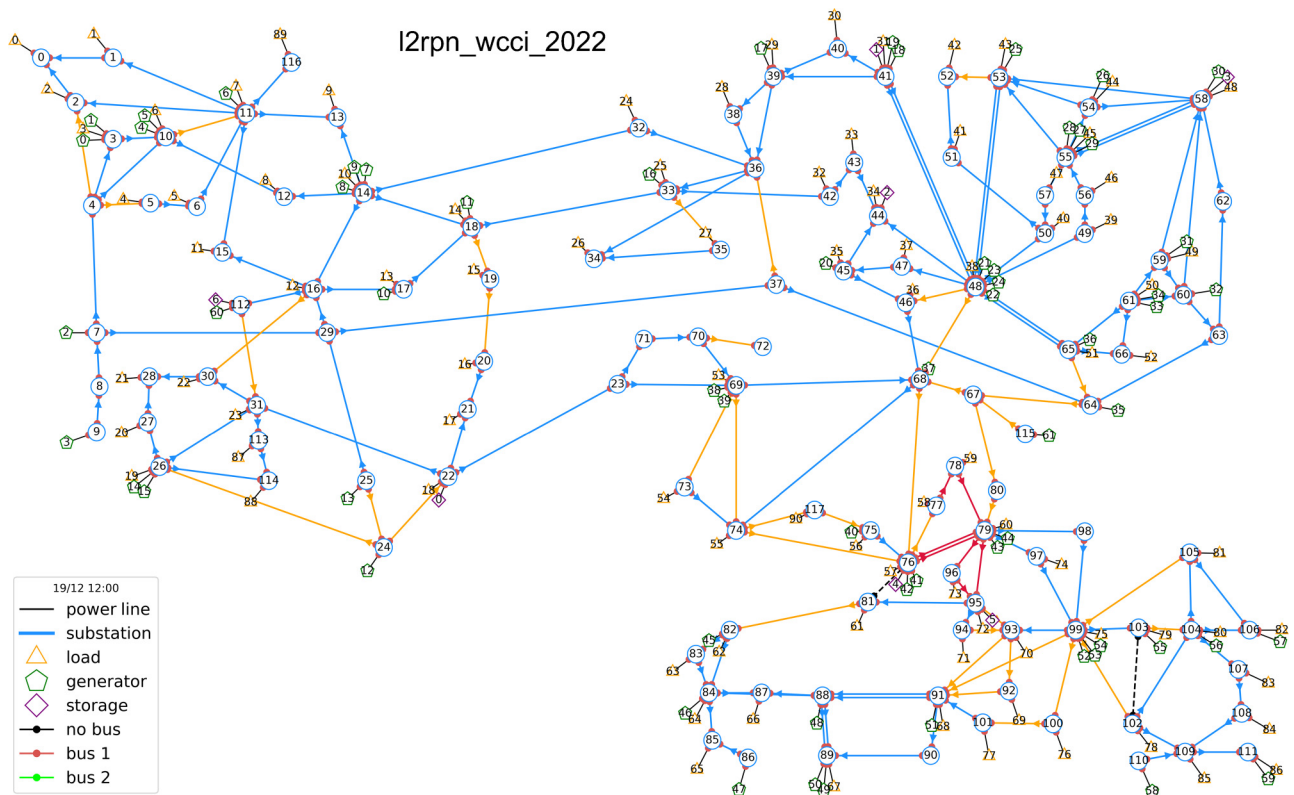


FIG. 9. A snapshot of the large power grid *l2rpn_wcci_2022* used in our study. The other large power grid, *l2rpn_idf_2023*, shares the same structure but differs in the number of loads and chronics.

detection based on partial state observations. A related problem is to estimate the state of the whole system on the basis of the partial observations. The two problems are extremely challenging, but modern machine learning can be exploited to provide a solution. The main result of this work is a demonstration that this is indeed the case. In particular, by using the machine-learning framework LSTM, we have demonstrated its capability of reconstructing the state and attack information for complex power grids using only observation of the current flows through a limited subset of all the transmission lines. The justification for choosing LSTM lies in its superior capability to capture the long-term dependencies in the network data, which are essential for learning the dynamical patterns of the system in the absence of any attack and distinguishing it from those when an attack has occurred. Simulation results on the effects of the extent of partial state observations, robustness regarding noise, and performance comparison with two alternative machine-learning frameworks reinforce the choice of LSTM. Taken together, our results highlight the inherent strengths of the LSTM-based framework in capturing temporal dependencies and handling complex feature relationships, which are essential for attack detection and state estimation. A related recent work is reconstructing complex networks from partial nodal state observations [10], where link existence is inferred from the data. Our

work is different in that we focused on using partial link states, i.e., the currents in a small number of random transmission lines in a power grid, to determine whether the network is under attack and to identify its location. In our work, partial observations were also used to reconstruct the network dynamics, i.e., to estimate the full scope of the link states and other critical indicators for the entire power grid.

Possibilities for future research are as follows. First, the LSTM framework can be extended to complex cyber-physical systems beyond power grids. For example, it can be applied in the contexts of synchronization [83], information spreading [84], and symmetry detection [85] in complex networks. Next, while the LSTM-based framework outperforms the FNN and random forest in detecting

TABLE I. Parameters of the power grids *l2rpn_wcci_2022* and *l2rpn_idf_2023* for comparison.

Property	<i>l2rpn_wcci_2022</i>	<i>l2rpn_idf_2023</i>
Substations	118	118
Power lines	186	186
Loads	91	99
Generators	62	62
Data length	32 years	16 years

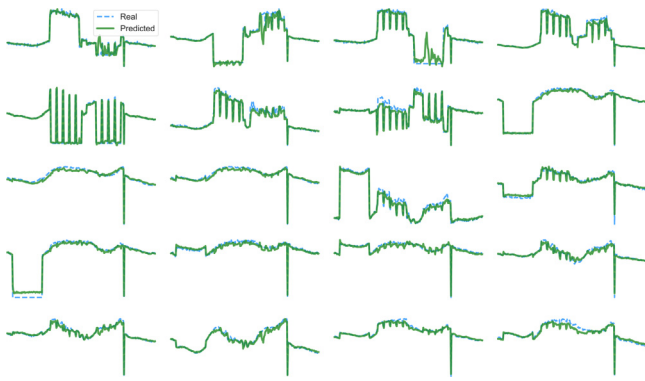


FIG. 10. Additional examples of estimating all the state variables ρ for the small power grid `l2rpn_case14_sandbox`. The partial state observation parameter $P_o = 0.3$.

attacks, future research could explore alternative machine-learning frameworks such as graph neural networks [86], transformers [87], and diffusion models [88] for multitask learning in a single neural-network architecture. Moreover, the attack-detection and state-estimation results reported here can be used to develop reinforcement-learning control to enable real-time protection of cyber-physical systems [89,90]. Finally, for large and complex cyber-physical systems, the LSTM-based framework can be ineffective when the state observations are severely limited, especially for the task of locating an attack. For systems that are more complex than the three power-grid networks studied here, additional input features and more observations are required to achieve reliable performance.

ACKNOWLEDGMENTS

We thank Dr. Ling-Wei Kong for discussions. This work was supported by AFOSR under Grant No. FA9550-21-1-0438 and also by the U.S.-Israel Energy Center managed by the Israel-U.S. Binational Industrial Research and Development Foundation.

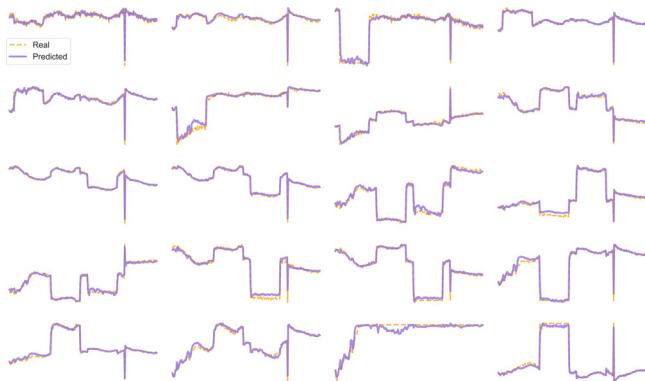


FIG. 11. Additional examples of estimating all the state variables p_{or} for the small power grid `l2rpn_case14_sandbox`. The partial state observations are from ρ with $P_o = 0.3$.

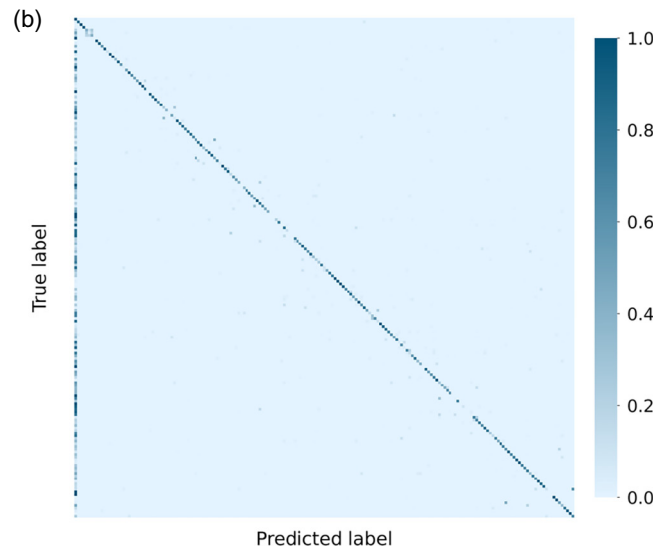
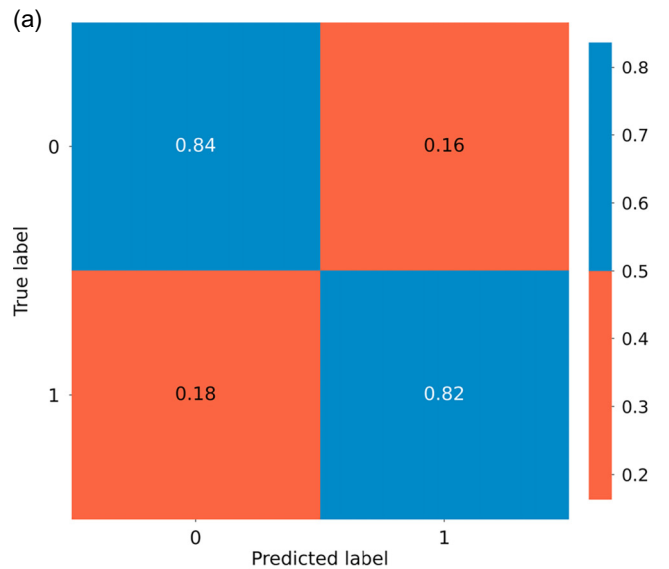


FIG. 12. Attack-detection performance for the large power grid `l2rpn_wcci_2022`. Confusion matrices for (a) attack occurrence detection and (b) attack location detection are shown. The extent of partial state observations $P_o = 0.5$ and the dimension of the input vector is $186 \times 0.5 = 93$.

DATA AVAILABILITY

The simulation data for the three power grid benchmarks can be found in the Zenodo repository [91]. The data that support the findings of this article are openly available [91,92], embargo periods may apply.

The codes for generating all the results can be found on GitHub [92].

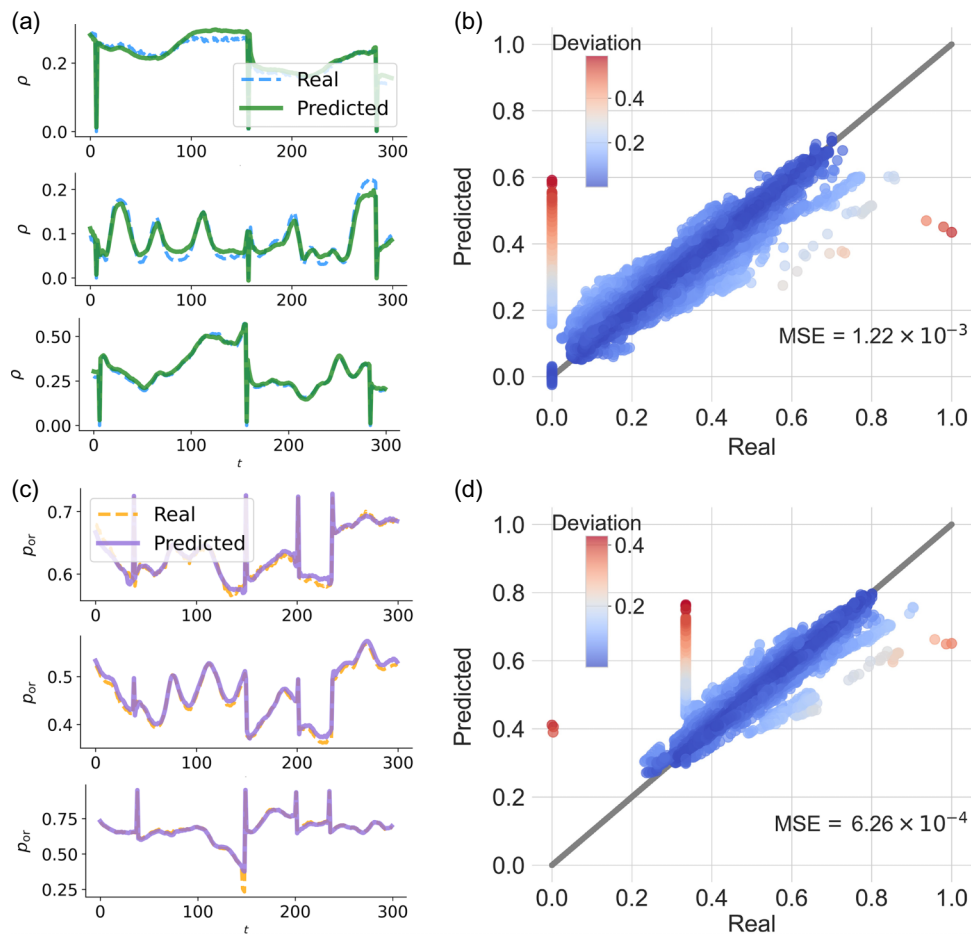


FIG. 13. State-estimation performance for the power grid l2rpn_wcci_2022. (a),(b) Examples of state estimation in which the LSTM machine aims to predict ρ of all the transmission lines from partial state observations. (c),(d) Examples of state estimation of p_{or} for all the lines. In (a),(c), several segment examples are shown that compare the true and predicted values of ρ and p_{or} , respectively. In (b),(d), the regression results by comparison of the true and predicted values of an example line are shown. The extent of partial state observations $P_o = 0.3$ and the dimension of the input vector is $186 \times 0.3 = 56$.

APPENDIX A: DESCRIPTION OF THE LARGE POWER GRID AND AN ATTACK SCENARIO

Three power-grid networks of different sizes—l2rpn_case14_sandbox, l2rpn_wcci_2022, and l2rpn_idf_2023—were used to test the capabilities of three machine-learning frameworks for attack detection and state estimation. The simulation results from the small power grid l2rpn_case14_sandbox with 14 substations and 20 power lines are described in the main text. The results from the large power grids l2rpn_wcci_2022 and l2rpn_idf_2023 are presented here.

Both power grids l2rpn_wcci_2022 and l2rpn_idf_2023 comprise 118 substations and 186 power lines, as shown in Fig. 9, where the line numbers and power line capacities ρ for each power line cannot be visualized from Fig. 9 due to the large size. The difference lies in the number of loads and the chronics, as listed in Table I. Similarly to the case of the small power grid, the blue power lines

between substations are healthy, while the orange power lines indicate relatively high current flows with a potential risk of overload. A red power line means that the current flow it carries is overloaded. Without appropriate control measures, a red line may lead to a blackout of the power grid. Additionally, a dotted power line represents one under an attack, which may last for a certain period or result in physical disconnection if it becomes excessively overloaded. As shown in Fig. 9, the power line between substation 102 and substation 103 is under attack. Before this time step, the power grid network operated normally for approximately 2 h, even when under attack. However, at the current time step, several power lines in the grid are overloaded, as indicated by the red power lines in the lower-right corner in Fig. 9. Notably, due to excessive overload, the power line between substation 76 and substation 81 is physically disconnected (not under attack). Consequently, a blackout is imminent in the power grid within 15 min, at which point the entire power grid will

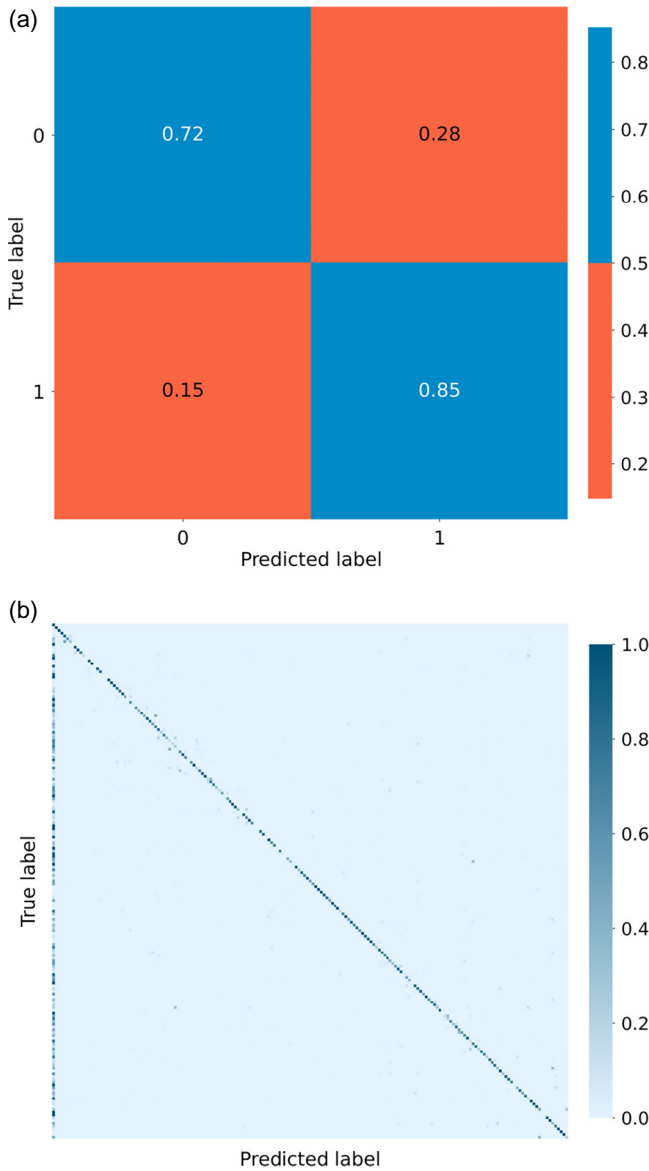


FIG. 14. Attack-detection performance for the large power grid l2rpn_idf_2023. Confusion matrices for (a) attack occurrence detection and (b) attack location detection are shown. The extent of partial state observations $P_o = 0.5$ and the dimension of the input vector is $186 \times 0.5 = 93$.

cease to function and the simulation terminates. Note that we have deliberately selected a scenario where a blackout occurs within 15 min (three time steps in the simulation) to illustrate the state of the grid before a blackout after a prolonged attack.

APPENDIX B: DETAILS OF THE MACHINE-LEARNING METHODS USED IN THIS STUDY

1. LSTM

The core components of an LSTM cell include the input gate, forget gate, and output gate, which collaboratively

regulate the flow of information throughout the network. These gates act as neural-network layers with sigmoid activation functions, $\sigma(x) = 1/(1 + e^{-x})$, generating values between 0 and 1 that determine the extent to which information is discarded or preserved.

The input gate governs the integration of new input data into the cell state, while the forget gate controls the retention of the existing cell state. At each time step t , these decisions are made with the use of separate sigmoid functions:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f), \quad (\text{B1})$$

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i), \quad (\text{B2})$$

where σ denotes the sigmoid activation function. The weight matrices and bias terms for the input and forget gates are represented by W_i , b_i , W_f , and b_f , respectively. Additionally, a hyperbolic tangent-activated layer is used to obtain the candidate cell state:

$$\tilde{C}_t = \tanh(W_C \times [h_{t-1}, x_t] + b_C), \quad (\text{B3})$$

with W_C and b_C denoting the weight matrix and the bias term, respectively. The cell state is then updated according to

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \quad (\text{B4})$$

where \odot represents elementwise multiplication. The cell determines the output, which is a filtered version of the cell state, through the output gate:

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o), \quad (\text{B5})$$

with W_o and b_o representing the weight matrix and the bias term of the output gate, respectively. Subsequently, the cell state is processed through a hyperbolic tangent layer and multiplied by the output gate o_t to update the hidden state h_t :

$$h_t = o_t \odot \tanh(C_t). \quad (\text{B6})$$

2. Random forest

In a classification task, the final prediction of random forest is determined through majority voting among the individual trees, while in a regression task, the average of the individual tree predictions is used. This ensemble approach enables random forest to capture complex feature interactions, reduce overfitting, and improve generalization to unseen data. Further, the technique known as “bagging” or “bootstrap aggregating” used in random forest introduces variation among the trees and reduces the risk of overfitting. Additionally, at each decision-tree split,

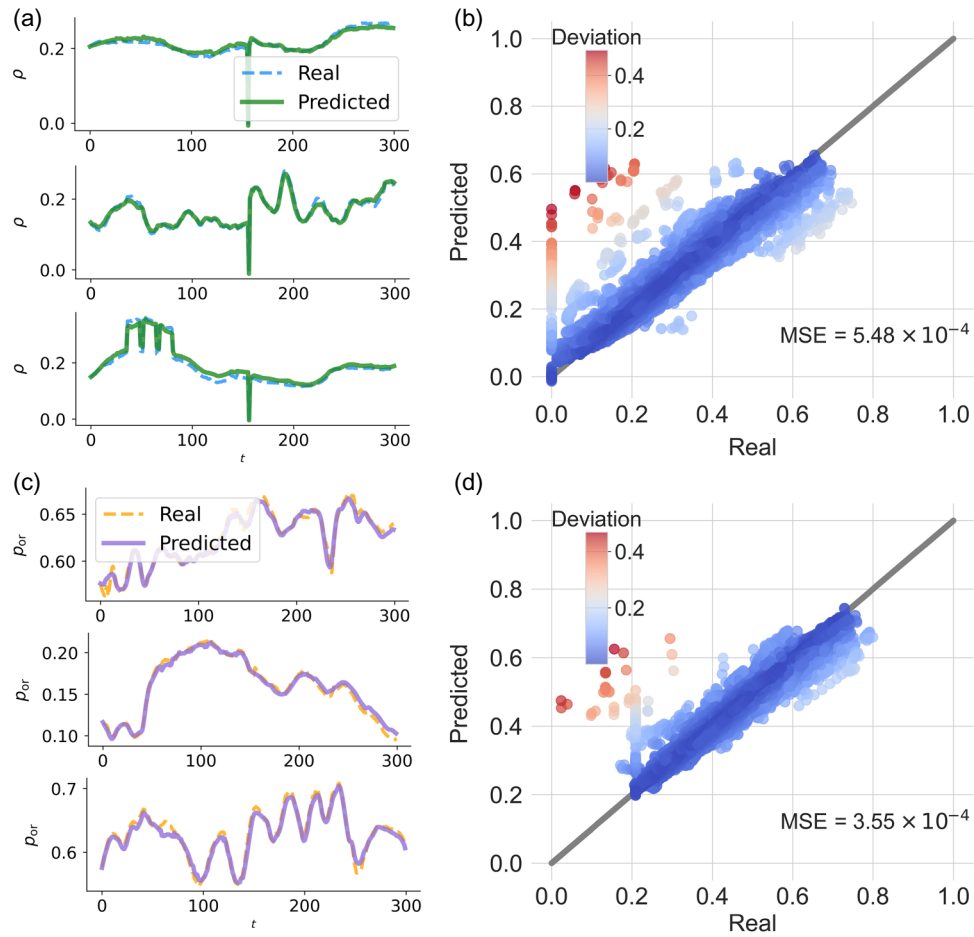


FIG. 15. State-estimation performance for the large power grid l2rpn_idf_2023. (a),(b) Examples of state estimation in which the LSTM machine aims to predict ρ of all the transmission lines from partial state observations. (c),(d) Examples of state estimation of p_{or} for all the lines. In (a),(c), several segment examples are shown that compare the true and predicted values of ρ and p_{or} , respectively. In (b),(d), the regression results by comparison of the true and predicted values of an example line are shown. The extent of partial state observations $P_o = 0.3$ and the dimension of the input vector is $186 \times 0.3 = 56$.

a random selection of features is used instead of all available features. The feature randomness further increases the diversity among the trees, making the ensemble more resilient to noise and data outliers. Numerous hyperparameters, such as the number of trees in the ensemble and their maximum depth, influence the performance of random forest. Choosing appropriate hyperparameter values is crucial for achieving optimal prediction performance. In our work, we use a random search to identify the best combinations of hyperparameters for the given power-grid datasets.

APPENDIX C: ADDITIONAL EXAMPLES OF STATE ESTIMATION FOR THE SMALL POWER GRID l2rpn_case14_sandbox

In the main text, a few examples of state estimation are presented for the small power grid. Here we present additional examples, as shown in Figs. 10 and 11 for $P_o = 0.3$. These examples further demonstrate that the LSTM-based

framework is effective for predicting the full state of the power grid in terms of ρ and p_{or} when it is provided with partial observations of ρ .

APPENDIX D: ATTACK DETECTION AND STATE ESTIMATION FOR THE LARGE POWER GRID l2rpn_wcci_2022

For attack detection on the power grid l2rpn_wcci_2022, we set $P_o = 0.5$, so the input is a vector of capacity ρ values of 93 power lines. Figure 12 shows the results for attack occurrence detection [Fig. 12(a)] and attack location detection [Fig. 12(b)]. In Fig. 12(a), the LSTM framework has high accuracy, correctly predicting the absence or presence of an attack with probabilities of 0.84 and 0.82, respectively. Due to the large size of the power grid, identification of the location of the attack is more challenging. Figure 12(b) demonstrates that the LSTM framework can still achieve a high success rate for this task. While there

are a few incorrect predictions of the attack on different lines, most failures occur when the framework incorrectly predicts an under-attack scenario as a no-attack scenario, which is consistent with the results of attack occurrence detection in Fig. 12(a).

For state estimation on the power grid `l2rpn_wcci_2022`, we set $P_o = 0.3$, so 56 transmission lines are observed. Figures 13(a) and 13(c) present three segment examples comparing the true and predicted values for estimating ρ and p_{or} , respectively, with the corresponding comparison results shown in Figs. 13(b) and 13(d). Different colors represent the values of the deviation, with red indicating relatively large deviations. The overall MSE is indicated in the lower-right corner in Figs. 13(b) and 13(d). Here, the lines that can be directly observed are excluded. These results suggest the LSTM framework is effective for state estimation.

APPENDIX E: ATTACK DETECTION AND STATE ESTIMATION FOR THE LARGE POWER GRID `l2rpn_idf_2023`

For attack detection on `l2rpn_idf_2023`, we set $P_o = 0.5$, i.e., the input is a vector of the values of the capacity ρ associated with 93 power lines. Figure 14 shows the results for attack occurrence detection [Fig. 14(a)] and attack location detection [Fig. 14(b)]. In Fig. 14(a), the LSTM framework correctly predicts the absence or presence of an attack with probabilities of 0.85 and 0.72, respectively. Due to the large size of this power grid, identification of the location of the attack is more challenging. Figure 14(b) demonstrates that the LSTM framework can still achieve a high success rate for this task. While there are a few incorrect predictions of the attack on different lines, most failures occur when the framework incorrectly predicts an under-attack scenario as a no-attack scenario, which is consistent with the results of attack occurrence detection in Fig. 14(a).

For state estimation on `l2rpn_idf_2023`, we set $P_o = 0.3$, so 56 transmission lines are observed. Figures 15(a) and 15(c) present three segment examples comparing the true and predicted values of the estimates of ρ and p_{or} , respectively, with the corresponding comparison results shown in Figs. 15(b) and 15(d). Different colors represent the values of the deviation, with red indicating relatively large deviations. The overall MSE is indicated in the lower-right corner in Figs. 15(b) and 15(d). Here, the lines that can be directly observed are excluded. These results suggest that the LSTM framework is effective for state estimation.

APPENDIX F: PERFORMANCE OF LSTM UNDER DIFFERENT CLASS WEIGHTS

The attack occurrence detection task reported in the main text was performed under the 1:1 weighting scenario, where the no-attack and under-attack data are given

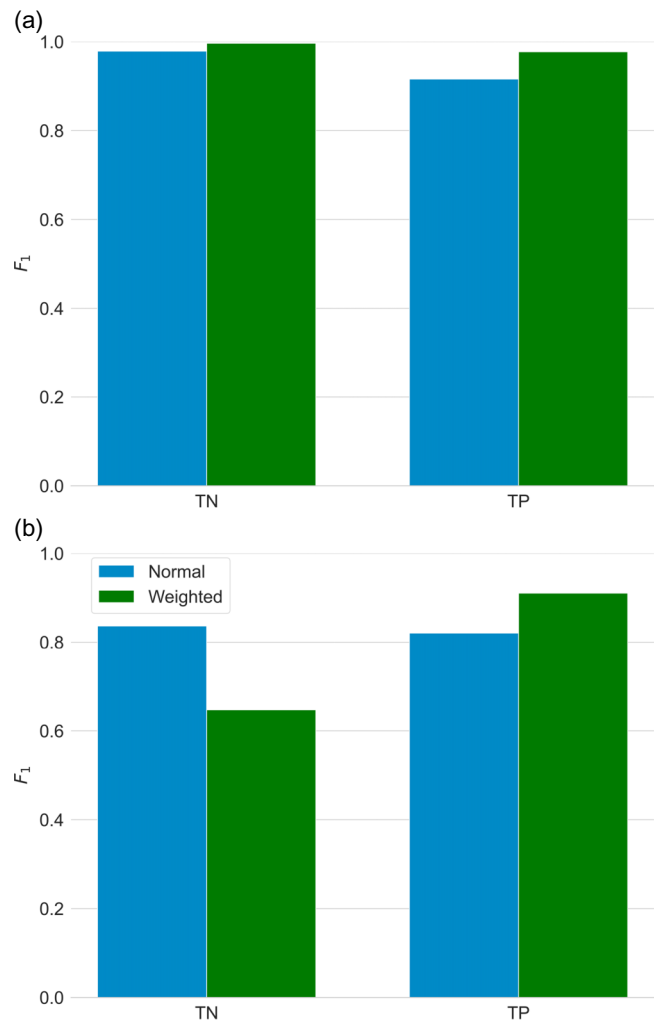


FIG. 16. Performance of LSTM under different class weights. Results are shown for (a) the small power grid with $P_o = 0.3$ and (b) the large power grid with $P_o = 0.5$. The blue and orange histograms are the results from LSTM trained under the class weights 1:1 and 1:2 separately. “TN” and “TP” indicate the probabilities of correctly predicting the no-attack and under-attack labels, respectively.

equal significance during the training. However, it may be desired to prioritize the detection of attacks in order to increase the sensitivity of the machine-learning framework to attack occurrence. To address this issue, we set the class weights to 1:2, giving higher importance to the under-attack label. This alternative weighting strategy can be advantageous in application scenarios where the machine needs to be more sensitive to attacks. Figure 16 presents a comparison between the 1:1 and 1:2 weighting cases for the small [Fig. 16(a)] and large [Fig. 16(b)] power grids. As shown in Fig. 16(a), the 1:2 weight ratio results in an improvement in correctly predicting the no-attack and the under-attack labels, generating a high classification accuracy. The corresponding results for the large power grid

are shown in Fig. 16(b), where an improvement in detecting the under-attack labels is achieved but at the expense of a reduced probability for correctly detecting the no-attack labels, suggesting that the trade-off between label accuracy and preference can be an important consideration in applications.

-
- [1] R. Baheti and H. Gill, Cyber-physical systems, *Impact Cont. Tech.* **12**, 161 (2011).
- [2] W. Wolf, Cyber-physical systems, *Computer* **42**, 88 (2009).
- [3] J. Ren, W.-X. Wang, B. Li, and Y.-C. Lai, Noise bridges dynamical correlation and topology in coupled oscillator networks, *Phys. Rev. Lett.* **104**, 058701 (2010).
- [4] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and M. A. F. Harrison, Time-series based prediction of complex oscillator networks via compressed sensing, *EPL (Europhys. Lett.)* **94**, 48006 (2011).
- [5] M. Timme and J. Casadiego, Revealing networks from dynamics: An introduction, *J. Phys. A Math. Theory* **47**, 343001 (2014).
- [6] Z. Shen, W.-X. Wang, Y. Fan, Z. Di, and Y.-C. Lai, Reconstructing propagation networks with natural diversity and identifying hidden sources, *Nat. Commun.* **5**, 1 (2014).
- [7] W.-X. Wang, Y.-C. Lai, and C. Grebogi, Data based identification and prediction of nonlinear and complex dynamical systems, *Phys. Rep.* **644**, 1 (2016).
- [8] J. Casadiego, M. Nitzan, S. Hallerberg, and M. Timme, Model-free inference of direct network interactions from nonlinear collective dynamics, *Nat. Commun.* **8**, 1 (2017).
- [9] H. Wang, C. Ma, H.-S. Chen, Y.-C. Lai, and H.-F. Zhang, Full reconstruction of simplicial complexes from binary contagion and Ising data, *Nat. Commun.* **13**, 1 (2022).
- [10] A. Banerjee, S. Chandra, and E. Ott, Network inference from short, noisy, low time-resolution, partial measurements: Application to *C. elegans* neuronal calcium dynamics, *Proc. Natl. Acad. Sci. USA* **120**, e2216030120 (2023).
- [11] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.* **9**, 1735 (1997).
- [12] L. R. Medsker and L. Jain, Recurrent neural networks, *Design Appl.* **5**, 64 (2001).
- [13] S. Grossberg, Recurrent neural networks, *Scholarpedia* **8**, 1888 (2013).
- [14] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, Recent advances in recurrent neural networks, [arXiv:1801.01078](https://arxiv.org/abs/1801.01078).
- [15] A. Graves, A.-R. Mohamed, and G. Hinton, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE, Vancouver, Canada, 2013), p. 6645.
- [16] H. Sak, A. Senior, K. Rao, and F. Beaufays, Fast and accurate recurrent neural network acoustic models for speech recognition, [arXiv:1507.06947](https://arxiv.org/abs/1507.06947).
- [17] K. M. Tarwani and S. Edem, Survey on recurrent neural network in natural language processing, *Int. J. Eng. Trends Technol.* **48**, 301 (2017).
- [18] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu, in *Interspeech* (International Speech Communication Association (ISCA), Lyon, France, 2013), p. 2524.
- [19] A. Jaech, L. Heck, and M. Ostendorf, Domain adaptation of recurrent neural networks for natural language understanding, [arXiv:1604.00117](https://arxiv.org/abs/1604.00117).
- [20] T. Guo, Z. Xu, X. Yao, H. Chen, K. Aberer, and K. Funaya, in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (IEEE, Montreal, Canada, 2016), p. 816.
- [21] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, A dual-stage attention-based recurrent neural network for time series prediction, [arXiv:1704.02971](https://arxiv.org/abs/1704.02971).
- [22] H. Fan, J. Jiang, C. Zhang, X. Wang, and Y.-C. Lai, Long-term prediction of chaotic systems with machine learning, *Phys. Rev. Res.* **2**, 012080 (2020).
- [23] H. Hewamalage, C. Bergmeir, and K. Bandara, Recurrent neural networks for time series forecasting: Current status and future directions, *Int. J. Forecast.* **37**, 388 (2021).
- [24] L.-W. Kong, H.-W. Fan, C. Grebogi, and Y.-C. Lai, Machine learning prediction of critical transition and system collapse, *Phys. Rev. Res.* **3**, 013090 (2021).
- [25] Z.-M. Zhai, L.-W. Kong, and Y.-C. Lai, Emergence of a resonance in machine learning, *Phys. Rev. Res.* **5**, 033127 (2023).
- [26] L.-W. Kong, Y. Weng, B. Glaz, M. Haile, and Y.-C. Lai, Reservoir computing as digital twins for nonlinear dynamical systems, *Chaos* **33**, 033111 (2023).
- [27] Z.-M. Zhai, M. Moradi, L.-W. Kong, B. Glaz, M. Haile, and Y.-C. Lai, Model-free tracking control of complex dynamical trajectories with machine learning, *Nat. Commun.* **14**, 5698 (2023).
- [28] S. Panahi and Y.-C. Lai, Adaptable reservoir computing: A paradigm for model-free data-driven prediction of critical transitions in nonlinear dynamical systems, *Chaos* **34**, 051501 (2024).
- [29] L.-W. Kong, G. A. Brewer, and Y.-C. Lai, Reservoir-computing based associative memory and itinerancy for complex dynamical attractors, *Nat. Commun.* **15**, 4840 (2024).
- [30] Z.-M. Zhai, M. Moradi, L.-W. Kong, and Y.-C. Lai, Detecting weak physical signal from noise: A machine-learning approach with applications to magnetic-anomaly-guided navigation, *Phys. Rev. Appl.* **19**, 034030 (2023).
- [31] K. Antczak, Deep recurrent neural networks for ECG signal denoising, [arXiv:1807.11551](https://arxiv.org/abs/1807.11551).
- [32] A. G. Parlos, S. K. Menon, and A. Atiya, An algorithmic approach to adaptive state filtering using recurrent neural networks, *IEEE Trans. Neural Netw.* **12**, 1411 (2001).
- [33] M. Moradi, Y. Weng, and Y.-C. Lai, Defending smart electrical power grids against cyberattacks with deep Q-learning, *PRX Energy* **1**, 033005 (2022).
- [34] Y. Peng, T. Lu, J. Liu, Y. Gao, X. Guo, and F. Xie, in *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (IEEE, Beijing, China, 2013), p. 442.
- [35] A. A. Zúñiga, A. Baleia, J. Fernandes, and P. J. D. C. Branco, Classical failure modes and effects analysis in the context of smart grid cyber-physical systems, *Energies* **13**, 1215 (2020).
- [36] Y. Jiang, S. Yin, and O. Kaynak, Data-driven monitoring and safety control of industrial cyber-physical systems: Basics and beyond, *IEEE Access* **6**, 47374 (2018).

- [37] C.-C. Sun, A. Hahn, and C.-C. Liu, Cyber security of a power grid: State-of-the-art, *Int. J. Electr. Power Energy Syst.* **99**, 45 (2018).
- [38] E. U. Soykan, M. Bagriyanik, and G. Soykan, Disrupting the power grid via EV charging: The impact of the SMS phishing attacks, *Sustainable Energy Grid. Netw* **26**, 100477 (2021).
- [39] H. Holm, W. R. Flores, and G. Ericsson, in *IEEE PES ISGT Europe 2013* (IEEE, Lyngby, Denmark, 2013), p. 1.
- [40] T. Akhtar, B. B. Gupta, and S. Yamaguchi, in *2018 IEEE International Conference on Consumer Electronics (ICCE)* (IEEE, Las Vegas, NV, USA, 2018), p. 1.
- [41] M. J. Assante, Confirmation of a coordinated attack on the Ukrainian power grid, SANS Industrial Control Systems Security Blog **207** (2016).
- [42] P. Srikantha and D. Kundur, in *2015 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)* (IEEE, Washington, D.C., USA, 2015), p. 1.
- [43] S. Liu, X. P. Liu, and A. El Saddik, in *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)* (IEEE, Washington, D.C., USA, 2013), p. 1.
- [44] J. Tian, B. Wang, T. Li, F. Shang, and K. Cao, Coordinated cyber-physical attacks considering DoS attacks in power systems, *Int. J. Robust Nonlinear Control* **30**, 4345 (2020).
- [45] D. U. Case, Analysis of the cyber attack on the Ukrainian power grid, *Elec. Info. Sharing Anal. Cen. (E-ISAC)* **388**, 1 (2016).
- [46] J. E. Sullivan and D. Kamensky, How cyber-attacks in Ukraine show the vulnerability of the US power grid, *Electricity J.* **30**, 30 (2017).
- [47] S. Soltan, M. Yannakakis, and G. Zussman, Joint cyber and physical attacks on power grids: Graph theoretical approaches for information recovery, *ACM SIGMETRICS Perf. Eva. Rev.* **43**, 361 (2015).
- [48] S. Lakshminarayana, E. V. Belmega, and H. V. Poor, Moving-target defense against cyber-physical attacks in power grids via game theory, *IEEE Trans. Smart Grid* **12**, 5244 (2021).
- [49] P. W. Parfomak, Physical security of the US power grid: High-voltage transformer substations (2014).
- [50] P. W. Parfomak, *NERC Standards for Bulk Power Physical Security: Is the Grid More Secure?* Congressional Research Service, Washington, DC (2018).
- [51] M. Landen, K. Chung, M. Ike, S. Mackay, J.-P. Watson, and W. Lee, Dragon: in *Proceedings of the 38th Annual Computer Security Applications Conference* (The Association for Computing Machinery (ACM), Austin, TX, USA, 2022), p. 13.
- [52] G. Cerullo, V. Formicola, P. Iamiglio, and L. Sgaglione, Critical infrastructure protection: having SIEM technology cope with network heterogeneity, [arXiv:1404.7563](https://arxiv.org/abs/1404.7563).
- [53] V. K. Singh, S. P. Callupe, and M. Govindarasu, in *2019 North American Power Symposium (NAPS)* (IEEE, Wichita, Kansas, USA, 2019), p. 1.
- [54] Z. He, A. Raghavan, G. Hu, S. Chai, and R. Lee, in *2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)* (IEEE, Rotorua, New Zealand, 2019), p. 160.
- [55] M. Panthi, in *2020 First International Conference on Power, Control and Computing Technologies (ICPC2T)* (IEEE, Raipur, India, 2020), p. 220.
- [56] W. Danilczyk, Y. L. Sun, and H. He, in *2020 52nd North American Power Symposium (NAPS)* (IEEE, Tempe, Arizona, USA, 2020), p. 1.
- [57] S. Li, Y. Han, X. Yao, S. Yingchen, J. Wang, and Q. Zhao, Electricity theft detection in power grids with deep learning and random forests, *J. Electr. Comput. Eng.* **2019**, 4136874 (2019).
- [58] O. F. Eikeland, I. S. Holmstrand, S. Bakkejord, M. Chiesa, and F. M. Bianchi, Detecting and interpreting faults in vulnerable power grids with machine learning, *IEEE Access* **9**, 150686 (2021).
- [59] Q. Li, G. Zou, W. Zeng, J. Gao, F. He, and Y. Zhang, ESG guidance and artificial intelligence support for power systems analytics in the energy industry, *Sci. Rep.* **14**, 11347 (2024).
- [60] E. Choi, S. Cho, and D. K. Kim, Power demand forecasting using long short-term memory (LSTM) deep-learning model for monitoring energy sustainability, *Sustainability* **12**, 1109 (2020).
- [61] H. Abbasimehr, M. Shabani, and M. Yousefi, An optimized model using LSTM network for demand forecasting, *Comput. Ind. Eng.* **143**, 106435 (2020).
- [62] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, A survey on anomaly detection for technical systems using LSTM networks, *Comput. Ind.* **131**, 103498 (2021).
- [63] C. Feng, T. Li, and D. Chana, in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (IEEE, Denver, CO, USA, 2017), p. 261.
- [64] M. M. Devi, M. Sharma, and A. Ganguly, in *2023 IEEE Int. Conf. Ind. Electron. Dev. Appl. (ICIDEA)* (IEEE, Imphal, India, 2023), p. 146.
- [65] D. Guha, R. Chatterjee, and B. Sikdar, Anomaly detection using LSTM-based variational autoencoder in unsupervised data in power grid, *IEEE Syst. J.* **17**, 4313 (2023).
- [66] A. S. Musleh, G. Chen, Z. Y. Dong, C. Wang, and S. Chen, Attack detection in automatic generation control systems using LSTM-based stacked autoencoders, *IEEE Trans. Ind. Inf.* **19**, 153 (2022).
- [67] S. Wang, S. Bi, and Y.-J. A. Zhang, Locational detection of the false data injection attack in a smart grid: A multilabel classification approach, *IEEE Internet Things J.* **7**, 8218 (2020).
- [68] S. Peng, Z. Zhang, R. Deng, and P. Cheng, Localizing false data injection attacks in smart grid: A spectrum-based neural network approach, *IEEE Trans. Smart Grid.* **14**, 4827 (2023).
- [69] J. Zhu, W. Meng, M. Sun, J. Yang, and Z. Song, FLLF: A fast-lightweight location detection framework for false data injection attacks in smart grids, *IEEE Trans. Smart Grid.* **15**, 911 (2023).
- [70] M. Mohammadpourfard, I. Genc, S. Lakshminarayana, and C. Konstantinou, in *2021 IEEE SmartGridComm* (IEEE, Aachen, Germany, 2021), p. 121.
- [71] O. Boyaci, M. R. Narimani, K. R. Davis, M. Ismail, T. J. Overbye, and E. Serpedin, Joint detection and localization of stealth false data injection attacks in smart grids using

- graph neural networks, *IEEE Trans. Smart Grid.* **13**, 807 (2021).
- [72] J. Yu, Q. Li, and L. Li, Localization of coordinated cyber-physical attacks in power grids using moving target defense and machine learning, *Electronics* **13**, 2256 (2024).
- [73] B. Donnot, Grid2op—A testbed platform to model sequential decision making in power systems, <https://GitHub.com/rte-france/grid2op> (2020).
- [74] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, OpenAI Gym, [arXiv:1606.01540](https://arxiv.org/abs/1606.01540).
- [75] A. Marot, B. Donnot, G. Dulac-Arnold, A. Kelly, A. O’Sullivan, J. Viebahn, M. Awad, I. Guyon, P. Panciatichi, and C. Romero, in *NeurIPS 2020 Competition and Demonstration Track* (PMLR, 2021), p. 112.
- [76] L. Omnes, A. Marot, and B. Donnot, in *2021 IEEE Madrid PowerTech* (IEEE, Madrid, Spain, 2021), p. 1.
- [77] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, *ACM Comput. Surv.* **55**, 1 (2023).
- [78] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, Image segmentation using deep learning: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 3523 (2021).
- [79] L. Breiman, Random forests, *Mach. Learn.* **45**, 5 (2001).
- [80] M. Moradi, Z.-M. Zhai, A. Nielsen, and Y.-C. Lai, Random forests for detecting weak signals and extracting physical information: A case study of magnetic navigation, *APL Mach. Learn.* **2**, 016118 (2024).
- [81] H. Henderi, T. Wahyuningsih, and E. Rahwanto, Comparison of min-max normalization and Z-score normalization in the k-nearest neighbor (KNN) algorithm to test the accuracy of types of breast cancer, *Int. J. Info. Inf. Syst.* **4**, 13 (2021).
- [82] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, Massachusetts, USA, 2016).
- [83] Y.-S. Long, Z.-M. Zhai, M. Tang, and Y.-C. Lai, Metamorphoses and explosively remote synchronization in dynamical networks, *Chaos* **32**, 043110 (2022).
- [84] L. Zeng, M. Tang, and Y. Liu, The impacts of the individual activity and attractiveness correlation on spreading dynamics in time-varying networks, *Commun. Nonlinear Sci. Numer. Simul.* **122**, 107233 (2023).
- [85] Y.-S. Long, Z.-M. Zhai, M. Tang, Y. Liu, and Y.-C. Lai, Structural position vectors and symmetries in complex networks, *Chaos* **32**, 093132 (2022).
- [86] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, A comprehensive survey on graph neural networks, *IEEE Trans. Neu. Net. Learn. Syst.* **32**, 4 (2020).
- [87] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, *Adv. Neural Inf. Process. Syst.* **30** (2017).
- [88] J. Ho, A. Jain, and P. Abbeel, Denoising diffusion probabilistic models, *Adv. Neural Inf. Process. Syst.* **33**, 6840 (2020).
- [89] M. Moradi, Y. Weng, J. Dirkman, and Y.-C. Lai, Preferential cyber defense for power grids, *PRX Energy* **2**, 043007 (2023).
- [90] M. Moradi, S. Panahi, Z.-M. Zhai, Y. Weng, J. Dirkman, and Y.-C. Lai, Heterogeneous reinforcement learning for defending power grids against attacks, *APL Mach. Learn.* **2**, 026121 (2024).
- [91] Z.-M. Zhai, Simulation data of the three power grid benchmarks, <https://zenodo.org/records/14004431> (2024).
- [92] Z.-M. Zhai, Codes for generating all the results, <https://github.com/Zheng-Meng/Power-Grid-Attack-Detection> (2024).