Received: 1 November 2021 | Revised: 31 January 2022 | Accepted: 10 March 2022

IET Smart Grid

DOI: 10.1049/stg2.12066

ORIGINAL RESEARCH

Data-driven detection and identification of IoT-enabled load-altering attacks in power grids

Subhash Lakshminarayana[1] | Saurav Sthapit[2] | Hamidreza Jahangir[1] | Carsten Maple[2] | H. Vincent Poor[3]

[1]School of Engineering, University of Warwick, Coventry, UK

[2]Warwick Manufacturing Group, University of Warwick, Coventry, UK

[3]Department of Electrical and Computer Engineering, Princeton University, Princeton, New Jersy, USA

Correspondence

Subhash Lakshminarayana, School of Engineering, University of Warwick, Coventry, UK.
Email: Subhash.Lakshminarayana@warwick.ac.uk

Funding information

U.S. National Science Foundation, Grant/Award Number: 2039716

Abstract

Advances in edge computing are powering the development and deployment of Internet of Things (IoT) systems to provide advanced services and resource efficiency. However, large-scale IoT-based load-altering attacks (LAAs) can seriously impact power grid operations, such as destabilising the grid's control loops. Timely detection and identification of any compromised nodes are essential to minimise the adverse effects of these attacks on power grid operations. In this work, two data-driven algorithms are proposed to detect and identify compromised nodes and the attack parameters of the LAAs. The first method, based on the Sparse Identification of Nonlinear Dynamics approach, adopts a sparse regression framework to identify attack parameters that best describe the observed dynamics. The second method, based on physics-informed neural networks, employs neural networks to infer the attack parameters from the measurements. Both algorithms are presented utilising edge computing for deployment over decentralised architectures. Extensive simulations are performed on IEEE 6-, 14-, and 39-bus systems to verify the effectiveness of the proposed methods. Numerical results confirm that the proposed algorithms outperform existing approaches, such as those based on unscented Kalman filter, support vector machines, and neural networks (NN), and effectively detect and identify locations of attack in a timely manner.

1 | INTRODUCTION

Critical infrastructures (such as power grids, transportation systems, and nuclear plants) are subject to unprecedented cyber attacks against their system operations [1–3]. In particular, power grids are attractive targets since their disruption can have severe social and economic consequences. The growing integration of Internet of Things (IoT)-enabled smart-home appliances at the consumer end, such as Wi-Fi-enabled air conditioners, battery storage systems, plug-in-electric vehicles, and so forth, present a new threat to power grid operations. These intelligent appliances are already widely prevalent since they provide convenience to consumers and improve energy efficiency. In the year 2020, it was assessed that globally, there are 30 billion IoT-enabled devices [4]. However, due to their poor security features [5, 6], they may become convenient entry points for malicious parties to gain access to the system. Thus, cyber security can be the main impediment in the widespread adoption of IoT-enabled electrical appliances.

Cyber security of power grids has received widespread attention in recent years. A significant body of work is dedicated to utility-side cyber attacks and the associated SCADA system security. In particular, false data injection (FDI) attacks against power system state estimation have received significant attention [7–9]. Different from the aforementioned works, this paper considers cyber attacks that target the end-user consumer devices [10], and their impact on power grid operations. Unlike utility-side and SCADA assets, end-user devices (such as IoT-smart home appliances) can be compromised by adversarial units due to their large numbers, and inefficient monitoring. Thus, cyber threats targeting demand-side attacks pose unique challenges to power grid operators.

Sudden and abrupt manipulation of the power grid demand due to large-scale cyber attacks against IoT-smart-home appliances (e.g. a Botnet-type attack) can potentially disrupt the balance between supply and demand and lead to severe effects on power grid operations. Recent works have quantified their impact on power grid operations. In particular, [11, 12] showed that large-scale IoT-based load-altering attacks (LAAs) could lead to unsafe frequency excursions, subsequently triggering generator trips and resulting in large-scale blackouts. Moreover, such attacks can also be used to increase the operational cost of the power grid [12]. Subsequent work [13] showed that existing protection mechanisms such as under-frequency load shedding (UFLS) are capable of preventing large-scale blackouts. Nevertheless, LAAs can still partition bulk power systems and/or a controlled load-shedding event. Demand-side cyber attacks can also propagate from the distribution grid to the transmission grid, which can lead to an attack impact over a larger geographical area [14]. Ospina et al. [15] studied the feasibility of LAAs under different power grid loading conditions.

Static LAAs leading to a sudden one-time manipulation of the demand (as proposed in [11–14]) cannot destabilise the power grid's frequency control loop. A more severe attack known as the *dynamic LAAs* was introduced in [16], which showed that if the attacker manipulates the load over a period of time in accordance with the fluctuations of the grid frequencies, the resulting attack can destabilise the frequency control loop, which can subsequently trigger cascading failures. An analytical method to understand the impact of static/dynamic LAAs using the theory of second-order dynamical systems was presented in [17], which was used to find the locations of nodes from which an attacker can launch the most impactful LAAs. Subsequent work investigated techniques to enhance the resilience of power grids to LAAs by improving the security features of a fraction of smart loads [16, 17], and finding generator operating points that ensure that no transmission lines are overloaded after any potential LAAs [18].

Despite the growing number of works on analysing the impact and improving the resilience of power grids against LAAs, there is relatively less work on detecting and identifying such attacks in real-time, which is the main focus of this paper. Following the onset of an LAA, swift detection and identification of the compromised nodes are critical to minimising the adverse effects and service disruptions. In this work, we focus specifically on detecting and identifying LAAs by real-time monitoring and analysis of the grid's physical signals such as the voltages, frequencies etc. The growing deployment of phasor-measurement units (PMUs) provide fine-grained monitoring of these signals. The key advantage of this approach is that grid operators can leverage existing sensing infrastructure in the power grid (e.g. advanced metering infrastructure, phasor measurement units etc.), and no major device upgrades to IoT-enabled electrical appliances (e.g. enabling encryption/device remote attestation) would be necessary. Such an approach is particularly relevant in the context of IoT-based attacks due to the massive number of devices and lack of a unified approach to implementing security standards.

The task is challenging, however, since well-designed LAAs can remain undetected by traditional pattern recognition algorithms applied to the load consumption data [19]. Amini et al. [19] showed that LAAs can, however, be detected by analysing the system measurements (e.g. load, frequency etc.) in the frequency domain since LAAs can result in new frequency components that are non-existent in the original data. The method was subsequently extended to identifying the location of the compromised nodes using an optimization-based framework in [20]. An unscented Kalman filter (UKF)-based approach to detect and identify the attack parameters was proposed in [21] in which the attack parameters were augmented to the power grid's state variables and estimated jointly.

However, a major limitation of existing work (such as [19, 20]) is that the formulation used is specific to a linear power system model. While the theory of UKF proposed in [21] is applicable for non-linear systems, when the power grid dimensions grow, the number of attack parameters to be estimated (which in turn depends on the different combinations of nodes that can be attacked as well as the attacker's chosen parameters) scales much faster than the number of system state measurements. Thus, the UKF approach is not scalable to power grids of large dimensions since the estimation problem becomes under-specified [21]. Moreover, our results show that the UKF approach cannot track the attack parameters accurately when the system is highly dynamic.

To overcome the aforementioned issues, this paper adopts state-of-the-art data-driven methods for dynamical systems in order to detect and identify LAAs using real-time monitoring of the power grid's physical signals, such as frequency/phase angle measurements monitored by PMUs. The application is based on the observation is that LAAs lead to a change in the parameters of the power system's dynamical model [16]. Thus, the attack detection/identification involves solving the following question: *given a series of frequency/phase angle observations over time*, *what are attack the parameters that best describe the observed data?* We model the problem as one of *data-driven discovery* of power grid's non-linear dynamical equations [22, 23].

We apply two state-of-the-art data-driven methods to solve the aforementioned problem, namely (i) a Sparse Regression (SR) algorithm and (ii) a physics-informed neural network (PINN) algorithm [23]. The SR algorithm is motivated by the Sparse Identification of Nonlinear Dynamics (SINDy) approach proposed in [22]. It applies sparse regression to determine the fewest attack parameters that best describe the observed measurements. In the context of attack detection/identification, the rationale for seeking a sparse solution is that in a real-world scenario, the attacker can likely monitor the frequency fluctuations and manipulate the load at only a few nodes in the power grid. In particular, we use the least absolute shrinkage and selection operator (LASSO) method to obtain a sparse solution to the LAA detection/identification problem. The PINN method on the other hand uses deep neural networks (DNNs) to approximate both the state of the power grid (voltage phase angles/frequency) as well as the unknown nonlinear dynamics. It treats the DNN weights as well as the attack parameters as variables to obtain the parameters that best describe the observed dynamics and the mechanisms that govern the evolution of the observed measurements. In both the algorithms, the solution to the attack parameter estimation problem will automatically determine which load buses are compromised by the attacker and those that are not. Furthermore, we present a methodology for the implementation of each method, particularly showing the implementation utilising edge computing deployed over decentralised architectures, requiring limited information exchange between the neighbouring nodes.

We note that there is a growing body of work that applies ML to detect FDI attacks against power grids. Despite their effectiveness, they cannot be applied directly to the problem at hand, since (i) we are interested in both detecting the attacks as well as identifying the attack locations. Existing works on FDI attack detection (e.g. [24–26]) apply binary classification solutions, that is, classify whether the attack has occurred or not; they do not localise the attack. Identifying the attack location requires a multi-class classification approach. For the problem at hand, such an approach will have exponential complexity, since the defender must consider every combination of victim nodes that the attacker can target. To overcome this issue, the proposed SR and PINN algorithms leverage the knowledge of the power grid's 'physical model' to infer the attack parameters with a small amount of training data. (ii) ML techniques applied to the FDI attack detection problem only consider the steady-state operation of power grids. In contrast, our work considers ML algorithms that consider the power grid dynamics and applies data-driven methods for dynamical systems.

We conduct extensive simulations using benchmark IEEE bus systems to evaluate the performance of these algorithms in detecting and identifying LAAs. Our results show that both the SR and the PINN algorithms are more effective in estimating the attack parameters as compared to the other benchmark approaches, including support vector machines (SVM), neural network (NN), and the UKF approach, especially in systems that exhibit fast dynamics. Furthermore, both the algorithms can estimate the attack parameters within a short observation time window. However, the PINN algorithm does not perform well on systems exhibiting slow, and oscillatory dynamics since the training may get stuck in local minima. On the other hand, the SR algorithm performs reliably well under all system conditions.

To summarise, the main contributions of this work are as follows:

- Proposing novel data-driven approaches (SR and PINN) to detect and localise IoT-based load-altering attacks against power grids.
- Estimating the static/dynamic attack parameters by monitoring the frequency/phase angle dynamics over an observation time window.
- Verifying the effectiveness of the proposed methods on slow oscillatory and fast dynamics networks.
- Comparing the performance of the proposed methods with benchmark techniques, including SVM, NN, and UKF.

The rest of the paper is organised as follows. Section 2.1 introduces the system model; Section 3 describes the SR and the PINN algorithms. Section 4 includes discussion on practical aspects as well as a decentralised implementation of the proposed algorithms. Section 5 describes the simulation results and Section 6 concludes. The simulation parameters are listed in Appendix.

2 | PRELIMINARIES

2.1 | System model

We consider a power grid consisting of a set of $\mathcal{N} = \{1, \ldots, N\}$ buses. For each node $i \in \mathcal{N}$, the set of neighbouring nodes is denoted by $\mathcal{N}_i$. The buses are divided into generator buses $\mathcal{N}_G$ and load buses $\mathcal{N}_L$ and $\mathcal{N} = \mathcal{N}_G \cup \mathcal{N}_L$. The power grid dynamic model is given by [27]:

$$\dot{\delta}_i = \omega_i, i \in \mathcal{N}_G \tag{1}$$

$$M_i\dot{\omega}_i = -D_i\omega_i - K_i^P\omega_i - K_i^I\delta_i - P_i^F,\ i \in \mathcal{N}_G, \tag{2}$$

$$D_i\dot{\delta}_i = -P_i^L - P_i^F,\ i \in \mathcal{N}_L, \tag{3}$$

where $P_i^F = \sum_{j\in\mathcal{N}} B_{i,j}\sin(\delta_{ij})$. In the above, δ_i is the phase angle deviation at bus $i \in \mathcal{N}$, $\delta_{ij} = \delta_i - \delta_j$, $i,j \in \mathcal{N}$ and ω_i denotes the rotor frequency deviation at the generator buses $i \in \mathcal{N}_G$. The generator inertia coefficient at bus $i \in \mathcal{N}_G$ is denoted by M_i. The damping coefficients at generator/load buses are denoted by D_i $i \in \mathcal{N}$. $B_{i\,j}$ is the suscpetance of line i, j. Finally, P_i^L denotes the load at bus $i \in \mathcal{N}_L$.

We denote ω_{nom} as the grid's nominal frequency, for example, 50 Hz in Europe or 60 Hz in North America. For safe operations, the frequency must be maintained within the safety limits. We denote $\omega_{\max}$ as the maximum permissible frequency deviation for system safety. Thus, $|\omega_{\text{nom}} - \omega_i| \leq \omega_{\max}, \forall i \in \mathcal{G}$. We note that in steady state, $\dot{\omega}_i = 0, \forall i \in \mathcal{G}$.

2.2 | Load-altering attacks

IoT-enabled end-user appliances often have poor security features. Several such vulnerabilities have been discovered in commercial IoT high-wattage appliances. For instance, it has been shown that Mitsubishi ACs are prone to XML external entity injection (XXE) vulnerabilities [28, 29]. They can be exploited to launch denial-of-service (DoS) or privilege escalation (unauthorised access) attacks. In particular, the latter can be exploited by an external entity (such as an attacker) to change the operational settings of ACs (e.g. switch ON/OFF or change the power settings). Similarly, other IoT power appliances such as photovoltaic (PV) inverters [30] and electric vehicle (EV) chargers [31] are also known to have cyber vulnerabilities.

Botnet-type attacks against IoT devices, such as the Mirai attack that controlled up to 600,000 devices, have been well-documented [32]. If such attacks are launched against IoT-enabled electrical appliances, they can significantly disrupt power grid operations. Assuming that a commercial AC has a power rating of 2 kW, the exploitation of tens of thousands of such devices can lead to several megawatts of load alteration, significantly disrupting the balance between supply and demand. In this work, we do not explicitly model how an attacker can launch Botnet-type attacks against the power grid. Rather, inspired by the aforementioned vulnerabilities, we focus on the impact of such attacks on power grid operations and the corresponding detection/identification mechanisms. In the following, we present a model of LAAs in the context of power grid control loops described in (1)–(3).

Under IoT-based LAAs, the attacker manipulates the system load by synchronously switching on or off a large number of high-wattage devices. Assume that the demand at the load buses consists of two components $P_i^L = P_i^{LS} + P_i^{LV}$, where P_i^{LS} denotes the secure part of the load (i.e. load that cannot be altered) and P_i^{LV} denotes the vulnerable part of the load at node $i \in \mathcal{N}_l$. We denote the set of victim nodes by $\mathcal{N}_v (\subseteq \mathcal{N}_L)$, and $N_v = |\mathcal{N}_v|$, which are the subset of load buses at which the attacker can manipulate the load. The system load under LAAs is given by

$$P_i^L = -\sum_{j=1}^{S} K_{i,k_j}^L \omega_{k_j} + \epsilon_i^L + P_i^{LS} \tag{4}$$

Herein, ϵ_i^L is a step-change in the load introduced by the attacker (static LAA component). Note $\epsilon_i^L = 0$ if $i \notin \mathcal{N}_v$. The component $-\sum_{k_j \in \mathcal{S}} K_{i,k_j}^L \omega_{k_s}$ is the dynamic LAA component. Note that to inject the dynamic LAA, the attacker must monitor the frequency fluctuations at a subset of the buses in the system, $\omega_{k_j}, k_j \in \mathcal{S}$. Herein, $\mathcal{S} (\subseteq \mathcal{N})$, denotes the set of *sensing* buses, that is, buses at which the attacker can monitor the frequency fluctuations, and k_j denotes the index of the jth sensing bus and $S = |\mathcal{S}|$. K_{i,k_j}^L denotes attack controller gain values corresponding to the attack at load bus $i \in \mathcal{N}_L$ that follows the frequency at bus $k_j \in \mathcal{S}$. Note that $K_{i,k_s}^L \geq 0$, and therefore, the dynamic LAA component increases the system load when the frequency falls below the setpoint and vice versa; this has the opposite effect of arresting the frequency deviation. Furthermore, we have $\epsilon_i^L \leq P_i^{LV}$ and

$$\sum_{j=1}^{S} K_{i,k_j}^L \omega^{\max} \leq \left(P_i^{LV} - \epsilon_i^L\right)/2, \forall i \in \mathcal{V}. \tag{5}$$

The above limit on the dynamic LAA component can be explained as follows. The left-hand side of (5) is the maximum load value that must be altered by the attacker at victim bus $i \in \mathcal{V}$ before the frequency at the sensor bus k_j exceeds the safety limit $\omega^{\max}$. This must be less than the amount of vulnerable load $P_i^{LV} - \epsilon_i^L$ (after removing the static LAA component). Note that the amount of load that can be compromised under dynamic LAA must allow for both over and under frequency fluctuations before the system frequency exceeds $\omega^{\max}$ (see [16]). Thus, the right-hand side of (5) is divided by 2. Under LAA described in (4), the power grid dynamics noted in (1)–(3) becomes

$$\dot{\delta}_i = \omega_i, i \in \mathcal{N}_G \tag{6}$$

$$M_i \dot{\omega}_i = -D_i \omega_i - K_i^P \omega_i - K_i^I \delta_i - P_i^F, \ i \in \mathcal{N}_G, \tag{7}$$

$$D_i \dot{\delta}_i = \sum_{j=1}^{S} K_{i,k_j}^L \omega_{k_j} - \epsilon_i^L - P_i^{LS} - P_i^F, \ i \in \mathcal{N}_L, \tag{8}$$

Equations (6)–(8) jointly describe the power grid dynamics under LAAs.

2.3 | Impact of LAAs

To illustrate the attack impact, we show the phase angle and frequency dynamics for the IEEE-6 bus system under LAAs in Figure 1.

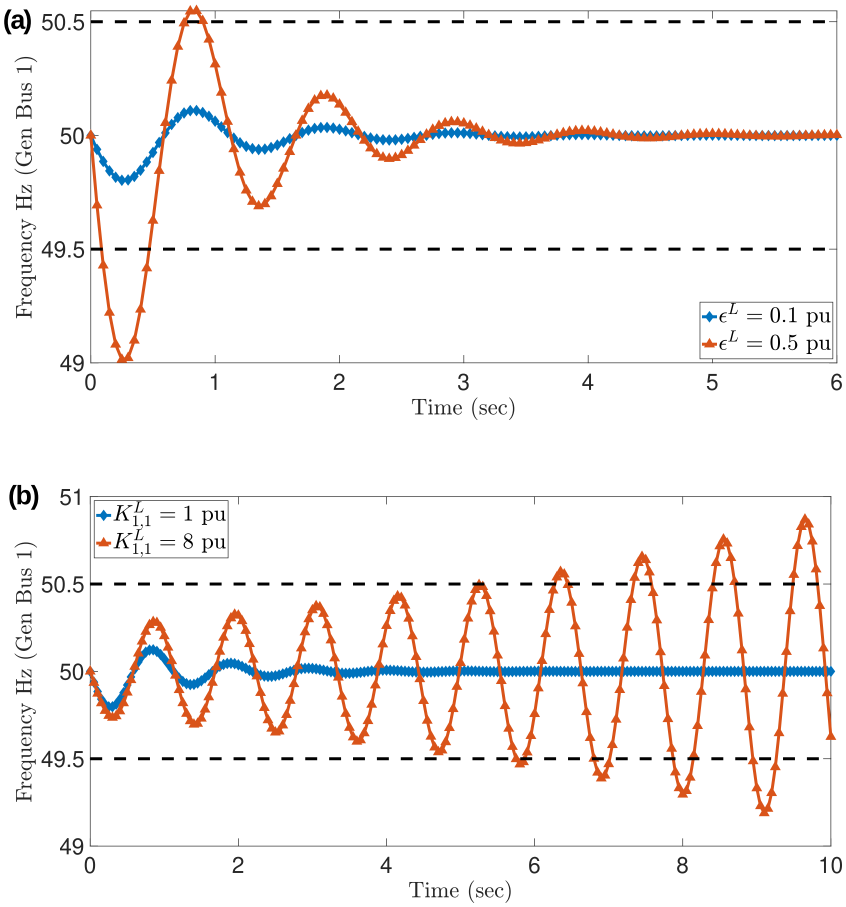
2.3.1 | Static LAAs

First, we consider static LAAs only, that is, when $\epsilon_i^L \neq 0$, whereas $K_{i,k_j}^L = 0$ (dynamic component). Note that static LAAs only act as an initial perturbation on the system. Small perturbations (corresponding to natural load fluctuations in the system) lead to minor frequency fluctuations, and the generator's control loops act quickly to counter the frequency deviation. However, LAAs that cause large-scale fluctuation in the system frequency can lead to large frequency fluctuations, leading to unsafe frequency excursions. An example of frequency fluctuations following a static LAA of 0.5 pu (base load of 100 MW) is plotted shown in Figure 1a. Note also from Figure 1a that small-scale load changes only cause minor frequency deviations.

2.3.2 | Dynamic LAAs

As compared to static LAAs, dynamic LAAs, that is, attacks with non-zero values of K_{i,k_j}^L change the dynamic model of the system, which can be noted by comparing Equations (3) and (8). This can alter the system design considerations and potentially destabilise the grid's frequency control loop. In Figure 1b, we plot the frequency dynamics for different values of K_{i,k_j}^L for the IEEE-6 bus system. As evident, the frequency control loop destabilises for large values of attack controller

FIGURE 1 Frequency dynamics under (a) static large-scale IoT-based load-altering attacks (LAAs); (b) dyanmic LAAs for IEEE-6 bus system



gain (see [17] for more analysis on this parameter). Moreover, we note that at different values the attack controller gain generate different patterns of frequency dynamics (more discussion is provided in Section 5).

3 | DATA-DRIVEN TECHNIQUES FOR DETECTING AND IDENTIFYING LAAS

This work focuses on detecting and identifying IoT-based LAAs by real-time monitoring and analysis of the grid's physical signals. In particular, we assume that the system operator has deployed PMUs in the power grid, which enables them to monitor the voltage phase angles $\left\{\delta_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\ldots,T}$ and frequency fluctuations $\left\{\dot{\delta}_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\ldots,T}$, respectively, over a period of time. Herein, we assume a slotted time system with $x^{(\tau)}$ as the value of the signal x at time slot τ, where the slots are sampled at a time interval of T_s and T is the total number of time slots. For instance, according to IEEE/IEC standards, for a 50 Hz system, the PMU sampling frequency can be between 10 and 100 frames per second. Therefore, T_s is in the range 10–100 ms [33]. Attack detection/identification problem involves inferring the parameters $\left\{K^L_{i,k_j}\right\}_{i\in\mathcal{V},k_j\in\mathcal{S}}$ and ϵ^L_i, $i \in \mathcal{V}$ by monitoring the power grid dynamics $\left\{\delta_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\ldots,T}$ and $\left\{\omega_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\ldots,T}$.

The overall implementation of the proposed methodology and its integration into power grid wide-area monitoring systems (WAMS) is shown in Figure 2. IoT-enabled electrical appliances are installed and operated by end-users, who typically lack security hygiene. Moreover, due to their massive numbers, and the lack of a unified approach to implementing security standards, IoT-based electrical appliances are insecure and vulnerable to attacks [5, 6]. In contrast, the PMU networks that monitor the power grid's physical signals are deployed by the system operators (e.g. at substations). It is feasible to deploy advanced security solutions to these systems and ensure the confidentiality and integrity of the monitored physical signals. For instance, the IEC 62351 standard provides security recommendations for different power system communication protocols, including the widely used IEC 61850 protocol for PMU/substation communication [34]. Thus, we assume that the attacker cannot modify the PMU-monitored signals, making the proposed data-driven attack detection/identification particularly suitable for detecting LAAs. In particular, the proposed approach will localise the buses (i.e. the substations) that are targeted by the attacker under LAAs. We provide further details on the integration of the proposed detection scheme into WAMS in Section 4.

In particular, we propose physics-informed machine learning algorithms to detect and identify LAAs. We note that the problem at hand can be modelled as supervised learning that learns only based on the training samples and does not rely

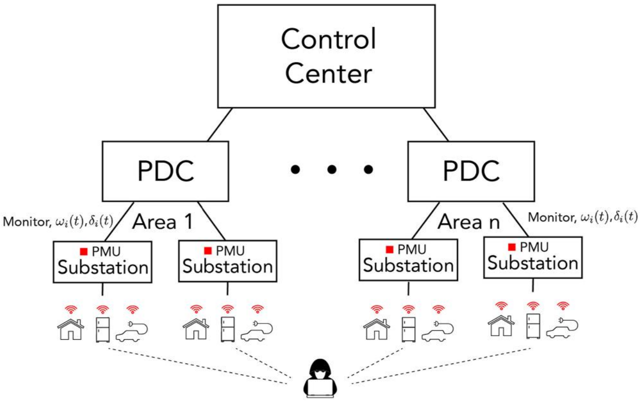


FIGURE 2 Proposed detection framework and integration into wide-area monitoring systems

on any prior knowledge or the laws of physics. Specifically, one can train a machine learning classifier that distinguishes between the system's dynamics under different values of the attack parameters $\left\{K^L_{i,k_j}\right\}_{i\in\mathcal{V},k_j\in\mathcal{S}}, \epsilon^L_i, i\in\mathcal{V}$. However, under such an approach, identifying the locations and the magnitude of LAA will be challenging since it will involve a combinatorial search across nodes of the grid that the attacker may target (note that this number can potentially be $2^{|\mathcal{V}|\times|\mathcal{S}|}$). Training such a classifier would be computationally complex, which requires a tremendous amount of training data.

On the other hand, if the physical model governing the system dynamics is known, this knowledge can be incorporated into the learning. This approach is beneficial when the inference must be performed with a limited number of samples, which is the case in our problem, where the identification must be completed quickly before the system is damaged. Motivated by these challenges, we propose two state-of-the-art techniques for data-driven identification of dynamical systems to solve the LAA detection/identification problem, namely, the SR algorithm and the PINN algorithm. The details are presented next.

3.1 | LAA detection and identification using SR approach

The SR approach, which is based on the SINDy algorithm proposed in [22], applies sparse regression to find the attack parameters that best represent the observed power grid dynamics. To formalise the SR framework for detecting and identifying LAAs, we define the following additional notations. First, we define $\dot\theta_i$ as

$$\dot\theta_i^{(\tau)} = \dot\delta_i^{(\tau)} + \frac{1}{D_i}\left(P_i^{LS} + \sum_{j\in\mathcal{N}} B_{i,j}\sin\left(\delta_{ij}^{(\tau)}\right)\right). \quad (9)$$

Using (9), a discretised version of the load bus dynamics in (8) over $\tau = 1, 2, \ldots, T$ time slots can be rewritten as

$$\underbrace{\begin{bmatrix}\dot\theta_i^{(1)}\\ \vdots\\ \dot\theta_i^{(T)}\end{bmatrix}}_{\dot{\boldsymbol\theta}_i} = \frac{1}{D_i}\underbrace{\begin{bmatrix}\omega_{k_1}^{(1)},\ldots,\omega_{k_S}^{(1)},-1\\ \vdots\\ \omega_{k_1}^{(T)},\ldots,\omega_{k_S}^{(T)},-1\end{bmatrix}}_{\Omega}\underbrace{\begin{bmatrix}K^L_{i,k_1}\\ \vdots\\ K^L_{i,k_S}\\ \epsilon_i\end{bmatrix}}_{\mathbf{k}_i^L}, i\in\mathcal{N}_L. \quad (10)$$

Equation (10) represents the relationship between the attack parameters $\mathbf{k}_i^L$ and the load bus dynamics at bus $i\in\mathcal{N}_L$. Note that the system operator can compute the elements of $\dot{\boldsymbol\theta}_i$ and Ω using the frequency and phase angle measurements obtained from PMU. Thus, the system operator can estimate the attack parameters $\mathbf{k}_i^L$ by solving (10).

At this stage, several comments are in order. First, Equation (10) is a system of T linear equations with $S+1$ unknown variables. In general $T \gg S+1$, since PMUs have a high sampling frequency (e.g. 50 measurement samples per second [33]). Second, the measurements in $\dot{\boldsymbol\theta}_i$ and Ω are corrupted by measurement noises. Third, the value of $\mathbf{k}_i^L$ that solves (10) may not be unique. For instance, there may be multiple attack vectors that result in similar dynamics. Thus, we seek to find the sparsest attack vector $\mathbf{k}_i^L$ that best explains the dynamics observed in $\dot{\boldsymbol\theta}_i$ and Ω. The rationale is that in a real-world scenario, the attacker can likely monitor the frequency fluctuations and manipulate the load at only a few nodes in the power grid. To promote sparsity, we use the LASSO algorithm, which applies an L1-regularisation term to promote sparsity. The optimization problem corresponding to the LASSO algorithm is given by [35]

$$\widehat{\mathbf{k}}_i^L = \arg\min_{\mathbf{k}_i^L} \|\Omega\mathbf{k}_i^L - \dot{\boldsymbol\theta}_i\|^2 + \lambda\|\mathbf{k}_i^L\|_1, \; i = 1\ldots,|\mathcal{V}|. \quad (11)$$

Herein, $\|\mathbf{k}_i^L\|_1$ represents a penalty term that and $\lambda > 0$ is a scaling parameter. The overall SR approach is described in Algorithm 1.

Algorithm 1 Sparse Regression Algorithm

```
Input: Phase angle data $\left\{\delta_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\ldots,T}$ and
frequency data $\left\{\omega_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\ldots,T}$
Output: Estimates of the static
($\widehat{\epsilon}_i^L, i = 1,\ldots,N_L$) and dynamic
($\widehat{K}_{i,k_j}^L, i = 1,\ldots,N_L, j = 1,\ldots,S$) LAA parameters
Monitor phase angle $\left\{\delta_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\ldots,T}$ and
  frequency $\left\{\omega_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\ldots,T}$ data.
From the phase angle data and frequency
  data, compute $\left\{\dot\theta_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\ldots,T}$ as in (9).
Construct the vector and the vector $\dot{\boldsymbol\theta}$ and
  the matrix $\Omega$ from the phase angle and
  frequency measurements.
```

```
Using $\dot{\boldsymbol{\theta}}$ and $\Omega$, solve for $\widehat{\mathbf{k}}_i^L$ using the LASSO
  algorithm.
Output $\widehat{\mathbf{k}}_i^L$, $i = 1, ..., |\mathcal{N}_L|$.
```

3.2 | Identification based on physics-informed neural networks

We now describe the PINN framework [23, 36] applied to detecting/identifying LAAs. PINN incorporates the knowledge of the physical model guiding the observed dynamics to train a neural network with limited training samples. As in [23], we incorporate the underlying differential equations as an additional loss function to solve two different types of problems. The first problem is the forward problem, which uses NN to obtain a solution to the differential equations, that is, in our context, the signals $\delta_i(t)$ and $\omega_i(t)$ that solve (6)–(8) under LAAs. The second problem that is of more interest to our work is the inverse problem, in which, given the measurements and the structure of the differential equations, estimate the unknown parameters $\left\{K^L_{i,k_j}\right\}_{i\in\mathcal{V},k_j\in\mathcal{S}}$ and $\epsilon_i^L, i\in\mathcal{V}$ that best describe the observed data.

The PINN framework accomplishes these two tasks by defining appropriate loss functions and training a NN to minimise them. The overall PINN framework is shown in Figure 3 and we present the details next. First, let $\widehat{\omega}_i(t,\phi)$ and $\widehat{\delta}_i(t,\phi)$ be the NN approximations of $\omega_i(t)$ and $\delta_i(t)$ as shown in Figure 3. Herein, ϕ represents the weights of the NN, which are trained to minimise the following losses:

3.2.1 | Mean-square loss

The first loss term involves the mean square loss between the observed measurements $\omega_i(t)$ and $\delta_i(t)$ and their NN approximations, that is,

$$L_1(\phi) = \frac{1}{T}\sum_{\tau=1}^{T}\left(\sum_{i\in\mathcal{N}_G}(\widehat{\omega}_i(\tau,\phi)-\omega_i(\tau))^2 + \sum_{i\in\mathcal{N}}\left(\widehat{\delta}_i(\tau,\phi)-\delta_i(\tau)\right)^2\right). \tag{12}$$

3.2.2 | Physics based loss

The system dynamics are incorporated as a second loss function. Specifically, let us define

$$f_i^{(1)} = \dot{\widehat{\delta}}_i - \widehat{\omega}_i, i\in\mathcal{N}_G \tag{13}$$

$$f_i^{(2)} = M_i\dot{\widehat{\omega}}_i + D_i\widehat{\omega}_i + K_i^P\widehat{\omega}_i + K_i^I\widehat{\delta}_i + \sum_{j\in\mathcal{N}} B_{i,j}\sin\left(\widehat{\delta}_{ij}\right),\ i\in\mathcal{N}_G, \tag{14}$$

$$f_i^{(3)} = D_i\dot{\widehat{\delta}}_i - \sum_{j=1}^{S}\widehat{K}^L_{i,k_j}\widehat{\omega}_{k_j} + \widehat{\epsilon}_i^L + P_i^{LS} + \sum_{j\in\mathcal{N}} B_{i,j}\sin\left(\widehat{\delta}_{ij}\right),\ i\in\mathcal{N}_L, \tag{15}$$

where in the above, we have dropped the notations showing the dependency of $f_i^{(1)}$ and $f_i^{(2)}$ on (t,ϕ) and $f_i^{(3)}$ on $(t,\phi,\mathbf{K}^L,\boldsymbol{\epsilon}^L)$ and simplicity. In the above, the derivatives are computed using NN's automatic differentiation. The physics-based loss function is then given by

$$L_2\quad\left(\phi,\mathbf{K}^L,\boldsymbol{\epsilon}^L\right) = \frac{1}{T}\sum_{\tau=1}^{T}\left(\sum_{i\in\mathcal{N}_G}\left(f_i^{(1)}(\tau,\phi)\right)^2 + \sum_{i\in\mathcal{N}_G} f_i^{(2)}(\tau,\phi)^2 + \sum_{i\in\mathcal{N}_L}\left(f_i^{(3)}\left(\tau,\phi,\mathbf{K}^L,\boldsymbol{\epsilon}^L\right)^2\right)\right). \tag{16}$$

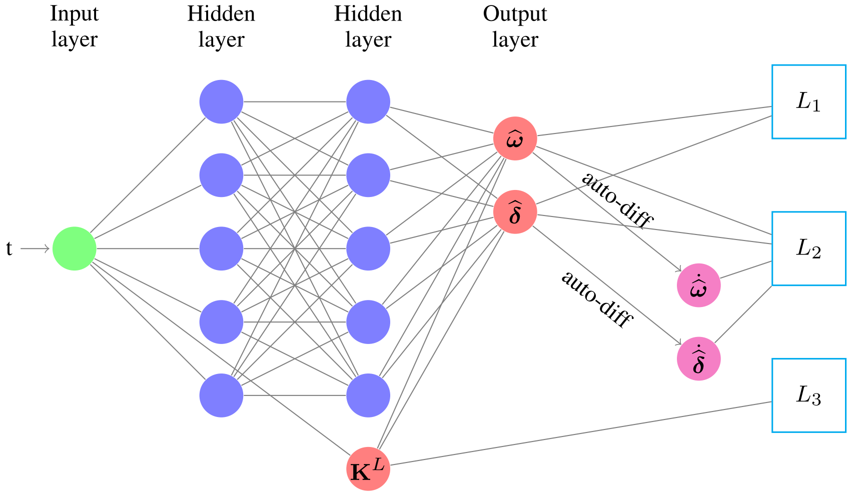


FIGURE 3 Physics-informed neural networks network showing input output layers and the output attack parameters

Note that the above loss function depends both on the NN weights $\boldsymbol{\phi}$ and the attack parameters $\mathbf{K}^L$, $\boldsymbol{\epsilon}^L$.

3.2.3 | Sparsity-promoting loss

As noted before, we are interested in sparse solutions to the attack parameter estimation problem. Thus, in the PINN framework, we also add the sparsity promoting term in the loss function given by

$$L_3\left(\mathbf{K}^L\right) = \alpha\left(\|\widehat{\mathbf{K}}^L\|_1 + \|\widehat{\boldsymbol{\epsilon}}^L\|_1\right), \tag{17}$$

where, α is the scaling factor and $\|\widehat{\mathbf{K}}^L\|_1$ and $\|\widehat{\boldsymbol{\epsilon}}^L\|_1$ are the 1-norm of the vectors consisting of all the attack parameters.

The NN is trained using the observed measurements $\left\{\delta_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\dots,T}$ and $\left\{\dot{\delta}_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\dots,T}$ as follows:

$$\boldsymbol{\phi}^*, \widehat{\mathbf{K}}^L, \widehat{\boldsymbol{\epsilon}}^L = \arg\min_{\boldsymbol{\phi},\mathbf{K}^L,\boldsymbol{\epsilon}^L} L_1(\boldsymbol{\phi}) + L_2\left(\boldsymbol{\phi}, \mathbf{K}^L, \boldsymbol{\epsilon}^L\right) + L_3\left(\mathbf{K}^L\right).$$

Note that when the sum of these two losses is minimised, we ensure three criteria: (i) the output of the NN replicates the observed system dynamics, (ii) $f_i^{(1)}, f_i^{(2)}$, and $f_i^{(3)}$ are close to zero, which in turn implies that 6 7–8 are satisfied, and (iii) the obtained solution is sparse. This implies the estimated attack parameters that best fit the observed data. The overall PINN approach is described in Algorithm 2.

3.3 | Comparison with black-box machine learning algorithms

As noted before, the SR and PINN algorithms are examples of the so-called 'physics-informed' ML algorithms. Alternately, one may also use 'black-box' ML algorithms (e.g. support vector machines or neural networks) that are purely data-driven and do not use the knowledge of the underlying physical models. We now clarify the difference between the two approaches in the context of detection and identification of IoT-enabled LAAs.

Note that both the SR and the PINN algorithms do not require 'offline' training. The data monitored in real-time can be directly fed as inputs to Algorithms 1 and 2 to generate estimates of the attack parameters. In contrast, the black-box ML algorithms require offline training, which can be very data-intensive, especially for the considered case of IoT-based LAAs. In particular, we must train the ML model with training data corresponding to LAAs at different nodes (note that there can be $2^{|\mathcal{V}|}$ combinations of victim nodes, considering multi-point attacks). Moreover, the attack parameters at the victim nodes can also have different magnitudes. Thus, training ML models considering many combinations will be computationally complex and will require a tremendous amount of training data. In Section 5.4, we provide a numerical comparison of the proposed algorithms with SVM and NN trained using black-box training techniques. We observe that the accuracy of training depends heavily on the range and the amount of the training data. To find more details about SVM and NN, see [37].

Algorithm 2 PINN Algorithm

```
Input: Phase angle data $\left\{\delta_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\dots,T}$ and frequency data $\left\{\omega_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\dots,T}$
Output: Estimates of the static $\left(\widehat{\in}_i^L, i=1,\dots,N_L\right)$ and dynamic $\left(\widehat{K}_{i,k_j}^L, i=1,\dots,N_L, j=1,\dots,S\right)$ attack parameters
Parameters: Neural Network Parameters $\boldsymbol{\phi}$
Setup neural network with parameter $\Theta$ and unknown attack parameters
    $\left(K_{i,k_j}^L, i=1,\dots,N_L, j=1,\dots,S\right)$ as TensorFlow variables as shown in Figure 3.
Use auto-gradient to calculate the derivatives of the phase angle and frequency data.
Setup loss functions $L_1(\boldsymbol{\phi})$, $L_2(\boldsymbol{\phi}, \mathbf{K}^L, \in^L)$ and $L_3(\mathbf{K}^L)$ defined in Sections 3.2.1 to 3.2.3.
Monitor phase angle $\left\{\delta_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\dots,T}$ and
    frequency data $\left\{\omega_i^{(\tau)}\right\}_{i\in\mathcal{N},\tau=1,\dots,T}$.
Use the collected data to train the neural network.
Output the attack parameters
    $\widehat{K}_{i,k_j}^L, i=1,\dots,N_L, j=1,\dots,S$.
```

4 | DISCUSSION AND IMPLEMENTATION

In this section, we present a discussion on some practical aspects of the proposed algorithms.

4.1 | Differentiating attacks from natural load fluctuations

Note that the power system is naturally dynamic, that is, the frequency/phase angle fluctuations occur due to natural load changes. Thus, the grid operator will need a mechanism to differentiate these natural variations from LAAs. There are two important differences between natural load fluctuations and an LAA: (i) for static LAAs, ϵ_i^L will have a large value (to cause unsafe frequency excursions, please see Figure 1a) (ii) for dynamic LAAs, the attack controller gain parameter $K_{i,j}^L$ will be non-zero and positive. Note that this is designed to work in an opposite manner to the generator's governor control (that arrests deviations), that is, the dynamic LAA decreases the load when the frequency is increasing (about the setpoint), and LAA increases the load when the frequency is decreasing (below the setpoint) [16].

In general, the operator can adopt the following procedure to differentiate natural load fluctuations to attacks. (i) Implement the SR or PINN algorithm periodically to infer ϵ_i^L and $K_{i,j}^L$ values from the observed data. (ii) If $\widehat{\epsilon}_i^L \neq 0$ and $\widehat{K}_{i,4j}^L = 0$, then, compare $\widehat{\epsilon}_i^L$ against a threshold value, which is computed based on the historical or forecasted value of the load fluctuations. If it exceeds the threshold, raise an attack alert. (iii) If $\widehat{K}_{i,j}^L > 0$, compare against a threshold (around zero). If it exceeds the threshold, raise an attack alert.

4.2 | Implementing the methods

This section highlights the implementation aspects of the SR and PINN algorithms. The proposed algorithms can be straightforwardly integrated into existing wide-area monitoring systems (WAMS). The hierarchical structure of the WAMS network is shown in Figure 2. PMUs installed at the substation monitor the phase angle/frequency measurements and send these measurements to a phasor data concentrator (PDC) and subsequently to a control centre (CC). The proposed algorithms can be implemented in a centralised (at the CC) or decentralised manner. Since they are not significantly resource-intensive, both the SR and PINN algorithms can be implemented at the edge, either at a local phasor data concentrator (PDC) or at individual substations with limited information exchange between the nodes. This allows efficient response and detection, more control over data flows, and therefore enhanced security, privacy and data handling. We elaborate the method in the following.

Recall that the SR algorithm involves solving (11) for every victim node $i \in \mathcal{V}$ to determine $\mathbf{k}_i$. This in turn requires the following signals at each node $i \in \mathcal{V}$: (i) δ_i (ii) $\left\{\delta_j\right\}_{j \in \mathcal{N}_i}$, and (iii) $\left\{\omega_{k_j}\right\}_{k_j \in \mathcal{S}}$. Note that (i) can be monitored locally at individual substations, and (ii) only requires information exchange with the neighbouring nodes of node i, that is, $j \in \mathcal{N}_i$. On the other hand, (iii) requires information exchange from all the potential nodes at which the attacker can sense the grid frequency. In practice, due to the attacker's limited capabilities, these nodes are likely to be restricted to nodes that belong to the same control area or those that are connected to the same PDC. Thus, the SR algorithm can be implemented locally with limited information exchange.

The PINN algorithm can be also be implemented in a decentralised manner. In particular, each substation can locally solve the following optimization problem to determine $\mathbf{k}_i^L$.

$$\phi_i^*, \widehat{\mathbf{k}}_i^L = \arg\min_{\phi_i, \mathbf{k}_i^L} L_1(\phi_i) + L_2\left(\phi_i, \mathbf{k}_i^L\right) + L_3\left(\mathbf{k}_i^L\right), \tag{18}$$

where $L_1(\phi) = \frac{1}{T} \sum_{\tau=1}^{T} \left(\sum_{j \in \{i \cup \mathcal{N}_i\}} \left(\widehat{\omega}_i(\tau, \phi) - \omega_i(m)\right)^2 \right.$

$$\left. + \sum_{j \in \{i \cup \mathcal{N}_i\}} \left(\widehat{\delta}_i(\tau, \phi) - \delta_i(m)\right)^2 \right),$$

$$L_2(\phi) = \left(f_i^{(3)}\left(\phi_i, \mathbf{k}_i^L\right)\right)^2, L_3(\phi) = \alpha \|\mathbf{k}_i^L\|_1.$$

As noted in the case of the SR algorithm, all the signals required to solve (18) can be obtained locally with limited information exchange between nodes of the same control area.

5 | SIMULATIONS

In this section, we present simulations to show the effectiveness of the proposed approaches.

5.1 | Simulation settings and methodology

5.1.1 | Bus systems

The proposed algorithms are tested using the IEEE 6-, 14-, and 39-bus systems. The power grid topological data is obtained from the MATPOWER simulator. For the IEEE 39-bus system, $\mathcal{N}_G = \{30, \ldots, 39\}, |\mathcal{N}_G| = 10$ and $\mathcal{N}_L = \{1, \ldots, 29\}, |\mathcal{N}_L| = 29$. We assume $\mathcal{S} = \mathcal{N}_G$ and $\mathcal{V} = \mathcal{N}_L$. Thus, in our simulation setup, the attack parameter identification involves estimating $|\mathcal{N}_G| \times |\mathcal{N}_L| = 290$ values of $K_{v,s}^L, v \in \mathcal{V}, s \in \mathcal{S}$. For the IEEE 14-bus system, $\mathcal{N}_G = \{1, 2, 3, 6, 8\}, |\mathcal{N}_G| = 5$ and $|\mathcal{N}_L| = 9$. Finally, for the IEEE 6-bus system, $\mathcal{N}_G = \{1, 2, 3\}, |\mathcal{N}_G| = 3$ and $|\mathcal{N}_L| = 3$. More details on the simulation parameters used in the case studies are listed in the Appendix.

5.1.2 | Algorithm implementation and settings

We implement 5 algorithms in total—(i) SR (ii) PINN (iii) SVM (iv) NN, and (v) UKF. The details are as follows: (i) The SR algorithm is implemented using the *Lasso* function in MATLAB. (ii) The PINN algorithm is implemented using Tensorflow. We use a multi-layer perceptron for PINN with three hidden layers, each with 50 neurons. We use automatic differentiation to calculate the gradients. The NN is trained using the limited memory Broyden–Fletcher–Goldfarb–Shanno algorithm (LBFGS) optimizer (similar to [23]), as it is better suited when the training samples are limited. (iii) SVM is implemented in MATLAB using Gaussian kernel function. (iv) NN is also implemented in MATLAB. For NN, three multi-layer perceptrons with 10 neurons for each layer is considered. To avoid over-fitting problems, the cross-validation technique and the ReLU activation function are also implemented. (v) Finally, UKF is implemented using *unscentedKalmanFilter* function in MATLAB.

5.1.3 | Generating training/testing data

In order to generate the dataset corresponding to power grid signals under LAA, we simulate the differential Equations in (6)–(8) using the *ode-45* function of MATLAB.

For brevity, we only present details from the IEEE-39 bus system (the settings for the other bus system can be found in the Appendix). In this bus system, the node pair $(v, s) = (19, 33)$ corresponds to the least-effort LAA [17]. Thus, for single-point attacks, we set $K^L_{19,33}$ to a non-zero value (to be specified subsequently) and rest of the values of $K^L_{v,s}$ to be zero. We consider two sets of generation parameters for each test case as listed in the Appendix. The frequency/phase angle dynamics of the IEEE 39-bus system corresponding to the two sets of parameters are shown in Figure 4. We refer to the parameter set A1 as *fast dynamics*, in which the frequency deviates away from the setpoint of 50 Hz relatively quickly within a few oscillation cycles. It can be observed that the oscillations are relatively smooth. The attack parameter in this case is set to $K^L_{19,33} = 18$. We refer to parameters set B1 as *slow oscillatory dynamics*, in which we observe several cycles of oscillations before the frequency deviates away from the setpoint of 50 Hz. The attack parameter in this case is set to $K^L_{19,33} = 25$. For both cases, we set $\epsilon_{19} = 0.1$ pu. To generate noisy data (as inputs to the algorithms), we add Gaussian noise to the frequency and the phase angle waveforms (obtained by solving (1)–(3)), whose standard deviation is set to 0.01 pu. We assume PMU sampling frequency of 50 measurements per second. Hence, the number of measurements involved in the training process is 50 T, where T is the measurement time window (specified subsequently).

As explained in Section 3.3, SR and PINN are physics-informed ML algorithms that do not require 'offline training'. Thus, the data generated using the procedure stated above can be directly fed as inputs to the algorithm for inferring the attack parameters. The SVM and NN approaches on the other hand are 'black-box' ML algorithms that require 'offline training. In the offline data generation procedure, 2000 samples, based on two features, δ_i and ω_i, $i \in \mathcal{N}_G$, with 100 time-steps, are generated for each victim bus $\mathcal{N}_v(\subseteq \mathcal{N}_L)$. To incorporate diversity in the offline-training data, both the location of the sensing bus/victim bus $\mathcal{V}, \mathcal{S}(\subseteq \mathcal{N})$ as well as the dynamic attack parameter $K^L_{v,s}$ are varied between a certain range. As noted in Section 3.3, the range of $K^L_{v,s}$ considered in the offline training data is critical to the performance of SVM and NN. To illustrate these effects, we consider two different ranges, (i) (1.2–23.4) and (ii) (0.8–32.4) and discuss the dependence of this parameter in Section 5.4.

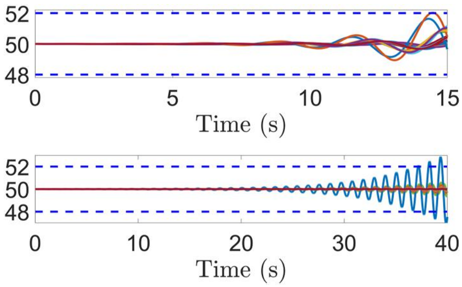


FIGURE 4 Frequency dynamics under DLAAs for the IEEE-39 bus system. Top figure: Fast dynamics (System parameters A) with $K^L_{19,33} = 18$. Bottom figure: Slow oscillatory dynamics (System parameters B) with $K^L_{19,33} = 25$. The horizontal lines indicate the safety limit of 2 Hz

5.1.4 | Algorithm evaluation metrics

Attack detection and identification algorithms in the context of LAAs must have two characteristics (i) the targeted node(s) and the corresponding attack parameter(s) must be identified correctly, and (ii) for the nodes that are not targeted by the attacker, the estimated value of must be close to zero. Since our objective is not only finding the attack locations, but also determining the attack parameters, we evaluate the algorithm performance in terms of two metrics. The first, given by

$$\eta_1 = \left(K^L_{v,s} - \widehat{K}^L_{v,s}\right)/K^L_{v,s},$$

where v and s are the actual victim and sensing buses respectively (at which the attack is injected to generate the frequency and phase angle dynamics). The second metric is given by

$$\eta_2 = \frac{1}{|\mathcal{V}||\mathcal{S}|}\sum_{v\in\mathcal{V}, s\in\mathcal{S}}\left(K^L_{v,s} - \widehat{K}^L_{v,s}\right)^2,$$

that is, the mean-square error of the estimated attack parameters at nodes that are not targeted by the attacker. Note that a low value of η_1 indicates high precision in estimating the correct attack parameters (true positive), where as a high value indicates that the attack is not detected accurately (false negative). Similarly, a low value of η_2 indicates that the nodes not targeted by the attacker are not identified as attacked nodes (true negative), where as a high value indicates large number of false positives.

5.2 | Simulation results

We present the simulation results next. Once again, for brevity, the simulation results from the IEEE-39 bus system are presented as plots and other results from other bus systems are presented in a tabular form.

5.2.1 | Single-point attacks

We examine the effectiveness of the SR and PINN methods in detecting the DLAAs and compare their performance with the SVM, NN and UKF methods.

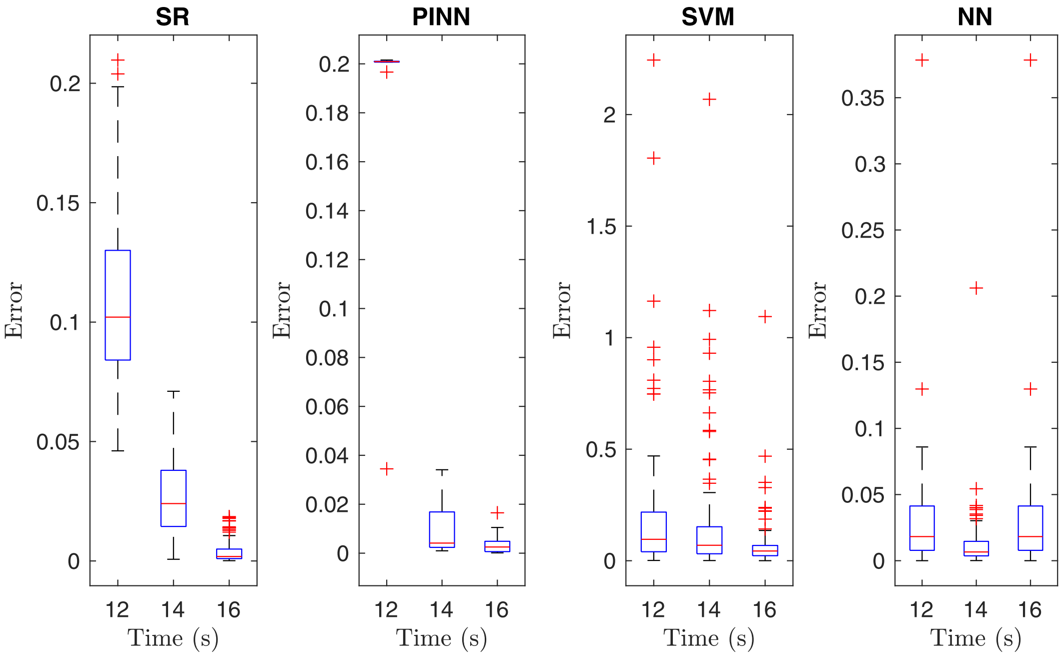


FIGURE 5 Attack parameter estimation error η_1 (boxplots) for fast dynamics ($12-16$ s, System parameters A) with $K^L_{19,33}=18$

The parameter estimation accuracy η_1 (boxplots) for the IEEE-39 bus system with fast dynamic parameters is shown in Figure 5 for the SR, PINN, SVM, and NN algorithms, respectively. The results are plotted for different observation time windows. The attack parameter estimation is repeated 100 times, where the measurements differ due to the noise realization. Furthermore, for the range of the dynamic attack parameter in the training data for SVM and NN algorithms, we consider (i) (1.2–23.4) (see Section 5.4 for more discussion on this).

Fast Dynamics: It can be observed that the SR method can estimate the dynamic attack parameters more accurately compared to PINN, SVM, and NN for fast dynamics before the system dynamics breach the safety limits. As expected, the attack parameters can be estimated with greater accuracy when the observation time window is larger, due to the availability of more data.

Slow Oscillatory Dynamics: The SR algorithm achieves high precision for slow-oscillatory dynamics as well. However, we observed that other benchmark algorithms have a large error in inferring the attack parameters in this simulation setting. Accordingly, for this simulation setup, only SR results are presented in Figure 6, and other benchmarks, which have more than 50% error are neglected.

To further investigate the poor performance PINN, SVM and NN algorithms for slow oscillatory dynamics, we investigate their online training procedure. For brevity, we only present the result for PINN. As illustrated in Figure 7, which presents the overall loss as a function of the training epochs, while in the case of fast dynamics, the error is minimized over several training epochs (47,000 epochs) and reaches a very low value, for slow oscillatory dynamics, the training terminates relatively quickly (640 epochs) since the loss does not decrease further. We conjecture this is because the system with slow dynamics has several oscillatory cycles (see Figure 4) and the training process may be potentially stuck at local minima. In contrast, for the system with fast dynamics, the oscillatory cycles are relatively smooth and conducive to training the NN. Thus, the PINN algorithm is not suited for attack detection/identification for systems with slow dynamics.

Further, we also enlist the overall mean-square error of attack parameter estimation (η_2) in Table 1. A similar trend as the previous result is observed, that is, while the SR, PINN, SVM, and NN algorithms achieve good performance for the system with fast dynamics. However, for slow dynamics, only the SR algorithm achieves satisfactory performance. In particular, the low value of η_2 indicates that the value of $K^L_{v,s}$ for nodes that are not targeted by the attacker are close to zero. The error for the PINN, SVM and NN algorithms under slow oscillatory dynamics is very high; hence we do not include them in Table 1.

Finally, we also compare the UKF approach for attack parameter estimation in Table 1 (last column). It can be observed that while UKF method performs well in the case of slow dynamics, it is unable to do so in the case of fast dynamics. This is because UKF is a sequential algorithm, and hence, it is unable to capture the effect of rapid fluctuations in the system dynamics over a relatively short period of time. The poor performance of UKF can also be observed by noting the value of η_1 in Table 2 (IEEE-39 bus, corresponding to fast dynamics), where we observe an error of 48% in estimating the value of the attack parameter.

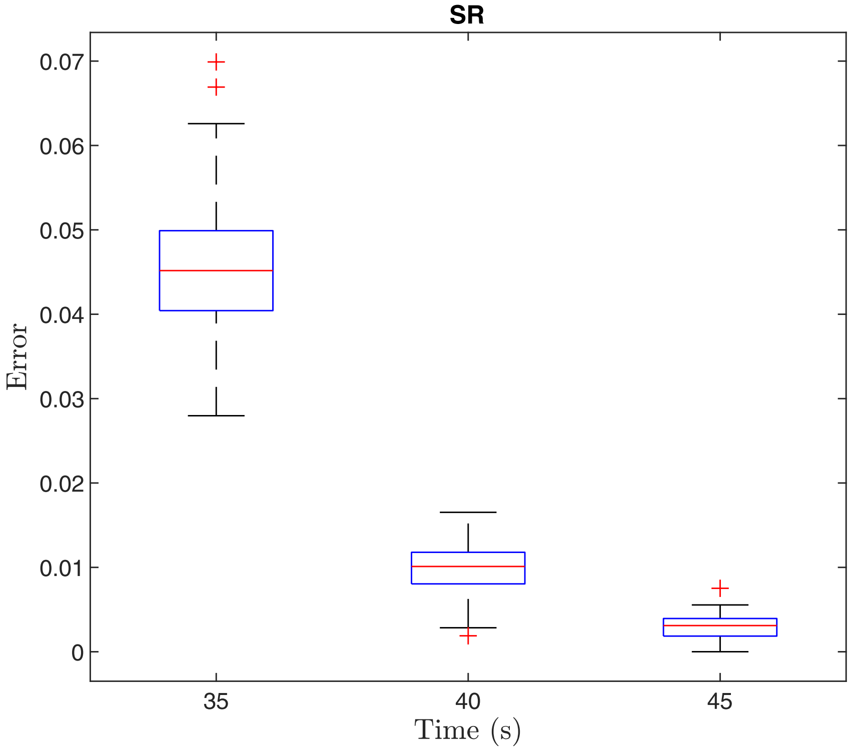


FIGURE 6 Attack parameter estimation error η_1 (boxplots) for slow oscillatory dynamics (30 − 45 s, System parameters (b) with $K^L_{19,33} = 25$

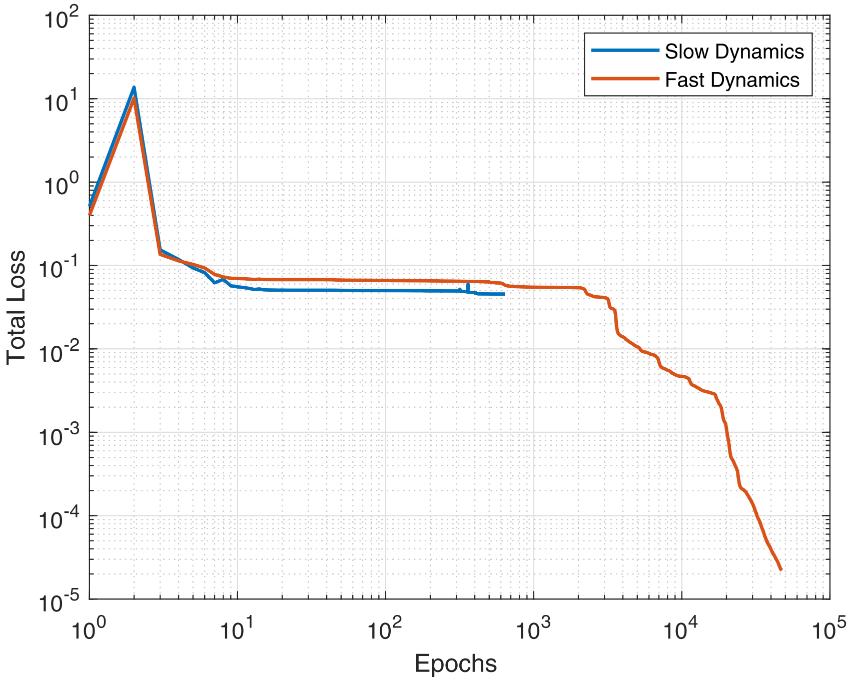


FIGURE 7 Physics-informed neural networks training for the slow and fast dynamics cases showing loss over the training epochs. Both X and Y axes are in the logarithmic scale

5.3 | Simulation results for different bus systems

To verify the effectiveness of the proposed methods in different bus systems, the results from IEEE 6-, 14-, 39-bus systems are presented for fast dynamic parameters. The dynamic attack parameters are considered as $K^L_{4,1} = 6.8$, $K^L_{5,1} = 11.3$ and $K^L_{19,33} = 18$, for IEEE 6-,14-, 39- bus systems, respectively. The numerical results, which present the η_1 and η_2 values, are shown in Table 2. The findings confirm the effectiveness of the SR method against other benchmarks in all cases. As shown in Table 2, the η_1 and η_2 values for the SR method are in range (0.028–0.060) and (0.011–0.038), respectively, while larger error values are observed for other benchmark algorithms.

5.4 | Multi-point attacks

We also consider simulations with multi-point attacks (multiple victim buses). We only present results for the case of fast dynamics (due to the difficulties in training the other algorithms for slow-oscillatory dynamics, as mentioned in Section 5.2). The attack parameters in this case are given by $K^L_{15,33} = 4, K^L_{19,33} = 14, K^L_{20,33} = 4$. Once again, the parameters are chosen to represent a system that is stable without attacks, but unstable due to the LAAs. The accuracy results for the different attack parameters are plotted in Figure 8 considering an observation time widow of $T = 16$ s. We observe that both SR and PINN algorithms produce reliable estimates within the considered observation time window.

TABLE 1 Mean-square error of attack parameter estimation (η_2) for IEEE-39 bus system with single-point attack. $K^L_{19,33} = 18$ for fast dynamics (Parameters A) with ($T = 15$ s) and $K^L_{19,33} = 25$ for slow oscillatory dynamics (Parameters B) with ($T = 40$ s)

	Metric	SR	PINN	SVM	NN	UKF
Fast dynamics	η_2	0.011	0.08	1.350	0.097	0.44
Slow oscillatory dynamics	η_2	0.1	–	–	–	0.08

Abbreviations: NN, neural networks; PINN, physics-informed neural networks; SR, Sparse Regression; SVM, support vector machines; UCF, unscented Kalman filter.

TABLE 2 Simulation results for different case studies under single-point attacks for fast dynamic parameters

	Metric	SR	PINN	SVM	NN	UKF
IEEE 6-Bus system	η_1	0.028	0.07	0.219	0.097	0.47
	η_2	0.038	0.092	1.178	0.808	2.13
IEEE 14-Bus system	η_1	0.06	0.064	0.168	0.103	0.49
	η_2	0.0151	0.075	2.355	0.962	3.18
IEEE 39-Bus system	η_1	0.04	0.08	0.073	0.086	0.48
	η_2	0.011	0.085	1.350	0.097	0.44

Abbreviations: NN, neural networks; PINN, physics-informed neural networks; SR, Sparse Regression; SVM, support vector machines; UCF, unscented Kalman filter.

As stated in Section 3.3, the performance of the black-box machine learning algorithm such as SVM and NN relies on the range of the training data considered during the offline training phase. The simulation results also investigate this claim using two different case studies with different dynamic attack parameter ranges (i) (1.2–23.4) and (ii) (0.8–32.4). The outcome of the simulations is presented in Figures 9 and 10. The results illustrate that by increasing the range of the dynamic attack parameters considered in the training data (Figure 10), the prediction accuracy decreases. This is expected since a lower range (of the attack parameters) in the training data set implies more points around the actual attack parameter used by the attacker, and hence, better prediction accuracy (conversely, lower accuracy for higher range). However, the operator has no way of having prior knowledge of the actual attack parameter that will be used by the attacker. Thus, they must consider a large range of the attack parameters in the training data set (otherwise, they may risk misclassifying the attack completely). Thus, black-box machine learning methods have a high dependency on the input data structure. However, in the proposed methods in this study, especially in SR, we do not have these of limitations, which confirms that SR can be a good solution for a wide range of system parameters.

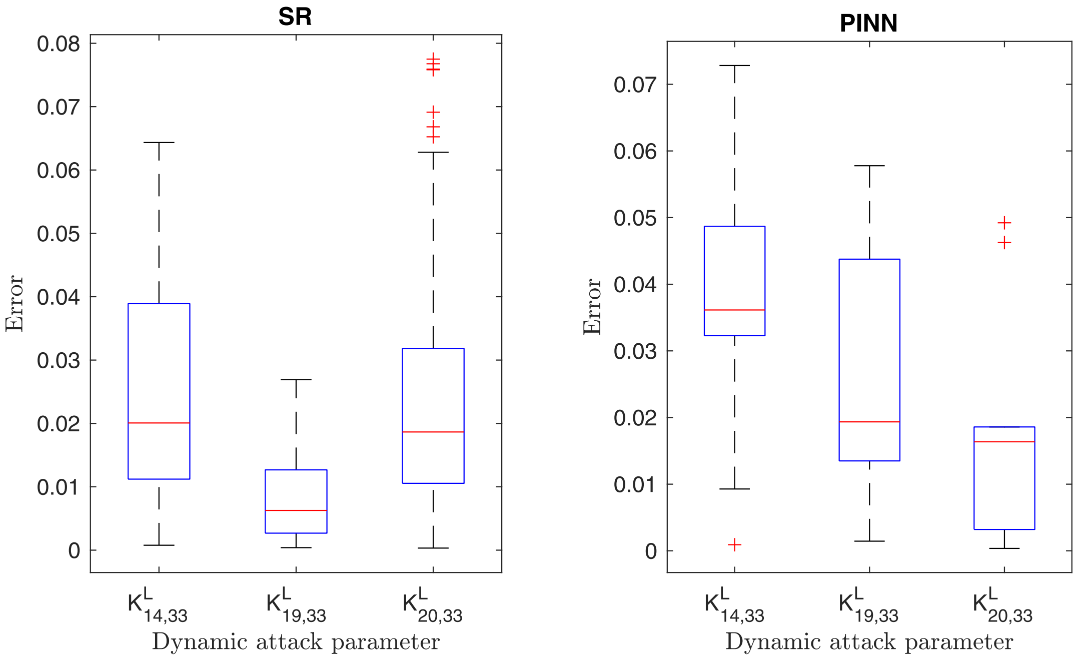


FIGURE 8 Attack parameter estimation accuracy η_1 (boxplots) for multi-point attacks with $T = 16$ s. Attack parameters: $K^L_{14,33} = 4, K^L_{19,33} = 18, K^L_{20,33} = 4$

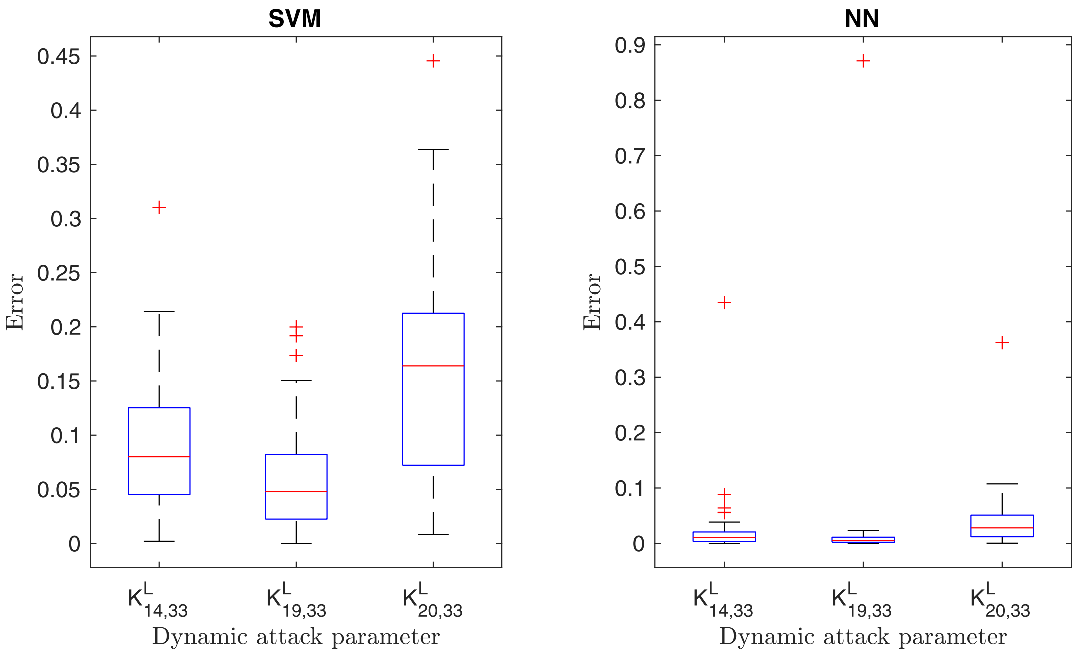


FIGURE 9 Attack parameter estimation error η_1 (boxplots) for multi-point attacks with $T = 16$ s for $K^L_{14,33} = 4, K^L_{19,33} = 18, K^L_{20,33} = 4$. Attack parameters range considered in the offline-training procedure is (1.2–23.4)

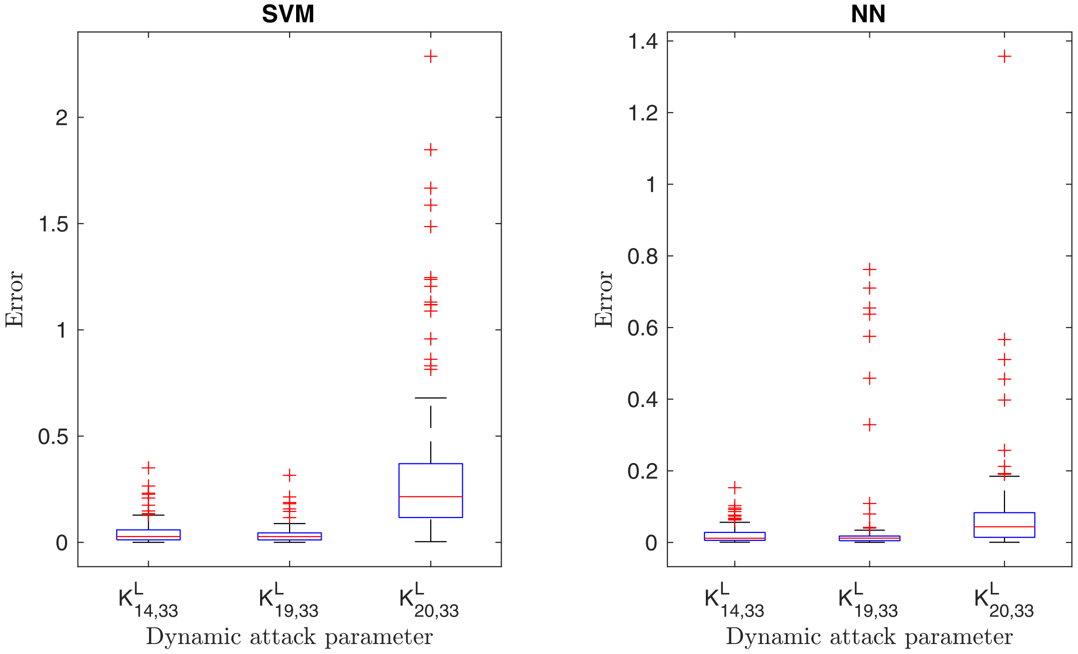


FIGURE 10 Attack parameter estimation error η_1 (boxplots) for multi-point attacks with $T = 16$ s for $K^L_{14,33} = 4, K^L_{19,33} = 18, K^L_{20,33} = 4$. Attack parameters range considered in the offline-training procedure is (0.8–32.4)

5.5 | Execution times

Finally, we also enlist the execution times of the SR, PINN, SVM, and NN algorithms. The simulations are conducted on a Windows PC with Intel Xeon(R) CPU E5-2630 v3 @ 2.40 GHz processor, RAM:24 GB GPU NVIDIA K80. For the PINN algorithm, we consider two training regimes. First, train from scratch, in which the NN weights are randomly initialised and trained with the observed data. In the second method, the NN weights are pre-trained with the phase angle/frequency measurements with no attacks. Then during the online phase, the NN weights are initialised with the pre-trained weights and fine-tuned with the real-time measurements (with attack). The results are tabulated in Table 3 averaged over 100 runs. It can be noted that the SR algorithm takes significantly less time to execute compared to the PINN, algorithm. This is because the PINN involves training a NN over several epochs, which is computationally complex. The pre-training reduces the computational

TABLE 3 Execution times for the SR and PINN algorithms

Method	Execution time (s)
SR	4.84
PINN	4071
SVM	27.13
NN	69.15

Abbreviations: NN, neural networks; PINN, physics-informed neural networks; SR, Sparse Regression; SVM, support vector machines; UCF, unscented Kalman filter.

time of the PINN algorithm. On the other hand, SR follows the LASSO method which is computationally much simpler. Thus, in addition to being broadly applicable over a wide range of system parameters, the SR method also has an additional advantage of quick execution in comparison with PINN, SVM, and NN algorithms, thus minimising the system downtime.

6 | CONCLUSIONS

This study proposes data-driven algorithms to detect and identify IoT-enabled LAAs against power grids. To this end, a system identification problem is investigated, in which the attack parameters that best describe the observed dynamics are estimated from the measurements. Two physics-informed ML algorithms, namely, SR and PINN are proposed, that estimate the attack parameters from the observed data. Their performance is tested extensively using different test systems, including IEEE 6-, 14- and 39-bus systems. Other benchmark approaches, including SVM, NN and UKF, are also investigated to verify the effectiveness of the proposed methods. The numerical results confirm that the proposed data-driven algorithms outperform other benchmark techniques. SR method presents high precision in estimating dynamic attack parameters in single and multi-point attacks with about 3% error on average for fast and slow dynamics. However, the PINN algorithm does not perform well for systems with slow dynamics due to difficulties in training the neural network. The main advantage of SR and PINN compared to fully data-driven algorithms, such as SVM and NN, is that their performance is not dependent on the offline training data structure. Future work includes (i) an extension to multi-area frequency control and optimal placement of PMUs to ensure reliable attack parameter estimation, (ii) machine learning algorithms to differentiate LAAs from generic faults in the system, and (iii) development of a co-simulation platform capable of emulating/simulating the cyber layer and simulating the physical layer (power grid).

ACKNOWLEDGEMENTS

This work has been supported in part by the PETRAS National Centre of Excellence for IoT Systems Cybersecurity, which has been funded by the UK EPSRC under grant number EP/S035362/1. The research of H. Vincent Poor was supported in part by grants from the C3.ai Digital Transformation Institute, the Princeton E-ffiliates Partnership, and the U.S. National Science Foundation (Grant ECCS-2039716).

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Subhash Lakshminarayana https://orcid.org/0000-0002-6349-9780
Saurav Sthapit https://orcid.org/0000-0002-7421-0479

REFERENCES

1. Karnouskos, S.: Stuxnet worm impact on industrial cyber-physical system security. In: Proceedings Annual Conference of the IEEE Industrial Electronics Society, pp. 4490–4494 (2011)
2. Analysis of the cyber attack on the Ukrainian power grid. https://bit.ly/3eEbSEm
3. Lakshminarayana, S., et al.: Signal jamming attacks against communication-based train control: attack impact and countermeasure. In: ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec), pp. 160–171 (2018)
4. Elsisi, M., et al.: Towards secured online monitoring for digitalized GIS against cyber-attacks based on IoT and machine learning. IEEE Access. 9, 78415–78427 (2012)
5. Fernandes, E., Jung, J., Prakash, A.: Security analysis of emerging smart home applications. In: Proceedings IEEE Symposium on Security and Privacy (S&P), pp. 636–654 (2016)
6. Maple, C.: Security and privacy in the internet of things. J. Cyber Policy. 2(2), 155–184 (2017)
7. Liu, Y., Ning, P., Reiter, M.K.: False data injection attacks against state estimation in electric power grids. In: Proc. ACM Conference on Computer and Communications Security (CCS), pp. 21–32 (2009)
8. Lakshminarayana, S., et al.: Data-driven false data injection attacks against power grids: a random matrix approach. IEEE Trans. Smart Grid. 12(1), 635–646 (2021)
9. Lakshminarayana, S., Belmega, E.V., Poor, H.V.: Moving-target defense against cyber-physical attacks in power grids via game theory. IEEE Trans. Smart Grid. 12(6), 5244–5257 (2021)
10. Mohsenian-Rad, A., Leon-Garcia, A.: Distributed internet-based load altering attacks against smart power grids. IEEE Trans. Smart Grid. 2(4), 667–674 (2011)
11. Dabrowski, A., Ullrich, J., Weippl, E.R.: Grid shock: coordinated load-changing attacks on power grids: the non-smart power grid is vulnerable to cyber attacks as well. In: Proc. Annual Computer Security Applications Conference (ACSAC), pp. 303–314 (2017)
12. Soltan, S., Mittal, P., Poor, H.V.: BlackIoT: IoT botnet of high wattage devices can disrupt the power grid. In: Proc. USENIX Security Symposium, Baltimore, pp. 15–32. (2018)
13. Huang, B., Cardenas, A.A., Baldick, R.: Not everything is dark and gloomy: power grid protections against IoT demand attacks. In: Proc. USENIX Security Symposium, pp. 1115–1132 (2019)
14. Dvorkin, Y., Garg, S.: IoT-enabled distributed cyber-attacks on transmission and distribution grids. In: Proc. North American Power Symposium (NAPS), pp. 1–6 (2017)
15. Ospina, J., et al.: On the feasibility of load-changing attacks in power systems during the Covid-19 pandemic. IEEE Access. 9, 2545–2563 (2021)
16. Amini, S., Pasqualetti, F., Mohsenian-Rad, H.: Dynamic load altering attacks against power system stability: attack models and protection schemes. IEEE Trans. Smart Grid. 9(4), 2862–2872 (2018)
17. Lakshminarayana, S., Adhikari, S., Maple, C.: Analysis of IoT-based load altering attacks against power grids using the theory of second-order dynamical systems. IEEE Trans. Smart Grid. 12(5), 4415–4425 (2021)

18. Soltan, S., Mittal, P., Poor, H.V.: Protecting the grid against mad attacks. IEEE Trans. Netw. Sci. Eng. 7(3), 1310–1326 (2020)
19. Amini, S., Pasqualetti, F., Mohsenian-Rad, H.: Detecting dynamic load altering attacks: a data-driven time-frequency analysis. In: Proc. IEEE SmartGridComm, pp. 503–508
20. Amini, S., et al.: Hierarchical location identification of destabilizing faults and attacks in power systems: a frequency-domain approach. IEEE Trans. Smart Grid. 10(2), 2036–2045 (2019)
21. Izbicki, M., et al.: Identification of destabilizing attacks in power systems. In: Proc. American Control Conference (ACC), pp. 3424–3429 (2017)
22. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. U. S. A. 113(15), 3932–3937 (2016)
23. Raissi, M., Perdikaris, P., Karniadakis, G.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707 (2019)
24. Ozay, M., et al.: Machine learning methods for attack detection in the smart grid. IEEE Trans. Neural Netw. Learn. Syst. 27(8), 1773–1786 (2016)
25. He, Y., Mendis, G.J., Wei, J.: Real-time detection of false data injection attacks in smart grid: a deep learning-based intelligent mechanism. IEEE Trans. Smart Grid. 8(5), 2505–2516 (2017)
26. Zhang, Y., Wang, J., Chen, B.: Detecting false data injection attacks in smart grids: a semi-supervised deep learning approach. IEEE Trans. Smart Grid. 12(1), 623–634 (2021)
27. Kundur, P., Balu, N., Lauby, M.: Power System Stability and Control. McGraw-Hill Education (1994)
28. Mitsubishi Electric: Information Disclosure and Denial-of-Service Vulnerability in multiple air conditioning systems. [Online]. Accessed 5 September 2021. https://www.mitsubishielectric.com/en/psirt/vulnerability/pdf/2021-005_en.pdf
29. Mitsubishi Electric: EW-50A Central Controller and Expansion Controller. [Online]. Accessed 19 September 2021. https://www.mitsubishitechinfo.ca/sites/default/files/SB_EW-50A_201604.pdf
30. Consult, S.E.C.: Multiple Vulnerabilities in Fronius Solar Inverter Series. [Online]. Accessed 19 September 2021. https://sec-consult.com/vulnerability-lab/advisory/multiple-vulnerabilites-in-fronius-solar-inverter-series-cve-2019-19229-cve-2019-19228/
31. Acharya, S., et al.: Cybersecurity of smart electric vehicle charging: a power grid perspective. IEEE Access. 8, 214434–214453 (2020)
32. Antonakakis, M., et al.: Understanding the Mirai Botnet. In: Proc. USENIX Security Symposium, Vancouver, pp. 1093–1110. (2017)
33. IEEE/IEC International Standard - Measuring Relays and Protection Equipment - Part 118-1: Synchrophasor for Power Systems - Measurements, IEC/IEEE 60255-118-1, pp. 1–78, (2018)
34. Hussain, S.M.S., Ustun, T.S., Kalam, A.: A review of IEC 62351 security mechanisms for IEC 61850 message exchanges. IEEE Trans. Ind. Inf. 16(9), 5643–5654 (2020)
35. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. B. 58(1), 267–288 (1996)
36. Lagaris, I.E., Likas, A., Fotiadis, D.I.: Artificial neural networks for solving ordinary and partial differential equations. IEEE Trans. Neural Netw. 9(5), 987–1000 (1998)
37. Jahangir, H., et al.: Short-term wind speed forecasting framework based on stacked denoising auto-encoders with rough ANN. Sustain. Energy Technol. Assess. 38, 100601 (2020)

How to cite this article: Lakshminarayana, S., et al.: Data-driven detection and identification of IoT-enabled load-altering attacks in power grids. IET Smart Grid. 1–16 (2022). https://doi.org/10.1049/stg2.12066

APPENDIX

Simulation parameters

System A1: Fast dynamics for IEEE 39-bus system

$$
\begin{aligned}
&M_1 = 2.3186; M_2 : M_8 = 2.6419; M_9 : M_{10} = 2.4862.\\
&K_1^P - K_{10}^P = [1; 0.45; 0.45; 0.1; 0.5; 0.4; 0.3; 0.2; 0.4; 0.5];\\
&K_i^I = 0.6, \forall i \in \mathcal{N}_G; D_i = 2, \forall i \in \mathcal{N}_G; D_i = 0.01, \forall i \in \mathcal{N}_L;
\end{aligned}
$$

System B1: Slow oscillatory dynamics for IEEE 39-bus system

$$
\begin{aligned}
&M_1 = 2.3186; M_2 : M_8 = 2.6419; M_9 : M_{10} = 2.4862.\\
&K_1^P - K_{10}^P = [100; 45; 45; 10; 50; 40; 30; 20; 40; 50];\\
&K_i^I = 60, \forall i \in \mathcal{N}_G; D_i = 2, \forall i \in \mathcal{N}_G; D_i = 0.01, \forall i \in \mathcal{N}_L;
\end{aligned}
$$

System A2: Fast dynamics for IEEE 14-bus system

$$
\begin{aligned}
&M_1 - M_5 = [0.125; 0.034; 0.016; 0.010; 0.015];\\
&D_1 - D_5 = [0.125; 0.068; 0.032; 0.068; 0.072];\\
&K_1^P - K_5^P = [0.02; 0.09; 0.03; 0.03; 0.08];\\
&K_1^I - K_5^I = [0.35; 0.40; 0.35; 0.35; 0.40];\\
&D_i = 0.01, \forall i \in \mathcal{N}_L;
\end{aligned}
$$

System B2: Slow oscillatory dynamics for IEEE 14-bus system

$$
\begin{aligned}
&M_1 - M_5 = [0.125; 0.034; 0.016; 0.010; 0.015];\\
&D_1 - D_5 = [0.125; 0.068; 0.032; 0.068; 0.072];\\
&K_1^P - K_5^P = [2; 9; 3; 3; 8];\\
&K_1^I - K_5^I = [35; 40; 35; 35; 40];\\
&D_i = 0.01, \forall i \in \mathcal{N}_L;
\end{aligned}
$$

System A3: Fast dynamics for IEEE 6-bus system

$$
\begin{aligned}
&M_1 - M_3 = [1.25; 1.25; 1.25];\\
&D_1 - D_3 = [0.125; 0.125; 0.125];\\
&K_1^P - K_5^P = [0.02; 0.09; 0.03];\\
&K_1^I - K_3^I = [0.35; 0.40; 0.35];\\
&D_i = 0.01, \forall i \in \mathcal{N}_L;
\end{aligned}
$$

System B3: Slow oscillatory dynamics for IEEE 6-bus system

$$
\begin{aligned}
&M_1 - M_3 = [1.25; 1.25; 1.25];\\
&D_1 - D_3 = [0.125; 0.125; 0.125];\\
&K_1^P - K_5^P = [2; 9; 3];\\
&K_1^I - K_5^I = [35; 40; 35];\\
&D_i = 0.01, \forall i \in \mathcal{N}_L;
\end{aligned}
$$