Robust Graph Autoencoder-Based Detection of False Data Injection Attacks Against Data Poisoning in Smart Grids

Abdulrahman Takiddin, *Graduate Student Member, IEEE*, Muhammad Ismail, *Senior Member, IEEE*, Rachad Atat, *Member, IEEE*, Katherine R. Davis, *Senior Member, IEEE*, and Erchin Serpedin, *Fellow, IEEE*

***Abstract*—Machine learning-based detection of false data injection attacks (FDIAs) in smart grids relies on labeled measurement data for training and testing. The majority of existing detectors are developed assuming that the adopted datasets for training have correct labeling information. However, such an assumption is not always valid as training data might include measurement samples that are incorrectly labeled as benign, namely, adversarial data poisoning samples, which have not been detected before. Neglecting such an aspect makes detectors susceptible to data poisoning. Our investigations revealed that detection rates (DRs) of existing detectors significantly deteriorate by up to $9-29\%$ when subject to data poisoning in generalized and topology-specific settings. Thus, we propose a generalized graph neural network (GNN)-based anomaly detector that is robust against FDIAs and data poisoning. It requires only benign datasets for training and employs an autoencoder with Chebyshev graph convolutional recurrent layers with attention mechanism to capture the spatial and temporal correlations within measurement data. The proposed convolutional recurrent graph autoencoder (CR-GAE) model is trained and tested on various topologies (from 14, 39, and 118-bus systems). Due to such factors, it yields stable generalized detection performance that is degraded by only $1.6-3.7\%$ in DR against high levels of data poisoning and unseen FDIAs in unobserved topologies.**

***Impact Statement*—Artificial Intelligence (AI) systems are used in smart grids to detect cyberattacks. They can automatically detect malicious actions carried out by malicious entities that falsify measurement data within power grids. The majority of such systems are data-driven and rely on labeled data for model training and testing. However, datasets are not always correctly labeled since malicious entities might be carrying out cyberattacks without being detected, which leads to training on mislabeled datasets. Such actions might degrade the detection rate (DR) of existing AI-based detectors by up to 29%, which causes misinformed decision-making by system operators. In this paper, we overcome this limitation by proposing a robust generalized AI-based detector of cyberattacks that captures spatial topological aspects and temporal correlations within the measurement data. Hence, it could offer a stable DR of 97% against unseen cyberattacks within unseen topologies despite the presence of mislabeled training samples. The proposed data-driven solution could be extended and applied in wider applications such as cyber-physical security of industrial infrastructures, Internet of Things (IoT), and sensor networks.**

Abdulrahman Takiddin, Katherine R. Davis, and Erchin Serpedin are with the Electrical and Computer Engineering Department, Texas A&M University, College Station, TX 77843 USA (e-mail: abdulrahman.takiddin@tamu.edu; katedavis@tamu.edu; eserpedin@tamu.edu).

Muhammad Ismail is with the Department of Computer Science, Tennessee Tech University, Cookeville, TN 38505 USA (e-mail: mismail@tntech.edu).

Rachad Atat is with the ECEN Program, Texas A&M University at Qatar, Doha, Qatar (email: rachad.atat@qatar.tamu.edu).

This work is supported by NSF EPCN Awards 2220346 and 2220347.

***Index Terms*—Adversarial samples, cybersecurity, data poisoning, false data injection attacks, graph autoencoder, graph neural networks, machine learning, smart grids.**

NOMENCLATURE

$\boldsymbol{E}$, $\boldsymbol{D}$	Graph encoder and decoder, respectively.
$\mathcal{G}=(\mathcal{V},\mathcal{E},\boldsymbol{W})$	Graph $\mathcal{G}$ with $\mathcal{V}$ vertices, $\mathcal{E}$ edges, and $\boldsymbol{W}$ weights.
$\boldsymbol{X}_{\mathrm{b}}(i,t)$	Benign sample at bus i and timestamp t with true $y=0$ benign label.
$\boldsymbol{X}_{\mathrm{m}}(i,t)$	Correctly labeled FDIA malicious sample with true $y=1$ malicious label.
$\boldsymbol{X}_{\mathrm{s}}(i,t)$	Adversarial data poisoning sample with false $y=0$ benign label.
Φ	Model parameters.
ζ	Reconstruction error.
ψ	Detection threshold.

I. INTRODUCTION

THE cyber-physical nature of a smart power grid makes it a complex system where lots of measurement data are being continuously interchanged among its components [1]. For proper operation, decision-making, and situational awareness, smart power grids rely on measurement data. Hence, ensuring the integrity of such data is a critical objective when it comes to power system reliability. Unfortunately, when power systems encounter false data injection attacks (FDIAs), the integrity of the data is jeopardized since measurement data is manipulated by malicious entities [2]. Such actions might overload the system due to the resultant incorrect operational decisions [3]. FDIAs present a major challenge since they might be performed in a stealthy manner [4], which can bypass the conventional bad data detection (BDD) systems [5].

A. Related Work and Limitations

Data-driven machine learning (ML)-based approaches were used to detect FDIAs. Such defenses employ (1) ML models with shallow or deep neural network (DNN) structures, or (2) graph-based approaches employing graph signal processing (GSP) filters [6] or graph neural networks (GNNs). Although such mechanisms exhibit promising detection performance, they still present drawbacks, as reviewed next.

1) ML-Based Detection Schemes: When it comes to shallow ML models, F1-Scores of 82% and 88% were reported using support vector machine (SVM) [7] and decision tree [8] detection schemes, respectively. A detection rate (DR) of 93% was offered with a random forest-based detection scheme [9]. However, since measurement data within smart power grids present complex patterns [10], the aforementioned shallow models fail to fully capture such patterns, which explains their relatively poor detection performance.

Unlike shallow models, DNNs can capture more complex patterns and can offer improved detection performance. For example, feedforward neural network (FNN) detection schemes reported accuracy scores of 90% [11] and 99% [12]. Recurrent neural network (RNN) [13] and autoencoder [14] detection schemes achieved DRs of 96% and 96.2%, respectively. Convolutional neural network (CNN) detection schemes exhibited accuracy scores of 93% [15] and 99% [16]. Nevertheless, these numbers do not wholly reflect the practicality of these schemes since their performance is evaluated based on only one network topology without being tested on different topological configurations. Also, these schemes present topology-unaware detectors that fail to capture the spatial aspects and relationships within the power grid data.

Due to the aforementioned limitations of the ML-based detection schemes, according to our investigations presented in Section IV-D, detection schemes based on shallow models and DNNs are highly vulnerable to data poisoning, where their DRs deteriorate by $17 - 29\%$ and $10 - 21\%$, respectively.

2) Graph-based Detection Schemes: To exploit the spatial relationships within power measurements and the grid topology information, graph-based FDIAs detection schemes (GSP and GNN) were proposed in the literature. GSP models exhibited DRs of 90% [17], [18]. A limitation of GSP models is that they necessitate manual and custom filter design, which constrains their scalability. Convolutional GNN (C-GNN) detection schemes present DRs of $83 - 96\%$ [5]. However, the reported performance is still based on only one topological configuration, without considering different topological reconfigurations [5]. Despite capturing the spatial relationships, the simulation results presented in Section IV-D show that the C-GNN detector is still susceptible to data poisoning exhibiting a $9 - 11\%$ degradation in DR. This is because it fails to generalize in the presence of power system topology reconfigurations that may take place for various reasons. Also, it does not fully capture the temporal correlations within the power grid time-series data. Furthermore, such detection schemes are vulnerable to zero-day (unseen) attacks since they are only trained on a predefined set of FDIAs.

It is worth mentioning that the aforementioned studies report different performance metrics for various FDIAs and system sizes, which makes it challenging to compare them due to the lack of common ground. In addition to the aforementioned limitations of existing FDIAs detection schemes, most importantly, they are trained on datasets assuming that the samples have correct labeling information. Neglecting such an aspect makes the detectors susceptible to data poisoning, where malicious samples are incorrectly labeled as benign since they have not been detected before.

3) Data Poisoning Detection Schemes: Only a few studies investigated the impact of data poisoning in smart power grids. For example, linear regression and neural network-based models were proposed to detect data poisoning in load forecasting [19]. Also, a sequential ensemble learning (SEL)-based detector offered high robustness against data poisoning in smart meters [20]. In different contexts, other studies proposed traditional defense schemes against data poisoning using outlier detection [21], local intrinsic dimensionality [22], and federated learning [23]. However, these detectors still present the same limitations as the aforementioned studies (in Sections I-A1 and I-A2) since they do not capture the spatial relationships within the power grid measurement data. Also, these detectors present impractical solutions since they fail in the presence of topological reconfigurations (i.e., lack generalization abilities). Furthermore, the schemes in [19], [21], [22], and [23] are designed specifically to detect data poisoning, without considering other cyberattacks (e.g., other FDIA types). Thus, employing them alone is insufficient when it comes to detecting multiple attack types. Besides requiring an additional data poisoning-filtering step before the actual FDIAs detection scheme, they present data poisoning detection designed only for a specific machine learning model (e.g., SVM) [22], or they are probabilistic, which requires additional computations at the server-side [23]. Therefore, in this work, we seek to create a robust detector that is marginally affected by data poisoning and at the same time, provides a stable detection performance against various types of FDIAs without requiring additional data filtering operations or computations.

Due to the aforementioned limitations of existing detection schemes, there is a need to develop a robust detector that takes into consideration several aspects, including capturing the (i) complex patterns within the measurement data, (ii) spatial aspects of the power system topology, (iii) and temporal correlations within the time-series measurements. Also, the detector should be topology-aware that (iv) offers generalization abilities to capture new topological reconfiguration of the power grid. Due to the lack of correctly labeled data, the detector should be robust against (v) unseen correctly labeled FDIAs within unseen topologies as well as (vi) data poisoning (vii) without the need for an additional data filtering operation. In this paper, we will show that capturing such aspects leads to a robust detection scheme that offers an improved detection performance (by $7 - 25\%$ in DR compared to benchmarks, as presented in Section IV-D2) that is stable despite the presence of data poisoning and unseen FDIAs.

B. Contributions

We overcome the limitations exhibited by data-driven FDIAs benchmark detectors by proposing a generalized GNN-based anomaly detector that is robust against correctly labeled FDIAs as well as data poisoning. The proposed detector requires only benign datasets for training and employs an autoencoder with Chebyshev graph convolutional recurrent layers with attention mechanism to capture the spatial and temporal correlations within measurement data. The proposed convolutional recurrent graph autoencoder (CR-GAE) model is trained and tested on multiple topologies corresponding to systems of different sizes. The contributions of this work are next summarized.

- We quantify the impact of data poisoning by injecting adversarial samples into the train sets of the detectors using multiple injection levels and assess the detection performance degradation. Adversarial samples are generated using six cyberattack functions that bypass traditional BDD. Adversarial samples present manipulated measurements that are incorrectly labeled as benign and are injected in multiple levels that represent 10%, 20%, or 30% of the train set. The simulation studies are conducted on 14, 39, and 118-bus systems, where detectors either utilize multiple topologies (generalized setting) or one topology (topology-specific setting).
- Our investigations revealed that benchmark detectors, including topology-unaware (shallow and deep models) and topology-aware (C-GNN model) are highly vulnerable to data poisoning. Specifically, DRs of shallow and deep models degrade by $17-29\%$ and $10-21\%$, respectively, with the highest injection levels. The C-GNN model offers slightly improved DR that degrades by $9-11\%$ due to its topology-aware nature, but it still requires labeled benign and malicious data for its supervised training. Generalized benchmark detectors degrade by $9-26\%$, whereas topology-specific ones degrade by $10-29\%$ with highest data poisoning injection levels.
- We propose a robust detector against data poisoning and correctly labeled FDIAs based on a CR-GAE model that exhibits the following features. First, it offers topology-aware detection that captures the spatial relationships within the power systems data. Second, it provides a generalization ability and can detect cyberattacks within unseen topologies. Third, it offers an unsupervised anomaly detection that necessitates only benign data for training and offers detection against totally unseen FDIAs, hence offering robustness against zero-day attacks. Fourth, it employs an autoencoder equipped with Chebyshev graph convolutional recurrent layers and attention mechanism to capture the complex patterns as well as the spatial and temporal correlations within the measurement data.
- The features offered by the proposed CR-GAE detector enhance the detection performance. Specifically, it offers a stable detection performance that degrades only by $1.6-3.7\%$ in the presence of high levels of adversarial data poisoning and unseen correctly labeled malicious samples in the train and test sets of unseen topologies, respectively. This means that it enhances the DR degradation by $16-25\%$, $8-17\%$, and $7-8\%$ compared to shallow, deep, and C-GNN models, respectively.

The remainder of the paper is organized as follows. Section II describes the dataset generation in terms of generating various power grid topologies, benign power measurements, and the threat model including the FDIA functions used to generate adversarial and malicious samples along with the attack injection levels. Section III presents the structure of the proposed CR-GAE detector that is robust against data poisoning. Section IV introduces the benchmark detectors and analyzes the impact of data poisoning on them compared to the proposed detector. Section V concludes this paper.

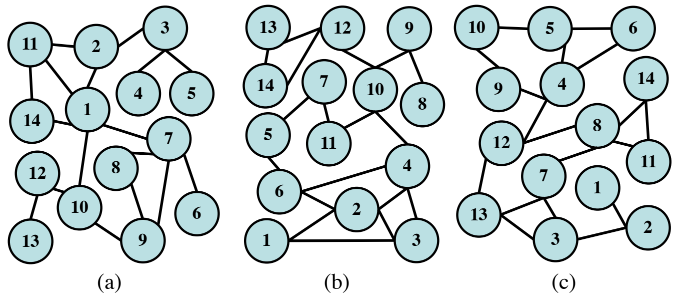


Fig. 1. Three topological configurations of a 14-bus system.

II. Data Generation

We herein discuss the modeling of smart grids via graphs and present the process of generating different bus system topologies and their respective power measurements. We also describe the threat model and attack functions used to generate adversarial and malicious samples. We then present the investigated generalized and topology-specific detection types.

A. Modeling Smart Power Grid Via Graphs

The motivation behind modeling power systems with graphs is to capture the spatial patterns and ascertain the system state (i.e., normal operation or under cyberattack). The tree structure of the power system enables its modeling using GNNs, which can be utilized to identify the system state using both temporal (power injections and flows) and topological (vertices spatial distribution and connections) features. Fig. 1 illustrates a system with 14 buses that are connected in three different configurations.

Our goal is to design a generalized detection scheme that captures topological reconfigurations within the power system, which requires huge amounts of datasets reflecting spatial and temporal features of various configurations of topologies from different system sizes. Unfortunately, such datasets with such features are not readily available. Existing graph-based modeling methods [24], [25], [26] present one of the following limitations. First, they offer unrealistic random graphs that are impractical as they do not capture temporal and spatial topological features. Second, they depend on a particular IEEE bus system topology, and hence, lack the generalization ability towards topological reconfigurations. Third, they adopt information limited to certain cities without disclosing temporal and spatial aspects due to security reasons and agreement policies.

B. Data Generative Model

To counter the aforementioned limitations, we adopt a generative model that creates various topological configurations that belong to practical power systems with different sizes. We adopt stochastic geometry [27] since it allows us to capture physical constraints when connecting different power elements [28] as well as taking into account the spatial coupling and correlations of the electrical elements [29]. In fact, the stochastic geometric morphogenesis of cities adopting iterated Poisson tessellations offers high similarity with the real world [30], which was validated further against IEEE test systems and real power grids [27]. Given the advantages offered by stochastic geometry, we adopt it herein to generate multiple topological configurations that are similar to real power grids.

To generate a topological configuration, we model the geographical area as a disk with radius R. Inside the desk, there are $|M| = 2\pi\lambda_m R$ lines created based on a Poisson line process (PLP) with density λ_m, where each line m is characterized by a line direction of $0 \leq \theta_m < 2\pi$ and a line location of $0 \leq\ \upsilon_m < R$. Then, on each Poisson line m, we create $|\mathcal{V}|_m$ buses based on a one-dimensional homogeneous Poisson point process (HPPP) with density $\lambda_{\mathcal{V}_m}$. Specifically, buses within the system represent an HPPP with density expressed as $\lambda_{\mathcal{V}} = \sum_{m=1}^{|M|} \lambda_{\mathcal{V}_m}$. Buses are linked based on the physical paths via a potential near-geodesic route and the shifted sum of exponential distributions of their degree [28]. Disconnected buses are linked according to the shortest pathways to ensure power supply to the loads and avoid power loss. To assign electrical parameters, we allocate active/reactive powers to buses following the exponential distribution with a mean parameter equal to the mean of active/reactive powers of an actual or test power system that has the same size [24]. Moreover, line impedance values are assigned using the empirical data of IEEE bus systems and New York independent system operator (NYISO) based on [31, Table V], which models the impedance using different distributions depending on the system. The obtained values are then matched with the real values of a similar-sized actual or test power system. By statistically matching the obtained parameters with those of an actual or test power system, we assign the remaining bus and branch electrical parameters (voltage magnitude, voltage angle, megavolt amperes rating, etc.).

Once the topological data (spatial features) are created, we generate the time-series data (temporal features) that simulate the power flow in each topology. For each topology, using MATLAB's MATPOWER toolbox [32], we run the power flow analysis using Newton's method to ascertain the active and reactive power flows. To generate the time-series active and reactive power values, we first normalize the load data profile from the Electric Reliability Council of Texas (ERCOT) [33] into a zero mean and unit standard deviation scalar vector $\digamma = \digamma_1, \digamma_2, ..., \digamma_T]$, to easily adapt it to our test system, where $\digamma_t$ is the scalar value at timestamp t. We multiply the active P and reactive Q power values at the previous timestamp by a scaling sample drawn from a normal distribution with $1 + 0.025 * \digamma_t$ mean and 0.01 standard deviation to get a dynamic variation in the time-series values (P and Q) with respect to their static case (fixed values). Thus, a dynamic range of load values due to the properties of the normal distribution is generated.

C. Generating Benign and Malicious Dataset

In order to create multiple topological reconfigurations using stochastic geometry [27], we replicate the aforementioned processes with a constant $\lambda_{|\mathcal{V}|}$. This way, the spatial features, including nodal degree, degree centrality, and eigenvalue spread of the topological configurations of a given system size are similar to real power systems. For example, in Fig. 1, we model three different 14-bus system topological configurations that have matching spatial features, with a high degree, to the IEEE 14-bus system. Such an approach helps us create multiple topological configurations that can be fed into a GNN-based FDIA detector to enhance its generalization ability to be effective against new unseen topological reconfigurations.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ denote a weighted undirected graph $\mathcal{G}$ that consists of a set of vertices/nodes (buses) $\mathcal{V}$ with $n = |\mathcal{V}|$ number of vertices. $\mathcal{E}$ denotes the edges within $\mathcal{G}$ representing power lines of the grid and $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ depicts the weighted adjacency matrix representing the line admittance within the power grid. For example, W_{ij} refers to the weight assigned to edge $e = (i, j)$ between the connected i and j buses. For each topological configuration, vertices are feature-labeled. For a given $\mathcal{G}$, the generated vertex data is composed of active power P_i (MW) and reactive power Q_i (MVAr).

To design detectors with generalization abilities, we create ten distinct topological configurations for each of the three investigated power system sizes, namely, 14, 39, and 118-bus systems. For each configuration, we include 96 power dynamics timestamps per day for a total period of six months, which leads to having $17,520$ timestamps. Using the generated data, we end up with three different types of data samples. First, the simulation of power flow resulted in generating benign data, where a benign sample $\boldsymbol{X}_{\text{b}}(t, i)$ at timestamp t and bus i has correct labeling information of $y = 0$, reflecting normal operation of the system. Second, using the attack functions to be described in Section II-D1, we generate adversarial data, where an adversarial sample $\boldsymbol{X}_{\text{s}}(t, i)$ has incorrect labeling information, reflecting data poisoning within the system. Samples $\boldsymbol{X}_{\text{s}}$ represent FDIAs that were mislabeled by the grid operator as benign data because they were not detected previously as attack samples. Hence, when they are used to train the detector, they represent poisoned data samples as they have false labels. Third, using the FDIA functions to be described in Section II-D1, we also generate correctly labeled FDIA malicious data, where a malicious sample $\boldsymbol{X}_{\text{m}}(t, i)$ has correct labeling information reflecting operation when the system undergoes FDIAs.

D. Threat Model

In this work, we consider a smart power grid that undergoes two realistic cybersecurity threats:

- The first threat is data poisoning, which refers to cases where a system operator trains a model on mislabeled datasets, which leads to feeding the model with samples that are associated with false labels [20]. Data poisoning occurs if an attack sample has not been detected before, and is being fed to under-development detectors with a false benign label. The threat herein is that the decision boundary of the trained detector will be shifted since it is trained on mislabeled data, which results in falsely marking future malicious data as benign samples [34]. This threat does not assume any special capability, system access, or adopted detector knowledge by the attacker since it reflects previous undetected attack samples present in the training set of the detector with false benign labels.

- The second threat is correctly labeled FDIAs where an attacker manipulates measurement data in a stealthy manner to bypass traditional BDD systems and falsify the actual state of the power system [2]. The threat herein is that falsifying measurement data could lead to making inaccurate decisions and overloading the system [3]. This threat only requires the attacker to capture a specific reading or a series of readings using network traffic monitoring tools (e.g., Wireshark) and falsely manipulate previous samples then inject them into the system without special capability nor prior knowledge about the system, graph structure, or adopted detector.

The goal is to design a detector that is robust against the aforementioned threats while offering generalization ability. Such a robust detector can be implemented by the system operators in order to detect data poisoning and correctly labeled FDIAs, which will allow system operators to make more accurate decisions and improve the stability of the power grid. Next, we discuss the details of the attack datasets along with the generation process of attack samples to simulate the presence of the two aforementioned threats.

We consider six FDIA functions to generate adversarial data poisoning and correctly labeled FDIA malicious samples. First, adversarial data poisoning samples $\boldsymbol{X}_{\text{s}}$ are injected into the training sets to simulate data poisoning with incorrect labeling information. Second, correctly labeled FDIA malicious samples $\boldsymbol{X}_{\text{m}}$ are injected into the test set of supervised and unsupervised detectors (and train set of supervised detectors) to simulate correctly labeled FDIAs. We consider multiple FDIA functions [35] including a random attack, four data replay attacks, and a general attack. To ensure the stealthiness of such attacks while not being detected, the difference between the altered and true measurement data is maintained below a threshold value that is deemed acceptable yet effective to bypass the conventional BDD of power systems.

1) Attack Functions: Below are the FDIA functions that create data poisoning and correctly labeled malicious samples.

a) Random Attack: Altering measurement data herein is carried out where a small perturbation value α is applied into benign samples and affect their integrity. An adversarial sample $\boldsymbol{X}_{\text{s}}(t,i)$ at timestamp t and bus i is generated as follows

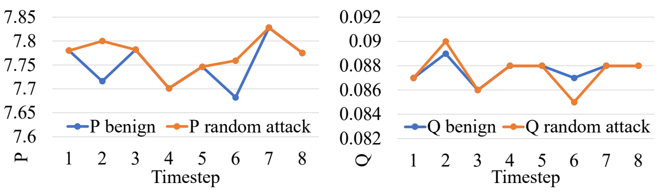
$$\boldsymbol{X}_{\text{s}}(t,i) = \boldsymbol{X}_{\text{b}}(t,i) + \alpha.\boldsymbol{X}_{\text{b}}(t,i), \tag{1}$$

where $-0.05 \leq \alpha \leq 0.05$ is a random variable denoting the perturbation magnitude that is randomly applied to a given benign sample $\boldsymbol{X}_{\text{b}}(t,i)$ to maliciously alter the measurement. Fig. 2a illustrates a random attack occurring at $t = 2$ and $t = 6$ in a 14-bus system compared to a normal operation.

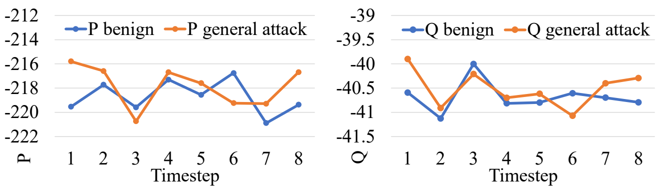
b) General Attack: Generating samples using the general attack is carried out such that

$$\boldsymbol{X}_{\text{s}}(t,i) = \boldsymbol{X}_{\text{b}}(t,i) + (-1)^{\beta}\alpha.\gamma.\text{Range}(\boldsymbol{X}_{\text{b}}(t,i)), \tag{2}$$

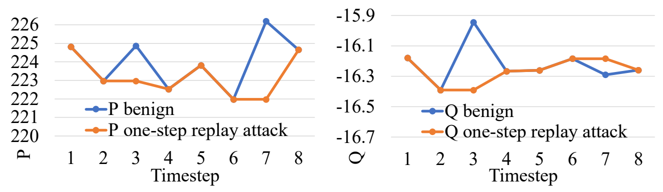
where β and α denote a binary random variable and the attack magnitude, respectively. γ is a uniform random variable between $(0,1)$. The last term in (2) depicts the true measurements range at timestamp t and bus i. Fig. 2b illustrates a comparison between benign and attack sample values generated via (2) over a two-hour period in a 14-bus system.



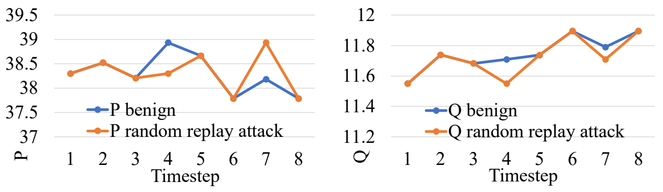
(a) P and Q of benign and random attack samples.



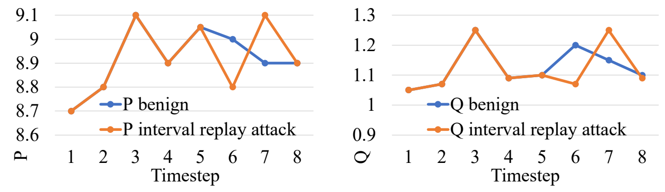
(b) P and Q of benign and general attack samples.



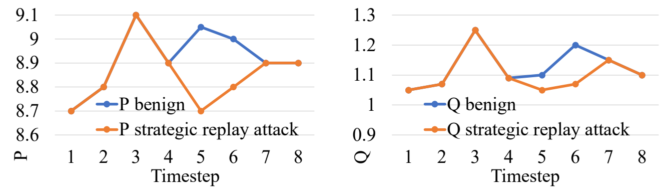
(c) P and Q of benign and one-step replay attack samples.



(d) P and Q of benign and random replay attack samples.



(e) P and Q of benign and interval replay attack samples.



(f) P and Q of benign and strategic replay attack samples.

Fig. 2. P and Q of benign and attack samples.

c) Replay Attacks: We consider four cases of replay attacks. The first two attacks, namely, one-step and random replay attacks, require selecting a benign sample and replacing it with a true measurement value from a previous timestamp. In the one-step replay attack, data from a previous timestamp $(t-1)$ is repeated, where $\boldsymbol{X}_{\text{s}}(t,i)$ is generated as follows

$$\boldsymbol{X}_{\text{s}}(t,i) = \boldsymbol{X}_{\text{b}}(t-1,i). \tag{3}$$

An illustration of a one-step replay attack occurring at $t = 3$ and $t = 7$ compared to a normal operation is presented in Fig. 2c. In the random replay attack, data from a randomly selected previous timestamp $2 \leq \hat{t} \leq 5$ is repeated, where $\boldsymbol{X}_\text{s}(t, i)$ is generated as follows

$$\boldsymbol{X}_\text{s}(t, i) = \boldsymbol{X}_\text{b}(t - \hat{t}, i). \tag{4}$$

An illustration of a random replay attack taking place at $t = 4$ and $t = 7$ compared to a normal operation is presented in Fig. 2d. The previous two replay attacks target repeating one attack sample at a time. The remaining replay attacks, namely, interval replay and strategic replay attacks repeat a series of benign readings and hence are considered to be stealthier since they introduce a series of consecutive attack samples that follow benign patterns. Specifically, the interval replay and strategic replay attacks require selecting a series of consecutive benign samples $[\boldsymbol{X}_\text{b}(t_n, i), \cdots, \boldsymbol{X}_\text{b}(t_m, i)]$ within time interval $[t_n, \cdots, t_m]$ and replacing them with a series of consecutive true measurement values from previous timestamps. In the interval replay attack, a random series of benign readings gets replaced with $[\boldsymbol{X}_\text{b}(t_{\hat{n}}, i), \cdots, \boldsymbol{X}_\text{b}(t_{\hat{m}}, i)]$ from a random previous time interval $[t_{\hat{n}}, \cdots, t_{\hat{m}}]$ such that

$$[\boldsymbol{X}_\text{s}(t_n, i), \cdots, \boldsymbol{X}_\text{s}(t_m, i)] = [\boldsymbol{X}_\text{b}(t_{\hat{n}}, i), \cdots, \boldsymbol{X}_\text{b}(t_{\hat{m}}, i)]. \tag{5}$$

An illustration of an interval replay attack occurring at timestamps 6, 7 and 8 compared to a normal operation is presented in Fig. 2e. In the strategic replay attack, $[\boldsymbol{X}_\text{b}(t_n, i), \cdots, \boldsymbol{X}_\text{b}(t_m, i)]$ get replaced with benign measurements from a previous time interval $[t_{\dot{n}}, \cdots, t_{\dot{m}}]$ such that

$$[\boldsymbol{X}_\text{s}(t_n, i), \cdots, \boldsymbol{X}_\text{s}(t_m, i)] = [\boldsymbol{X}_\text{b}(t_{\dot{n}}, i), \cdots, \boldsymbol{X}_\text{b}(t_{\dot{m}}, i)], \tag{6}$$

where samples $[\boldsymbol{X}_\text{b}(t_{\dot{n}}, i), \cdots, \boldsymbol{X}_\text{b}(t_{\dot{m}}, i)]$ must present higher /lower measurement values than $[\boldsymbol{X}_\text{b}(t_n, i), \cdots, \boldsymbol{X}_\text{b}(t_m, i)]$. This way, a series of relatively high measurement values get replaced with previous values that are lower and vice versa. An illustration of a strategic replay attack occurring at timestamps 5 and 6 compared to a normal operation is presented in Fig. 2f. In the interval replay and strategic replay attacks, the length of the time interval is randomly selected to be between 2 and 5 since, according to our experiments, replay attacks with longer time intervals can be easily spotted by the detectors.

2) Adversarial Data Poisoning Samples ($\boldsymbol{X}_s$): Data poisoning refers to cases where a model is trained on data with the implicit assumption that the training samples have correct information with true benign samples labeled as benign. However, in reality, this assumption is not always valid as sample values and labels fed into a given machine learning model are not always necessarily true. Consider the case where malicious entities have been altering measurement data (e.g., manipulating the sensor data) and carrying out FDIAs without being detected by traditional BDD. These altered measurements generated using cyberattacks will be incorrectly marked as benign with incorrect labeling information. Such generated samples are called adversarial data poisoning samples $\boldsymbol{X}_\text{s}$. When such poisoned data is used to train detectors of FDIAs, the train set will falsely contain adversarial samples with benign labels. Thus, detectors will be trained on and dealing with such adversarial patterns as if they present benign measurements [20]. These cases are depicted as data poisoning, which lead to shifting the detectors' decision boundary and degrading the FDIAs detection ability [34]. To quantify the impact of data poisoning, the aforementioned attack functions (1) - (6) are used to create adversarial samples that are injected into the train sets of the investigated detectors with a false benign label of $y = 0$ instead of the true malicious $y = 1$ label. The attack functions are used to generate an equal number of adversarial samples where samples from each attack function present $1/6$ of $\boldsymbol{X}_\text{s}$.

3) Correctly Labeled FDIA Malicious Samples ($\boldsymbol{X}_m$): The attack functions (1) - (6) are also adopted to generate correctly labeled FDIA malicious samples $\boldsymbol{X}_\text{m}$ and are assigned the true malicious label of $y = 1$. The attack functions are used to generate an equal number of correctly labeled FDIA samples where samples from each attack function present $1/6$ of $\boldsymbol{X}_\text{m}$. Samples $\boldsymbol{X}_\text{m}$ present in the test set of the investigated (supervised and unsupervised) detectors mimic a detector that is encountering FDIAs. Samples $\boldsymbol{X}_\text{m}$ are also present in the train sets of the supervised benchmark detectors denoting correctly labeled FDIAs that were previously detected.

4) Train and Test Sets: Benign samples $\boldsymbol{X}_\text{b}$ have correct labeling information of $y = 0$. Adversarial data poisoning samples $\boldsymbol{X}_\text{s}$ have incorrect labeling information of $y = 0$. Correctly labeled FDIA malicious samples $\boldsymbol{X}_\text{m}$ have correct labeling information of $y = 1$. Since supervised detectors perform binary classification, they require training on samples with $y = 0$ and $y = 1$ labels. Thus, supervised detectors are trained on $\boldsymbol{X}_\text{b}$ with $y = 0$ labels, $\boldsymbol{X}_\text{s}$ with $y = 0$ labels, and $\boldsymbol{X}_\text{m}$ with $y = 1$ labels, whereas they are tested on $\boldsymbol{X}_\text{b}$ with $y = 0$ labels and $\boldsymbol{X}_\text{m}$ with $y = 1$ labels. However, unsupervised anomaly detectors require training on samples with one label ($y = 0$). Thus, unsupervised detectors are trained on $\boldsymbol{X}_\text{b}$ with $y = 0$ labels and $\boldsymbol{X}_\text{s}$ with $y = 0$ labels, whereas they are tested on $\boldsymbol{X}_\text{b}$ with $y = 0$ labels and $\boldsymbol{X}_\text{m}$ with $y = 1$ labels. Since data poisoning denotes mislabeled samples $\boldsymbol{X}_\text{s}$ that have not been detected before, they are only present in the train sets. However, samples $\boldsymbol{X}_\text{m}$ are present in the test set of the investigated (supervised and unsupervised) detectors to mimic a situation when a system is encountering FDIAs. Samples $\boldsymbol{X}_\text{m}$ are also present in the train sets of the supervised benchmark detectors denoting correctly labeled FDIAs that were previously detected. The percentages of the samples are discussed in Section II-D5

5) Attack Injection Levels: Since data poisoning is only present in the training stage, we investigate their impact by launching adversarial samples $\boldsymbol{X}_\text{s}$ into $\boldsymbol{X}_\text{TR}$ using four levels for each training topology. We generate equal numbers of adversarial samples using (1) - (6) and inject them into $\boldsymbol{X}_\text{TR}$ through multiple attack injection levels where $\boldsymbol{X}_\text{b}$ and $\boldsymbol{X}_\text{s}$ are split as follows. The first level contains 0% adversarial samples (100% true benign samples). The second level contains 10% adversarial samples (90% true benign samples). The third level contains 20% adversarial samples (80% true benign samples). The fourth level contains 30% adversarial samples (70% true benign samples). The rationale behind investigating the impact of data poisoning using such levels is that deep and graph benchmark detectors (as will be shown in Table II) offer DR of $69-90\%$ and false alarm rate (FAR) of $8-31\%$ with correct labeling information (without the presence of data poisoning). This means that around $10 - 31\%$ of malicious samples go undetected and are falsely labeled as normal operation. Similarly, $8-31\%$ of benign samples are incorrectly labeled as malicious. Thus, we conclude that false labeling could take place in up to 30% of the readings. The aforementioned portions represent the entire $\boldsymbol{X}_{\text{TR}}$ of unsupervised detectors. However, in supervised detectors, such portions represent 50% of $\boldsymbol{X}_{\text{TR}}$ and the rest presents malicious samples $\boldsymbol{X}_{\text{m}}$. In $\boldsymbol{X}_{\text{TST}}$ of supervised and unsupervised detectors, the number of $\boldsymbol{X}_{\text{b}}$ and $\boldsymbol{X}_{\text{m}}$ samples is the same.

E. Detection Type

For comprehensive comparisons, we adopt two detection types: generalized and topology-specific. The investigated detectors are examined using both detection mechanisms.

1) Generalized Training: To achieve a generalized detection that performs well against unseen topology (i.e., when topological reconfigurations take place), we adopt a generalized training approach utilizing ten different topologies for each of the 14, 39, and 118-bus systems. The topologies are represented as $\boldsymbol{\Gamma} = [1, 2, ..., 10]$ where training, testing, and validating topologies are selected in each experiment following a leave-one-out method [36]. We carry out eight different generalized experiments for each of the 14, 39, and 118-bus systems. Each experiment defines seven, one, and two topologies as its own training $\boldsymbol{X}_{\text{TR}}$, validation $\boldsymbol{X}_{\text{VAL}}$, and test set $\boldsymbol{X}_{\text{TST}}$, respectively. The average detection performance of these eight experiments is reported in Section IV-D2a.

2) Topology-Specific Training: We also investigate the impact of data poisoning on detectors that are built based on a specific topological configuration. In such a detection type, the models are trained (e.g., $\boldsymbol{\Gamma} = \{1\}$) and validated (e.g., $\boldsymbol{\Gamma} = \{2\}$) using one topological configuration and tested on two unseen topological configurations (e.g., $\boldsymbol{\Gamma} = \{3, 4\}$). The leave-one-out is also carried out herein to perform multiple iterations of the experiments, where the average detection performance is reported in Section IV-D2b.

III. Robust Detection Mechanism

This section introduces the proposed FDIA detector that is robust against data poisoning and correctly labeled FDIAs. The proposed CR-GAE anomaly detector offers unsupervised training that requires only benign data $\boldsymbol{X}_{\text{b}}$ of normal operation for training [37] since it employs an autoencoder. It also offers generalized training that is carried out on multiple graph representations of various topologies, with the ability of detecting unseen FDIA types within different unseen topologies. The structure of the CR-GAE model allows it to capture the complex patterns along with temporal and spatial aspects within the data due to the presence of Chebyshev graph convolutional recurrent layers equipped with attention mechanism [38]. Capturing such features allows it to distinguish the different sample types, which makes it robust against data poisoning as well as other unseen malicious FDIAs.

A. Proposed Model Architecture

The structure of the proposed CR-GAE detector is presented in Fig. 3. Through the graph encoder and decoder, the model

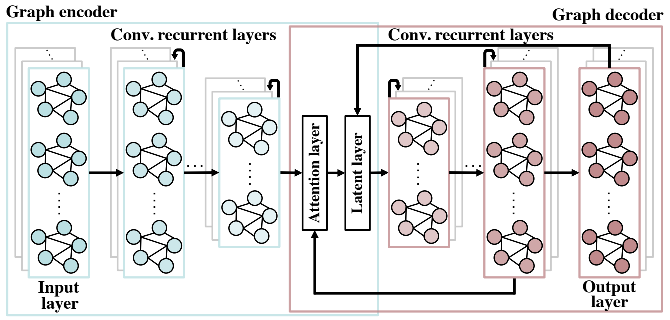


Fig. 3. Illustration of the proposed unsupervised CR-GAE detection.

learns the representations of different graphs of benign data from normal operations using a graph-reconstruction process [39]. During training, the input to the model is sampled with $y = 0$ labels (i.e., benign samples $\boldsymbol{X}_{\text{b}}$ with true labels and poisoned samples $\boldsymbol{X}_{\text{s}}$ with false labels). The input samples hold temporal measurement values $[P_i, Q_i] \in \mathbb{R}^{n\times 2}$. The input layer is followed by graph encoding hidden layers $\mathcal{L}_{\text{E}}$, which are followed by an attention L_A and latent layer L_S. Then, the graph decoding hidden layers $\mathcal{L}_{\text{D}}$ are placed, followed by the reconstructed outputs $\tilde{\boldsymbol{X}}$. Next, we describe each component of the proposed CR-GAE model.

1) Graph Encoder $\boldsymbol{E}$: The graph encoding hidden layers $\mathcal{L}_{\text{E}}$ present hidden Chebyshev graph convolutional recurrent layers. The channels of an encoding layer l_{E} are denoted as $c_{l_{\text{E}}}$. Each l_{E} outputs $\boldsymbol{X}^{l_{\text{E}}} \in \mathbb{R}^{n\times c_{l_{\text{E}}}}$ and takes $\boldsymbol{X}^{l_{\text{E}}-1} \in \mathbb{R}^{n\times c_{l_{\text{E}}-1}}$ as an input. The presence of $\mathcal{L}_{\text{E}}$ helps capture the spatial features since such layers perform the graph convolution operation. After each l_{E}, bias and ReLU activation function are added to enhance the model's nonlinear capability [40]. The ReLU function generates the output tensor $\boldsymbol{X}^{l_{\text{E}}}$ of l_{E} such that

$$\boldsymbol{X}^{l_{\text{E}}} = \text{ReLU}(\boldsymbol{\mu}^{l_{\text{E}}} *_{\mathcal{G}} \boldsymbol{X}^{l_{\text{E}}-1} + \boldsymbol{b}^{l_{\text{E}}}), \tag{7}$$

where $\boldsymbol{\mu}^{l_{\text{E}}} \in \mathbb{R}^{K\times c_{l_{\text{E}}-1}\times c_l}$ depicts the K order Chebyshev coefficients, $\boldsymbol{b}^{l_{\text{E}}} \in \mathbb{R}^{c_{l_{\text{E}}}}$ depicts the bias, and $*_{\mathcal{G}}$ represents the graph convolution operator. Using a recurrent flow of information that is controlled using long short-term memory (LSTM) gates, $\mathcal{L}_{\text{E}}$ are also able to capture the temporal correlations within the time-series data as well as handling the vanishing/exploding gradient problem during the learning process of the time-series patterns, especially with long intervals [41]. Specifically, at timestamp t, an LSTM cell presents a cell state $s_t^{l_{\text{E}}}$ and outputs a cell hidden state $h_t^{l_{\text{E}}}$. Each LSTM cell receives the cell state and hidden state of the previous cell denoted as $s_{t-1}^{l_{\text{E}}}$ and $h_{t-1}^{l_{\text{E}}}$, respectively. Access to an LSTM cell is managed by three gates, namely, input $i_t^{l_{\text{E}}}$, output $o_t^{l_{\text{E}}}$, and forget $f_t^{l_{\text{E}}}$ gates. In particular,

- $\boldsymbol{i}_t^{l_{\text{E}}} = \varphi(\boldsymbol{W}_i^{l_{\text{E}}}\boldsymbol{X}_t^{l_{\text{E}}} + \boldsymbol{U}_i^{l_{\text{E}}}\boldsymbol{h}_{t-1}^{l_{\text{E}}} + \boldsymbol{V}_i^{l_{\text{E}}}\boldsymbol{s}_{t-1}^{l_{\text{E}}} + \boldsymbol{b}_i^{l_{\text{E}}})$
- $\boldsymbol{o}_t^{l_{\text{E}}} = \varphi(\boldsymbol{W}_o^{l_{\text{E}}}\boldsymbol{X}_t^{l_{\text{E}}} + \boldsymbol{U}_o^{l_{\text{E}}}\boldsymbol{h}_{t-1}^{l_{\text{E}}} + \boldsymbol{V}_o^{l_{\text{E}}}\boldsymbol{s}_t^{l_{\text{E}}} + \boldsymbol{b}_o^{l_{\text{E}}})$
- $\boldsymbol{f}_t^{l_{\text{E}}} = \varphi(\boldsymbol{W}_f^{l_{\text{E}}}\boldsymbol{X}_t^{l_{\text{E}}} + \boldsymbol{U}_f^{l_{\text{E}}}\boldsymbol{h}_{t-1}^{l_{\text{E}}} + \boldsymbol{V}_f^{l_{\text{E}}}\boldsymbol{s}_{t-1}^{l_{\text{E}}} + \boldsymbol{b}_f^{l_{\text{E}}})$
- $\boldsymbol{s}_t^{l_{\text{E}}} = f_t^{l_{\text{E}}} s_{t-1}^{l_{\text{E}}} + i_t^{l_{\text{E}}} \tanh(\boldsymbol{W}_s^{l_{\text{E}}}\boldsymbol{X}_t^{l_{\text{E}}} + \boldsymbol{U}_s^{l_{\text{E}}}\boldsymbol{h}_{t-1}^{l_{\text{E}}} + \boldsymbol{b}_s^{l_{\text{E}}})$
- $\boldsymbol{h}_t^{l_{\text{E}}} = \boldsymbol{o}_t^{l_{\text{E}}} \tanh(\boldsymbol{s}_t^{l_{\text{E}}})$,

where $\varphi(\cdot)$ refers to the activation function and $\boldsymbol{W}_{(\cdot)}$, $\boldsymbol{U}_{(\cdot)}$, $\boldsymbol{V}_{(\cdot)}$, and $\boldsymbol{b}_{(\cdot)}$ denote the learnable weight and bias.

2) Attention Layer L_{A}: The attention layer L_{A} is placed to allocate higher importance to timestamps offering more contribution with respect to acquiring a given output [20]. Towards this objective, L_{A} takes, as inputs, the graph encoder's hidden state $\boldsymbol{h}_t^{L_{\text{E}}}$ at timestamp t and the graph decoder's hidden state $\boldsymbol{h}_{t-1}^{L_{\text{D}}}$ at timestamp $t-1$. The attention mechanism is performed through an alignment score $\boldsymbol{\kappa}$, Softmax $\boldsymbol{\Omega}$, and multiplication. The alignment score $\boldsymbol{\kappa}$ is calculated as follows

$$\boldsymbol{\kappa} = \xi(\boldsymbol{h}_t^{L_{\text{E}}/2}, \boldsymbol{h}_{t-1}^{L_{\text{D}}}), \tag{8}$$

where ξ denotes the alignment function, which presents an FNN that is trained using $\boldsymbol{h}_t^{L_{\text{E}}/2}$ and $\boldsymbol{h}_{t-1}^{L_{\text{D}}}$ jointly. The attention weight is presented as a Softmax of alignment scores such that

$$\boldsymbol{\Omega} = \frac{\exp(\boldsymbol{\kappa})}{\sum_{|\boldsymbol{\kappa}|}\exp(\boldsymbol{\kappa})}, \tag{9}$$

where $|\boldsymbol{\kappa}|$ stands for the cardinality of κ. L_{A} then outputs a context vector Λ_t at timestamp t that is calculated based on the weighted sum of the hidden state vector of the graph encoder:

$$\Lambda_t = \sum_{T} \Omega \times \boldsymbol{h}_t^{L_{\text{E}}/2}. \tag{10}$$

3) Latent Layer L_{S}: The latent layer L_{S} holds the representations of the compressed data during the graph encoding process. L_{S} helps in learning features as well as simpler representations of the data [42]. Specifically, L_{S} takes Λ_t along with the reconstructed output $\tilde{\boldsymbol{X}}$ of the graph decoder as inputs. The concatenation $\breve{\boldsymbol{X}} = \sum(\Lambda_t, \tilde{\boldsymbol{X}})$ then takes place and gets fed to the following graph decoder $\boldsymbol{D}$ block.

4) Graph Decoder $\boldsymbol{D}$: The graph decoder $\boldsymbol{D}$ performs the data reconstruction process through stacked graph Chebyshev convolutional recurrent decoding hidden layers $\mathcal{L}_{\text{D}}$. Specifically, the presented concatenation $\breve{\boldsymbol{X}}$ in L_{S} is fed into the proceeding decoding hidden layer l_{D}. Similar to the graph encoder, the channels of a decoder layer l_{D} are denoted as $c_{l_{\text{D}}}$. Each l_{D} outputs $\breve{\boldsymbol{X}}^{l_{\text{D}}} \in \mathbb{R}^{n \times c_{l_{\text{D}}}}$ and takes $\breve{\boldsymbol{X}}^{l_{\text{D}}-1} \in \mathbb{R}^{n \times c_{l_{\text{D}}-1}}$ as an input. After each l_{D}, bias and ReLU activation function are added where the ReLU function generates the output tensor $\breve{\boldsymbol{X}}^{l_{\text{D}}}$ of l_{D} as follows

$$\breve{\boldsymbol{X}}^{l_{\text{D}}} = \text{ReLU}(\boldsymbol{\mu}^{l_{\text{D}}} *_{\mathcal{G}} \breve{\boldsymbol{X}}^{l_{\text{D}}-1} + \boldsymbol{b}^{l_{\text{D}}}). \tag{11}$$

As presented in the graph encoder side, access to the LSTM cells in the graph decoder is controlled by three gates, namely, input $i_t^{l_{\text{D}}}$, output $o_t^{l_{\text{D}}}$, and forget $f_t^{l_{\text{D}}}$ gates, where

- $\boldsymbol{i}_t^{l_{\text{D}}} = \varphi(\boldsymbol{W}_i^{l_{\text{D}}}\breve{\boldsymbol{X}}_t^{l_{\text{D}}} + \boldsymbol{U}_i^{l_{\text{D}}}\boldsymbol{h}_{t-1}^{l_{\text{D}}} + \boldsymbol{V}_i^{l_{\text{D}}}\boldsymbol{s}_{t-1}^{l_{\text{D}}} + \boldsymbol{b}_i^{l_{\text{D}}})$
- $\boldsymbol{o}_t^{l_{\text{D}}} = \varphi(\boldsymbol{W}_o^{l_{\text{D}}}\breve{\boldsymbol{X}}_t^{l_{\text{D}}} + \boldsymbol{U}_o^{l_{\text{D}}}\boldsymbol{h}_{t-1}^{l_{\text{D}}} + \boldsymbol{V}_o^{l_{\text{D}}}\boldsymbol{s}_t^{l_{\text{D}}} + \boldsymbol{b}_o^{l_{\text{D}}})$
- $\boldsymbol{f}_t^{l_{\text{D}}} = \varphi(\boldsymbol{W}_f^{l_{\text{D}}}\breve{\boldsymbol{X}}_t^{l_{\text{D}}} + \boldsymbol{U}_f^{l_{\text{D}}}\boldsymbol{h}_{t-1}^{l_{\text{D}}} + \boldsymbol{V}_f^{l_{\text{D}}}\boldsymbol{s}_{t-1}^{l_{\text{D}}} + \boldsymbol{b}_f^{l_{\text{D}}})$
- $\boldsymbol{s}_t^{l_{\text{D}}} = f_t^{l_{\text{D}}}s_{t-1}^{l_{\text{D}}} + i_t^{l_{\text{D}}}\tanh(\boldsymbol{W}_s^{l_{\text{D}}}\breve{\boldsymbol{X}}_t^{l_{\text{D}}} + \boldsymbol{U}_s^{l_{\text{D}}}\boldsymbol{h}_{t-1}^{l_{\text{D}}} + \boldsymbol{b}_s^{l_{\text{D}}})$
- $\boldsymbol{h}_t^{l_{\text{D}}} = \boldsymbol{o}_t^{l_{\text{D}}}\tanh(\boldsymbol{s}_t^{l_{\text{D}}})$.

Finally, the reconstruction process of the network's original input is carried out via the graph decoding layers, which leads to generating the reconstructed output $\tilde{\boldsymbol{X}}$ at the output layer.

B. Proposed Model Training and Testing

The proposed CR-GAE model learns the normal operation of benign samples and marks abnormal operation of malicious samples based on the presented deviation from the learned benign patterns. Determining when anomalies take place is based on a reconstruction error ζ throughout the reconstruction process. Let $\boldsymbol{E} = f_\Phi(\boldsymbol{X})$ and $\boldsymbol{D} = g_\Phi(\boldsymbol{X})$ for the graph encoder and decoder, respectively. The cost function of the CR-GAE proposed model is expressed as

$$\min_{\Phi} C(\boldsymbol{X}, g_\Phi(f_\Phi(\boldsymbol{X}))), \qquad \boldsymbol{X} \in \boldsymbol{X}_{\text{TR}}, \tag{12}$$

where Φ depicts the parameters of the model. The goal of this cost function (i.e., mean squared error (MSE)) is to penalize $g_\Phi(f_\Phi(\boldsymbol{X}))$ due to the presented dissimilarity from $\boldsymbol{X}$.

In Algorithm 1, we present the training process of the proposed CR-GAE model, which aims to find the model parameters $\boldsymbol{\mu}^{l_{(\cdot)}}$, $\boldsymbol{b}^{l_{(\cdot)}}$, $\boldsymbol{W}_{(\cdot)}$, $\boldsymbol{U}_{(\cdot)}$, and $\boldsymbol{V}_{(\cdot)}$, denoted by Φ, that optimize (12). Such minimization is achieved via the iterative gradient descent algorithm, described in Algorithm 1, by utilizing a stochastic gradient descent execution. In Algorithm 1, η, ∇, and $|\boldsymbol{X}_{\text{TR}}|$ depict the learning rate, partial derivative, and number of training samples, respectively. We split the training samples $\boldsymbol{X} \in \boldsymbol{X}_{\text{TR}}$ into mini batches that are equally sized and feed them into the model in 128 epochs. Although all training samples $\boldsymbol{X} \in \boldsymbol{X}_{\text{TR}}$ (benign samples $\boldsymbol{X}_{\text{b}}$ with true labels and poisoned samples $\boldsymbol{X}_{\text{s}}$ with false labels) are labeled with $y = 0$, the proposed model is able to differentiate between samples and learn relevant features from true benign ones due to its generalization anomaly detection abilities. Also, its structure helps distinguish between samples via convolutional recurrent graph layers equipped with attention mechanism.

C. Setting Detection Threshold Value (ψ)

After the training stage on training samples $\boldsymbol{X} \in \boldsymbol{X}_{\text{TR}}$, given test samples $\boldsymbol{X} \in \boldsymbol{X}_{\text{TST}}$ could be either benign or malicious. This is determined based on the comparison between the reconstruction error ζ and a certain threshold ψ. ζ is used to quantify how familiar the model is with a given sample, where lower or higher ζ values indicate the presence of benign or malicious patterns, respectively. The value of ψ, which separates benign from malicious samples, is set based on the median of the interquartile range (IQR) of the receiver operating characteristic (ROC) curve. Using (12), we obtain ζ between $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$, if $\zeta < \psi$ or $\zeta > \psi$, $\boldsymbol{X} \in \boldsymbol{X}_{\text{TST}}$ are considered benign with $y = 0$ or malicious with $y = 1$ labels, respectively.

IV. Experimental Results

In this section, we investigate the impact of data poisoning on several benchmark generalized and topology-specific detectors with different characteristics. We apply hyperparameter optimization to all investigated detectors. Then, we report the impact of data poisoning using different attack injection levels. We then provide model analysis in terms of complexity and offline training time, generalized ability requirements, and scalability when testing in real time.

Algorithm 1: Proposed CR-GAE model training

```
Input Data: $\boldsymbol{X}_{\text{TR}}$
Initialization: parameters $\Phi$: $\boldsymbol{\mu}^{l_{(\cdot)}}$, $\boldsymbol{b}^{l_{(\cdot)}}$, $\boldsymbol{W}_{(\cdot)}^{l_{(\cdot)}}$, $\boldsymbol{U}_{(\cdot)}^{l_{(\cdot)}}$,
  and $\boldsymbol{V}_{(\cdot)}^{l_{(\cdot)}}$ $\forall\, l_{(\cdot)}$, $\boldsymbol{h}_{t-1}^{L_{\text{D}}}$, and $\tilde{\boldsymbol{X}}$
while not converged do
    for each topology $\boldsymbol{\Gamma}$ do
        for each training sample $\boldsymbol{X}$ do
            Feedforward:
            Graph Encoder ($\boldsymbol{E}$):
            for each graph encoding layer $l_{\text{E}} \in \mathcal{L}_{\text{E}}$ do
                $\boldsymbol{X}^{l_{\text{E}}} = \text{ReLU}(\boldsymbol{\mu}^{l_{\text{E}}} *_{\mathcal{G}} \boldsymbol{X}^{l_{\text{E}}-1} + \boldsymbol{b}^{l_{\text{E}}})$
                for each timestamp $t$ do
                    Compute:
                    $\boldsymbol{i}_t^{l_{\text{E}}}$, $\boldsymbol{o}_t^{l_{\text{E}}}$, $\boldsymbol{f}_t^{l_{\text{E}}}$, $\boldsymbol{s}_t^{l_{\text{E}}}$, and $\boldsymbol{h}_t^{l_{\text{E}}}$
                end
            end
            Attention Layer ($L_{\text{A}}$):
            $\boldsymbol{\kappa} = \xi(\boldsymbol{h}_t^{L_{\text{E}}/2}, \boldsymbol{h}_{t-1}^{L_{\text{D}}})$
            $\boldsymbol{\Omega} = \frac{\exp(\boldsymbol{\kappa})}{\sum_{|\boldsymbol{\kappa}|} \exp(\boldsymbol{\kappa})}$
            $\Lambda_t = \sum_T \Omega \times \boldsymbol{h}_t^{L_{\text{E}}/2}$
            $L_{\text{A}}$ outputs $\Lambda_t$
            Latent Layer ($L_{\text{S}}$):
            $\breve{\boldsymbol{X}} = \sum(\Lambda_t, \tilde{\boldsymbol{X}})$
            Graph Decoder ($\boldsymbol{D}$):
            for each graph decoding layer $l_{\text{D}} \in \mathcal{L}_{\text{D}}$ do
                $\breve{\boldsymbol{X}}^{l_{\text{D}}} = \text{ReLU}(\boldsymbol{\mu}^{l_{\text{D}}} *_{\mathcal{G}} \breve{\boldsymbol{X}}^{l_{\text{D}}-1} + \boldsymbol{b}^{l_{\text{D}}})$
                for each timestamp $t$ do
                    Compute:
                    $\boldsymbol{i}_t^{l_{\text{D}}}$, $\boldsymbol{o}_t^{l_{\text{D}}}$, $\boldsymbol{f}_t^{l_{\text{D}}}$, $\boldsymbol{s}_t^{l_{\text{D}}}$, and $\boldsymbol{h}_t^{l_{\text{D}}}$
                end
                $\tilde{\boldsymbol{X}}^{l_{\text{D}}} = \text{ReLU}(\boldsymbol{\mu}^{l_{\text{D}}} *_{\mathcal{G}} \breve{\boldsymbol{X}}^{l_{\text{D}}-1} + \boldsymbol{b}^{l_{\text{D}}})$
            end
            Back Propagation:
            Compute:
            $\min_{\Phi} C(\boldsymbol{X}, g_{\Phi}(f_{\Phi}(\boldsymbol{X}))), \quad \boldsymbol{X} \in \boldsymbol{X}_{\text{TR}}$
            Find the derivatives:
            $\nabla_{\boldsymbol{\mu}^{l_{(\cdot)}}} C$, $\nabla_{\boldsymbol{b}^{l_{(\cdot)}}} C$, $\nabla_{\boldsymbol{W}_{(\cdot)}^{l_{(\cdot)}}} C$, $\nabla_{\boldsymbol{U}_{(\cdot)}^{l_{(\cdot)}}} C$, and
              $\nabla_{\boldsymbol{V}_{(\cdot)}^{l_{(\cdot)}}} C$
        end
        Parameters Update:
          $\boldsymbol{\mu}^{l_{(\cdot)}} = \boldsymbol{\mu}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{\mu}^{l_{(\cdot)}}} C$
          $\boldsymbol{b}^{l_{(\cdot)}} = \boldsymbol{b}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{b}^{l_{(\cdot)}}} C$
        $\boldsymbol{W}_{(\cdot)}^{l_{(\cdot)}} = \boldsymbol{W}_{(\cdot)}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{W}_{(\cdot)}^{l_{(\cdot)}}} C$
        $\boldsymbol{U}_{(\cdot)}^{l_{(\cdot)}} = \boldsymbol{U}_{(\cdot)}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{U}_{(\cdot)}^{l_{(\cdot)}}} C$
        $\boldsymbol{V}_{(\cdot)}^{l_{(\cdot)}} = \boldsymbol{V}_{(\cdot)}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{V}_{(\cdot)}^{l_{(\cdot)}}} C$
    end
end
Output: Optimal $\boldsymbol{\mu}^{l_{(\cdot)}}$, $\boldsymbol{b}^{l_{(\cdot)}}$, $\boldsymbol{W}_{(\cdot)}^{l_{(\cdot)}}$, $\boldsymbol{U}_{(\cdot)}^{l_{(\cdot)}}$, and $\boldsymbol{V}_{(\cdot)}^{l_{(\cdot)}}$
```

A. Benchmark Detectors

The benchmarks are diverse, with shallow/deep structures, static/dynamic mechanisms, supervised/unsupervised training, and topology-aware/topology-unaware detection.

1) Topology-unaware Detection: The detectors listed herein are topology-unaware, which means that they do not capture the information about the topology $\boldsymbol{\Gamma}$ (i.e., do not capture the spatial aspects and relationships within the measurement data implied by the power system topology).

a) Shallow Detectors: Autoregressive integrated moving average (ARIMA) presents an unsupervised dynamic model that uses measurement data of normal operation to predict succeeding data patterns, and during testing, if a sample surpasses a defined threshold ψ, an abnormal operation is flagged [43]. SVM is a supervised static classifier trained on labeled benign and abnormal samples that classifies samples by defining a decision boundary to separate both classes [7].

b) Deep Detectors: FNN is a supervised static model trained on operation data from normal and abnormal patterns; it uses stacked hidden layers with fully-connected neurons and forward information flow [11]. LSTM is a supervised dynamic model that operates on normal and abnormal data using recurrent cells with recurrent information flow and thus holds previous knowledge [44]. CNN is a supervised model where feature extraction is carried out via the convolution operation [16]. The SEL-based model [45] presents an unsupervised dynamic model that employs an autoencoder with attention [42] to learn patterns of normal data operation then performs further processes using recurrent and fully-connected layers where the detection is based on a comparison between the reconstruction error ζ and the detection threshold ψ.

2) Topology-aware Detection: We adopt a C-GNN model [5] to present a benchmark topology-aware detector. The C-GNN model is supervised, and it is trained and tested on benign and malicious data. C-GNN models utilize multiple stacked graph convolution layers that capture the graphs' spatial features through the graph convolution operation [46]. The Chebyshev graph convolution layers are followed by a dense layer to estimate the attack probability of a given sample. The decision is then presented in the output layer accordingly.

B. Hyperparameters Selection

We perform a hyperparameter selection process to employ the optimal sets of hyperparameters that provide the best detection performance (i.e., highest DR and ACC with lowest FAR) against $\boldsymbol{X}_{\text{VAL}}$. Specifically, we perform a multi-stage sequential grid-search where a hyperparameter is selected from a selection space $\mathcal{P}$ throughout each stage [47].

1) Selection Space: Let ϱ define the value of the optimal hyperparameter that is selected from $\mathcal{P}$ for each of the following hyperparameters. Number of layers $\mathcal{L} = \{2, 3, 4, 5, 6, 8\}$, number of units $\mathcal{U} = \{4, 8, 16, 32, 64\}$, dropout rate $\mathcal{D} = \{0, 0.2, 0.4, 0.5\}$, order of neighborhood $\mathcal{K} = \{2, 3, 4, 5\}$, optimizer $\mathcal{O} = \{\text{Adam, Adamax, SGD, Rmsprop}\}$, and activation function $\mathcal{A} = \{\text{Sigmoid, Tanh, Relu, Elu}\}$.

TABLE I
OPTIMAL HYPERPARAMETERS OF DEEP DETECTORS

Detector	ϱ	14-bus	39-bus	118-bus
FNN	L	5	4	4
	U	16	32	32
	D	0	0.2	0
	O	Adam	SGD	Adam
	A	Relu	Relu	Relu
LSTM	L	3	2	3
	U	16	16	32
	D	0.2	0	0.2
	O	SGD	Adam	Adam
	A	Relu	Relu	Relu
CNN	L	4	4	4
	U	32	32	32
	K	5	5	5
	O	Adam	Adam	Rmsprop
	A	Relu	Relu	Relu
SEL	L	6	4	8
	U	32	32	32
	D	0	0	0.2
	O	Adam	Adam	SGD
	A	Sigmoid	Sigmoid	Sigmoid
C-GNN	L	5	4	5
	U	16	16	32
	K	3	5	4
	O	Adam	Rmsprop	Rmsprop
	A	Relu	Relu	Relu
Proposed CR-GAE	L	6	6	6
	U	32	64	64
	K	4	4	5
	O	Rmsprop	Adam	Adam
	A	Relu	Relu	Relu

2) *Optimal Hyperparameters:* After performing sequential grid-search, the hyperparameters listed in Table I turn out to be the optimal hyperparameters for the deep detectors. For the autoencoder-based models (SEL and CR-GAE), the reported U denotes the number of units in the first encoding layer. Thus, with L and U of 4 and 32, respectively, the model presents $(32, 16)$ and $(16, 32)$ units in the encoder and decoder layers, respectively. For shallow models, ARIMA presents the optimal values of 0 and 1 for the moving average and differencing degree, respectively, using $\{0,\ 1,\ 2,\ 3\}$ as $\mathcal{P}$. For the SVM model, $\mathcal{P}$ is taken from kernel = {Linear, Sigmoid, rbf}, gamma = {scale, auto}, and regularization = $\{1,\ 10,\ 100\}$, which turn out to be Sigmoid, auto, and 1, respectively.

C. Optimal Detection Threshold Values (ψ)

As discussed in Section III-C, unsupervised detectors use a detection threshold ψ that separates benign from malicious samples. When scores (MSE for ARIMA, and ζ for SEL and CR-GAE) are greater than ψ, a malicious sample is assigned the label $y = 1$; the sample is deemed benign otherwise. Using $\boldsymbol{X}_{\text{VAL}}$, the ψ values are found to be $\psi = 0.42$, $\psi = 0.53$, and $\psi = 0.55$ for the ARIMA, SEL, and CR-GAE, respectively.

D. Experimental Results

In this section, we introduce the used evaluation metrics to measure and report the impact of data poisoning on the detection performance of the investigated detectors in generalized and topology-specific settings.

1) *Evaluation Metrics:* To quantify the attack impact, we report the performance of the detectors using three performance metrics. First, to determine how well the model identifies malicious samples, we use detection rate (DR = TP/(TP + FN)), where TP and FN depict true positive and false negative samples, respectively. Second, to indicate the percentage of benign samples that were incorrectly detected as malicious, we use false alarm rate (FAR = FP/(FP + TN)), where FP and TN depict false positive and true negative samples, respectively. Third, to determine how well the model marks benign and malicious samples correctly, we use accuracy (ACC = (TP + TN)/(TP + TN + FP + FN)).

2) *Impact of Data Poisoning:* In Tables II and III, we report the impact of data poisoning against the investigated detectors in generalized and topology-specific settings, respectively. The reported detection performance follows the injection levels presented in Section II-D5.

a) Impact on Generalized Detectors: In Table II, we report the impact of data poisoning on the benchmark and proposed detectors in generalized settings (as per Section II-E1). The impact on the benchmark topology-unaware detectors (ARIMA, SVM, FNN, LSTM, CNN, and SEL), benchmark topology-aware detector (C-GNN), and proposed topology-aware (GR-GAE) detector in the presence of multiple injection levels for each system size is summarized next.

- For the 14-bus system, for topology-unaware benchmarks, the DRs deteriorate by $3.1 - 6.8\%$, $7.7 - 15.6\%$, and $13.8 - 26.4\%$ with 10%, 20%, and 30% data poisoning injection levels. For the topology-aware benchmark, the DRs deteriorate by 2.3%, 5.7%, and 10.2% with 10%, 20%, and 30% injection levels. For the proposed detector, the DRs deteriorate only by 0.6%, 1.6%, and 3.2% with 10%, 20%, and 30% data poisoning injection levels.
- For the 39-bus system, for topology-unaware benchmarks, the DRs deteriorate by $2.5 - 5.8\%$, $6.4 - 13.3\%$, and $11.8 - 22.5\%$ with 10%, 20%, and 30% data poisoning injection levels. For the topology-aware benchmark, the DRs deteriorate by 2.2%, 5.4%, and 9.6% with 10%, 20%, and 30% injection levels. For the proposed detector, the DRs deteriorate only by 0.5%, 1.4%, and 2.6% with 10%, 20%, and 30% data poisoning injection levels.
- For the 118-bus system, for topology-unaware benchmarks, the DRs deteriorate by $2 - 4.6\%$, $5.3 - 10.8\%$, and $9.9 - 18.2\%$ with 10%, 20%, and 30% adversarial injection levels. For the topology-aware benchmark, the DRs deteriorate by 1.8%, 4.8%, and 8.9% with 10%, 20%, and 30% injection levels. For the proposed detector, the DRs deteriorate marginally by 0.3%, 0.8%, and 1.6% with 10%, 20%, and 30% adversarial injection levels.

b) Impact on Topology-Specific Detectors: In Table III, we report the impact of data poisoning on the investigated detectors in topology-specific settings (as per Section II-E2). For the 14-bus system, the DRs of the shallow, deep, and topology-aware benchmarks deteriorate by $26.6 - 28.7\%$, $15.6 - 20.7\%$, and 11.4% with the highest injection levels, respectively, whereas the degradation is only 3.7% for the proposed detector. For the 39-bus system, the DRs of the shallow, deep, and topology-aware benchmarks deteriorate by

TABLE II
DATA POISONING IMPACT ON GENERALIZED DETECTORS (%)

System Size	Detector	Metric	Poisoning percentage			
			0	10	20	30
14-bus	ARIMA	DR	61.3	54.5	45.7	34.9
		FAR	46.4	53.0	61.5	71.9
		ACC	60.1	53.4	44.9	34.6
	SVM	DR	64.2	58.0	49.9	39.7
		FAR	38.6	44.7	52.7	62.6
		ACC	63.0	56.7	48.5	38.5
	FNN	DR	68.9	64.3	58.0	49.8
		FAR	31.2	35.8	42.1	50.2
		ACC	67.5	62.9	56.5	48.4
	LSTM	DR	74.4	70.0	63.9	56.1
		FAR	24.9	29.1	35.0	42.6
		ACC	72.7	68.5	62.5	54.8
	CNN	DR	77.8	74.5	69.6	63.0
		FAR	19.3	22.8	27.9	34.6
		ACC	77.5	74.2	69.2	62.6
	SEL	DR	82.8	79.7	75.1	69.0
		FAR	14.5	17.5	22.1	28.1
		ACC	82.0	78.9	74.2	68.0
	C-GNN	DR	89.7	87.4	84.0	79.5
		FAR	7.9	10.3	13.8	18.5
		ACC	88.7	86.4	82.9	78.3
	Proposed CR-GAE	DR	**93.2**	**92.6**	**91.6**	**90.0**
		FAR	**5.0**	**5.6**	**6.8**	**8.4**
		ACC	**92.9**	**92.2**	**91.1**	**89.4**
39-bus	ARIMA	DR	62.9	57.1	49.6	40.4
		FAR	44.8	50.5	58.1	67.4
		ACC	62.1	56.4	49.0	39.8
	SVM	DR	66.0	60.8	53.9	45.2
		FAR	36.7	41.8	48.7	57.4
		ACC	65.0	59.8	52.8	44.1
	FNN	DR	70.7	66.6	60.9	53.6
		FAR	29.7	33.8	39.5	46.8
		ACC	69.4	65.3	59.6	52.3
	LSTM	DR	76.0	72.1	66.5	59.3
		FAR	23.3	27.1	32.5	39.5
		ACC	74.6	70.8	65.5	58.5
	CNN	DR	79.8	77.1	73.0	67.4
		FAR	17.7	20.5	24.8	30.5
		ACC	78.6	75.8	71.6	65.9
	SEL	DR	84.6	82.1	78.2	72.8
		FAR	12.6	15.0	18.9	24.2
		ACC	83.5	81.0	77.0	71.6
	C-GNN	DR	91.8	89.6	86.4	82.2
		FAR	6.1	8.2	11.3	15.3
		ACC	91.2	89.2	86.3	82.5
	Proposed CR-GAE	DR	**95.4**	**94.9**	**94.0**	**92.8**
		FAR	**2.9**	**3.4**	**4.3**	**5.7**
		ACC	**94.8**	**94.2**	**93.2**	**91.8**
118-bus	ARIMA	DR	66.1	61.5	55.4	47.9
		FAR	42.1	46.6	52.6	60.2
		ACC	65.3	60.7	54.6	46.9
	SVM	DR	69.3	65.1	59.4	52.2
		FAR	33.7	37.8	43.4	50.6
		ACC	68.5	64.5	58.9	51.8
	FNN	DR	74.6	71.0	66.0	59.7
		FAR	26.3	29.8	34.8	41.2
		ACC	72.9	69.3	64.3	57.8
	LSTM	DR	80.1	77.1	72.7	66.8
		FAR	20.1	23.2	27.7	33.6
		ACC	78.3	75.2	70.6	64.6
	CNN	DR	84.0	81.9	78.5	73.8
		FAR	14.3	16.5	20.0	24.9
		ACC	83.1	80.9	77.4	72.6
	SEL	DR	87.9	85.9	82.6	78.0
		FAR	10.1	12.0	15.1	19.5
		ACC	87.2	85.1	81.7	77.0
	C-GNN	DR	96.1	94.3	91.3	87.2
		FAR	2.6	4.4	7.2	11.1
		ACC	95.9	94.1	91.2	87.3
	Proposed CR-GAE	DR	**99.1**	**98.8**	**98.3**	**97.5**
		FAR	**0.3**	**0.7**	**1.3**	**2.2**
		ACC	**98.8**	**98.3**	**97.5**	**96.5**

TABLE III
DATA POISONING IMPACT ON TOPOLOGY-SPECIFIC DETECTORS (%)

System Size	Detector	Metric	Poisoning percentage			
			0	10	20	30
14-bus	ARIMA	DR	48.9	41.5	31.9	20.2
		FAR	58.5	65.7	75.0	86.4
		ACC	48.3	41.0	31.7	20.4
	SVM	DR	52.1	45.3	36.5	25.5
		FAR	50.7	57.4	66.2	77.1
		ACC	51.0	44.1	35.1	24.1
	FNN	DR	56.9	51.9	45.0	36.2
		FAR	43.0	48.0	54.9	63.8
		ACC	55.7	50.7	43.7	34.8
	LSTM	DR	62.7	57.9	51.2	42.7
		FAR	36.8	41.4	47.9	56.3
		ACC	61.2	56.6	50.0	41.5
	CNN	DR	67.4	63.7	58.2	50.8
		FAR	29.9	33.8	39.5	47.0
		ACC	67.0	63.3	57.7	50.3
	SEL	DR	75.0	71.5	66.3	59.4
		FAR	21.3	24.7	29.9	36.7
		ACC	74.0	70.5	65.2	58.2
	C-GNN	DR	82.1	79.5	75.7	70.7
		FAR	16.1	18.8	22.7	27.9
		ACC	81.4	78.8	75.0	70.1
	Proposed CR-GAE	DR	**86.3**	**85.6**	**84.4**	**82.6**
		FAR	**12.4**	**13.1**	**14.4**	**16.2**
		ACC	**85.9**	**85.1**	**83.8**	**81.9**
39-bus	ARIMA	DR	49.9	43.5	35.2	25.1
		FAR	57.7	64.0	72.4	82.7
		ACC	49.1	42.8	34.6	24.4
	SVM	DR	53.2	47.4	39.7	30.0
		FAR	49.6	55.3	63.0	72.7
		ACC	52.3	46.5	38.7	29.0
	FNN	DR	57.6	53.1	46.8	38.6
		FAR	42.7	47.2	53.5	61.6
		ACC	56.7	52.2	45.9	37.8
	LSTM	DR	63.3	59.0	52.8	44.8
		FAR	36.0	40.2	46.2	54.0
		ACC	62.1	57.9	52.0	44.2
	CNN	DR	68.4	65.3	60.6	54.3
		FAR	29.2	32.4	37.3	43.8
		ACC	68.0	64.8	60.0	53.5
	SEL	DR	75.9	73.0	68.5	62.3
		FAR	20.5	23.3	27.8	33.8
		ACC	74.7	71.8	67.2	61.0
	C-GNN	DR	83.9	81.5	78.0	73.4
		FAR	14.0	16.2	19.5	23.8
		ACC	83.1	80.9	77.8	73.8
	Proposed CR-GAE	DR	**88.2**	**87.6**	**86.5**	**85.1**
		FAR	**10.2**	**10.8**	**11.9**	**13.5**
		ACC	**87.5**	**86.8**	**85.7**	**84.1**
118-bus	ARIMA	DR	52.3	47.1	40.2	31.6
		FAR	56.5	61.6	68.4	77.0
		ACC	50.7	45.5	38.6	29.9
	SVM	DR	55.4	50.6	44.1	35.9
		FAR	47.8	52.5	59.0	67.3
		ACC	53.9	49.3	42.9	34.8
	FNN	DR	60.6	56.6	51.0	43.8
		FAR	40.4	44.3	49.9	57.1
		ACC	58.3	54.3	48.7	41.4
	LSTM	DR	66.4	63.0	58.0	51.3
		FAR	33.6	37.1	42.2	48.9
		ACC	64.7	61.2	56.1	49.5
	CNN	DR	71.6	69.1	65.1	59.7
		FAR	26.7	29.3	33.3	38.9
		ACC	70.9	68.3	64.2	58.7
	SEL	DR	78.1	75.7	71.9	66.7
		FAR	18.9	21.1	24.7	29.7
		ACC	77.5	75.0	71.0	65.5
	C-GNN	DR	87.3	85.4	82.3	78.1
		FAR	11.7	13.6	16.6	20.8
		ACC	87.1	85.2	82.2	78.2
	Proposed CR-GAE	DR	**91.7**	**91.3**	**90.7**	**89.7**
		FAR	**7.5**	**8.0**	**8.8**	**9.9**
		ACC	**91.4**	**90.8**	**89.9**	**88.7**

$23.2-24.8\%$, $13.6-19\%$, and 10.5% with the highest injection levels, respectively, whereas the degradation is only 3.1% for the proposed detector. For the 118-bus system, the DRs of the shallow, deep, and topology-aware benchmarks deteriorate by $19.4-20.7\%$, $11.4-16.8\%$, and 9.2% with the highest injection levels, respectively, whereas the degradation is only 2% for the proposed detector.

3) Remarks: The following remarks are drawn from the aforementioned results.

- Generalized detection offers more robust detection performance than topology-specific detection by $7.8-16.3\%$ in DR. Specifically, the DRs of generalized shallow, deep, C-GNN, and proposed CR-GAE detectors offer lower degradation rates by $12.1-16.3\%$, $7.8-15.5\%$, $7.6-9.1\%$, and $6.9-7.8\%$, respectively, compared to topology-specific training. This is because generalized detectors utilize more data from different topologies, allowing them to capture the topological reconfigurations. However, the topology-specific detectors are only trained on a static topological configuration, and hence, do not have the generalization ability.
- The topology-aware benchmark detector (C-GNN) offers better DR than the shallow and deep topology-unaware benchmark detectors by $8.2-17.3\%$ and $2-9.3\%$, respectively, with highest injection levels. Such an improvement is because the C-GNN model captures the spatial aspect within the measurement data.
- The proposed CR-GAE detector offers the most stable detection performance with only $0.3-3.7\%$ DR degradation against data poisoning. Specifically, its DR outperforms the shallow, deep, and C-GNN detectors by $15.5-25\%$, $8.3-17\%$, and $7-7.7\%$, respectively, with highest injection levels. Such superior performance is because the proposed CR-GAE detector offers generalized topology-aware detection that captures the spatial, temporal, and complex nature of the measurement data.
- As system size increases, the detection performance improves. Specifically, in 118-bus systems, shallow, deep, C-GNN, and proposed CR-GAE detectors offer more robust DR of $7.1-8.2\%$, $3.9-4.2\%$, $1.5-2.2\%$, and 1.7%, respectively, compared to 14-bus systems. Also, in the 118-bus systems, shallow, deep, GNN, and proposed CR-GAE detectors offer more robust degradation rates of $3.7-4.3\%$, $1.9-2.2\%$, $1-1.3\%$, and 1.1%, respectively, compared to 39-bus systems with highest injection levels. This is because the 118-bus system offers more measurement data to train, and hence, allowing the detectors to capture more distinctive features from the data.

4) Impact of Each Attack Function: Tables II and III reported the average deterioration rates of all attacks at different injection levels. Tables IV and V show the impact on DR of each attack function (1) - (6) separately at the highest injection levels (30% data poisoning). The results show that the attacks lead to different deterioration levels depending on their strength. Compared to the average deterioration rates, the random (1), random replay (4), and one-step replay (3) attacks lead to lower deterioration rates, whereas the general (2), interval replay (5), and strategic replay (6) attacks present

TABLE IV
IMPACT OF EACH ATTACK ON DR OF GENERALIZED DETECTORS (%)

Sys. Size	Detector	Attack function						Avg.
		(1)	(2)	(3)	(4)	(5)	(6)	
14-bus	ARIMA	42.4	32.4	39.9	39.1	28.4	27.1	34.9
	SVM	47.0	36.5	44.1	46.4	33.2	31.1	39.7
	FNN	56.5	46.6	53.9	51.4	45.6	44.8	49.8
	LSTM	62.8	53.4	60.5	62.6	49.5	48.1	56.1
	CNN	69.2	60.2	67.0	65.9	58.4	57.4	63.0
	SEL	74.0	66.4	72.4	73.0	65.1	63.3	69.0
	C-GNN	83.7	76.8	82.5	81.6	76.5	76.1	79.5
	Proposed CR-GAE	**93.0**	**88.0**	**91.9**	**92.5**	**87.2**	**87.1**	**90.0**
39-bus	ARIMA	46.9	37.7	44.8	46.1	33.9	33.2	40.4
	SVM	51.6	42.5	49.1	50.1	39.4	38.5	45.2
	FNN	59.3	50.7	57.3	58.6	48.2	47.2	53.6
	LSTM	65.1	56.9	63.3	63.8	54.4	52.4	59.3
	CNN	72.7	64.7	70.9	72.8	62.7	60.8	67.4
	SEL	77.2	70.3	75.8	75.8	69.3	68.3	72.8
	C-GNN	86.2	79.9	85.2	85.4	78.8	77.9	82.2
	Proposed CR-GAE	**94.9**	**90.8**	**94.8**	**94.9**	**90.6**	**90.5**	**92.8**
118-bus	ARIMA	54.1	45.1	51.9	53.2	41.9	40.9	47.9
	SVM	58.4	49.5	56.1	57.1	46.4	45.5	52.2
	FNN	65.4	56.8	63.3	64.7	54.4	53.3	59.7
	LSTM	72.2	64.2	70.4	71.7	62.0	60.0	66.8
	CNN	78.7	71.2	76.9	78.7	69.5	67.5	73.8
	SEL	82.2	75.6	80.9	80.9	74.6	73.6	78.0
	C-GNN	91.0	84.9	89.8	90.3	84.1	83.4	87.2
	Proposed CR-GAE	**99.0**	**95.9**	**98.9**	**98.9**	**96.1**	**95.9**	**97.5**

TABLE V
IMPACT OF EACH ATTACK ON DR OF TOPOLOGY-SPECIFIC DETECTORS (%)

Sys. Size	Detector	Attack function						Avg.
		(1)	(2)	(3)	(4)	(5)	(6)	
14-bus	ARIMA	27.6	17.4	25.2	25.8	13.3	12.1	20.2
	SVM	32.8	22.4	29.9	32.8	18.7	16.6	25.5
	FNN	43.0	32.7	40.3	38.8	31.7	30.9	36.2
	LSTM	49.5	39.7	47.2	49.3	35.8	34.4	42.7
	CNN	57.2	47.8	55.0	53.9	45.8	45.1	50.8
	SEL	64.6	56.6	63.1	63.6	55.3	53.4	59.4
	C-GNN	75.1	67.8	73.9	73.0	67.5	67.1	70.7
	Proposed CR-GAE	**85.8**	**80.1**	**85.0**	**84.7**	**79.9**	**79.8**	**82.6**
39-bus	ARIMA	31.7	22.3	29.6	30.9	18.3	17.6	25.1
	SVM	36.6	27.1	34.1	35.1	24.0	23.1	30.0
	FNN	44.6	35.5	42.6	43.9	33.1	32.1	38.6
	LSTM	50.8	42.2	49.0	49.5	39.7	37.7	44.8
	CNN	59.8	51.4	58.0	59.7	49.8	47.3	54.3
	SEL	66.9	59.6	65.6	65.5	58.6	57.6	62.3
	C-GNN	77.6	70.9	76.6	76.8	69.8	68.9	73.4
	Proposed CR-GAE	**87.9**	**82.9**	**87.3**	**87.4**	**82.7**	**82.6**	**85.1**
118-bus	ARIMA	38.1	28.7	35.9	37.2	25.5	24.4	31.6
	SVM	42.3	33.1	40.0	41.2	29.9	29.0	35.9
	FNN	49.8	40.8	47.7	49.1	38.3	37.3	43.8
	LSTM	57.0	48.6	55.2	56.5	46.4	44.3	51.3
	CNN	64.9	56.8	63.1	65.1	55.3	53.2	59.7
	SEL	71.1	64.1	69.8	69.8	63.1	62.1	66.7
	C-GNN	82.1	75.6	80.9	81.4	74.7	74.1	78.1
	Proposed CR-GAE	**91.6**	**87.7**	**91.6**	**91.5**	**87.6**	**87.9**	**89.7**

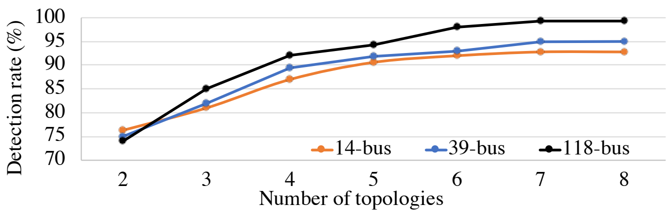
TABLE VI
IMPACT OF OTHER ADVERSARIAL ATTACKS ON THE DETECTORS (%)

Detector	Metric	System Size		
		18-bus	39-bus	118-bus
ARIMA	DR	37.3	39.2	42.6
	FAR	69.6	67.8	65.0
	ACC	36.0	38.2	41.5
SVM	DR	42.1	44.2	48.9
	FAR	60.5	58.6	55.5
	ACC	41.7	43.5	48.2
FNN	DR	50.0	51.7	55.7
	FAR	48.9	48.2	44.7
	ACC	50.3	51.1	53.7
LSTM	DR	57.9	59.7	64.1
	FAR	41.9	40.5	36.1
	ACC	53.3	59.1	60.3
CNN	DR	62.1	64.2	69.7
	FAR	33.4	32.6	30.2
	ACC	62.0	63.3	69.2
SEL	DR	71.5	73.4	76.8
	FAR	25.5	23.3	20.7
	ACC	70.9	72.5	76.4
C-GNN	DR	80.2	82.4	83.9
	FAR	17.0	15.1	13.8
	ACC	79.5	82.1	83.3
Proposed CR-GAE	DR	**89.7**	**92.1**	**96.3**
	FAR	**8.6**	**6.2**	**3.4**
	ACC	**89.5**	**90.9**	**95.9**

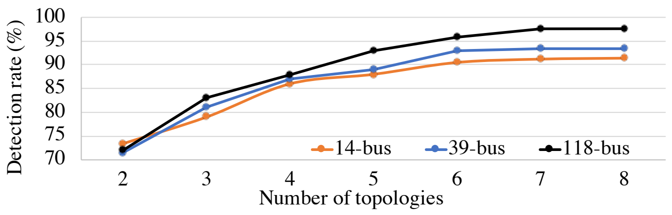
stronger attacks that lead to higher deterioration rates. Specifically, in benchmark detectors, the random, random replay, and one-step replay attacks lead to DR deterioration rates of 5.1–21.3%, 5.3–23.1%, and 6.3–23.7%, respectively, whereas the general, interval replay, and strategic replay attacks lead to DR deterioration rates of 11.3–31.5%, 12.1–35.6%, and 12.8–36.8%, respectively. The proposed CR-GAE offers stable DR that degrades only by 0.1–0.5%, 0.2–1.6%, 0.1–2%, 3.2–6.2%, 3–6.4%, and 3.2–6.5% when subject to random, random replay, one-step replay, general, interval replay, and strategic replay attacks, respectively.

5) Robustness Against Other Adversarial Attacks: Besides the FDIA functions discussed in Section II-D1, the model could encounter other adversarial attacks such as the fast gradient sign method [48], DeepFool [49], and basic iterative method [50]. Although such attacks are typically considered as evasion attacks that are present in the test set [45], we are considering them herein as data poisoning to reflect cases where they have not been detected before and thus the model is falsely being trained on them as benign samples. Table VI reports the impact of such adversarial attacks at the highest injection levels (30% data poisoning) on the generalized detectors since they exhibited superior detection performance in Table II compared to topology-specific detectors in Table III. Due to the advantages discussed in Section IV-D3 that the proposed CR-GAE detector offers, it still presents a stable DR with around 3.5% degradation, offering superior DR by $18.2 - 53.7\%$ and $9.5 - 12.4\%$ compared to topology-unaware and topology-aware benchmark detectors, respectively, against such adversarial attacks.

6) Model Analysis: After discussing the experimental environment and offline training time, we perform analyses related to generalization ability (i.e., ideal number of required topologies to offer generalized detection) as well as the testing



(a) Proposed CR-GAE model



(b) C-GNN model

Fig. 4. DR of the topology-aware models as training topologies increases.

time and scalability of the models.

a) Experimental Environment: The training process of the proposed and benchmark models is carried out offline utilizing an NVIDIA GeForce RTX 2070 hardware accelerator adopting Keras sequential API. Offline training takes between two to four hours and five to seven hours for the topology-unaware and topology-aware detectors, respectively. The expansion in the training time frame herein is directly proportional to the amount of training features. Topology-aware models require a comparatively prolonged training period as they capture a more comprehensive set of features, including the spatial characteristics of topological configurations. This results in improved detection performance but comes at the cost of an extended training duration. Fortunately, model training time and computation capabilities of the smart grid network do not present a significant concern as system operators can conduct the offline training on available datasets periodically (weekly or monthly). Testing is done online (real time) as per the reported decision time in Section IV-D6c.

b) Generalization Ability: As shown in Section IV-D, generalized detection offers superior detection performance compared to topology-specific detection. Thus, we study the number of topologies that are required in order for the detectors to achieve the generalization ability. In Fig. 4, we plot the increase of DR with the increase of the number of training topologies Γ for the proposed CR-GAE compared to the C-GNN topology-aware benchmark detector in generalized settings. Based on the plots, we conclude that the ideal total number of topologies to offer generalized detection is eight. Specifically, since the reported DR keeps improving from two to seven topologies before it saturates in the eighth topology, we train on seven topologies, validate on the eighth, and then the tests are conducted on the two remaining unseen topologies. This allows us to avoid over-fitting and over-saturating when training the models [41].

c) Scalability: We analyze the model scalability with respect to the required time for decision making regarding a given sample as the size of the system gets larger. To perform

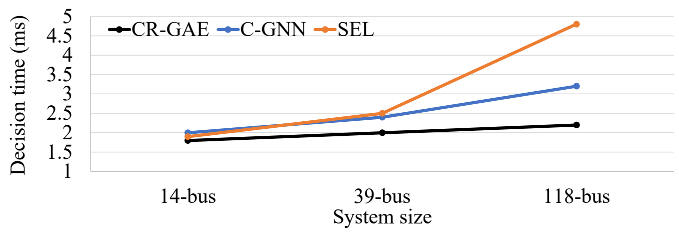


Fig. 5. Illustration of the linear scalability of the proposed model.

such analysis, in Fig. 5, we plot the decision time in milliseconds (ms) for each of the 14, 39, and 118-bus systems in generalized settings. We compare the plot of the proposed CR-GAE with the C-GNN and SEL models. We select these two benchmarks since they present the top performing topology-aware (C-GNN) and topology-unaware (SEL) benchmarks as per Tables II and III. Fig. 5 shows that the proposed CR-GAE detector offers higher linear scalability compared to the rest of the benchmarks as the system size increases. Thus, despite the higher complexity with bigger systems, the proposed CR-GAE model still offers linear scalability with the required decision time. Such linear scalability is because the proposed CR-GAE model presents topology-aware detection that captures the spatial aspects within the measurement data, and thus, being fed with more data to capture within bigger systems enhances the detection performance. The online (real time) testing of the proposed CR-GAE detector takes around 2 ms, which satisfies the latency requirements and computation capabilities of the smart grid network.

V. Conclusion and Future Work

This paper investigated the impact of adversarial data poisoning on data-driven FDIAs detectors in generalized and topology-specific settings through different attack injection levels. Based on our extensive simulation studies, we found out that data poisoning led to up to 29% DR degradation in benchmark detectors and reached the following. First, generalized detectors trained on multiple topological reconfigurations are $8-16\%$ more robust in DR than topology-specific detectors. Second, the topology-aware benchmark model (C-GNN) is $2-17\%$ more robust in DR than topology-unaware benchmarks (shallow and deep classical models) since it captures the spatial aspects within the data. However, it does not capture unseen attacks due to its supervised training nature. Third, our proposed CR-GAE detector offers superior DR that degrades only by $0.3-3.7\%$, offering $7-25\%$ DR enhancement compared to benchmarks with the highest data poisoning injection levels. The proposed CR-GAE detector offers generalized detection that is trained and tested on multiple topologies from 14, 39, and 118-bus systems, and hence, it offers detection of unseen attacks within unseen topologies due to its unsupervised training nature. Specifically, the proposed CR-GAE detector employs an autoencoder with Chebyshev graph convolution recurrent layers with attention mechanism to capture the spatial and temporal correlations within measurement data. This work focused on detecting FDIAs and analyzed the robustness of detectors against data poisoning on the graph level. Our future work will investigate the localization and classification of FDIAs on the node level.

References

[1] Z. Zhang *et al.*, "Cyber-physical coordinated risk mitigation in smart grids based on attack-defense game," *IEEE Trans. on Power Systems*, vol. 37, no. 1, pp. 530–542, Jan. 2022.
[2] Y. Liu *et al.*, "False data injection attacks against state estimation in electric power grids," *ACM Trans. on Information and System Security*, vol. 14, no. 1, pp. 1–33, Jun. 2011.
[3] X. Yin *et al.*, "A subgrid-oriented privacy-preserving microservice framework based on deep neural network for false data injection attack detection in smart grids," *IEEE Trans. on Industrial Informatics*, vol. 18, no. 3, pp. 1957–1967, Mar. 2022.
[4] K. Huang *et al.*, "False data injection attacks detection in smart grid: A structural sparse matrix separation method," *IEEE Trans. on Network Science and Engineering*, vol. 8, no. 3, pp. 2545–2558, Jul. 2021.
[5] O. Boyaci *et al.*, "Graph neural networks based detection of stealth false data injection attacks in smart grids," *IEEE Systems Journal*, vol. 16, no. 2, pp. 2946–2957, Jun. 2022.
[6] F. Xia *et al.*, "Graph learning: A survey," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 109–127, Apr. 2021.
[7] M. Esmalifalak *et al.*, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1644–1652, Sept. 2017.
[8] X. Lu *et al.*, "False data injection attack location detection based on classification method in smart grid," in *Int. Conf. on AI and Advc Manfct. (AIAM)*. Manchester, United Kingdom, 15–17 Oct. 2020, pp. 133–136.
[9] D. Wang *et al.*, "Detection of power grid disturbances and cyber-attacks based on machine learning," *Journal of Information Security and Applications*, vol. 46, pp. 42–52, Jun. 2019.
[10] A. Takiddin, R. Atat, M. Ismail, O. Boyaci, K. R. Davis, and E. Serpedin, "Generalized graph neural network-based detection of false data injection attacks in smart grids," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 3, pp. 618–630, Jun. 2023.
[11] D. Xue *et al.*, "Detection of false data injection attacks in smart grid utilizing ELM-Based OCON framework," *IEEE Access*, vol. 7, pp. 31 762–31 773, Mar. 2019.
[12] E. M. Ferragut *et al.*, "Real-time cyber-physical false data attack detection in smart grids using neural networks," in *Int. Conf. on Comp. Sci. and Com. Intel (CSCI)*. Las Vegas, NV, USA, 14–16 Dec. 2017.
[13] Y. Wang *et al.*, "Kfrnn: An effective false data injection attack detection in smart grid based on kalman filter and recurrent neural network," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6893–6904, May 2022.
[14] Y. Zhang *et al.*, "Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach," *IEEE Trans. on Smart Grid*, vol. 12, no. 1, pp. 623–634, Jan. 2021.
[15] S. Wang *et al.*, "Locational detection of the false data injection attack in a smart grid: A multilabel classification approach," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8218–8227, Sept. 2020.
[16] G. Zhang *et al.*, "Spatio-temporal correlation-based false data injection attack detection using deep convolutional neural network," *IEEE Trans. on Smart Grid*, vol. 13, no. 1, pp. 750–761, Jan. 2022.
[17] R. Ramakrishna *et al.*, "Detection of false data injection attack using graph signal processing for the power grid," in *IEEE Glbl. Conf. on Sgnl. and Info. Proc. (GSIP)*. Ottawa, ON, Canada, 11–14 Nov. 2019.
[18] E. Drayer and T. Routtenberg, "Detection of false data injection attacks in smart grids based on graph signal processing," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1886–1896, Jun. 2020.
[19] Y. Liang *et al.*, "Poisoning attack on load forecasting," in *IEEE Innov. Smart Grid Tech.* Chengdu, China, 21–24 May 2019, pp. 1230–1235.
[20] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Robust electricity theft detection against data poisoning attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2675–2684, May 2021.
[21] A. Paudice *et al.*, "Detection of adversarial training examples in poisoning attacks through anomaly detection," *arXiv preprint arXiv:1802.03041*, Feb. 2018.
[22] S. Weerasinghe *et al.*, "Defending support vector machines against data poisoning attacks," *IEEE Trans. on Information Forensics and Security*, vol. 16, pp. 2566–2578, Feb. 2021.
[23] J. Zhang *et al.*, "Robustfl: Robust federated learning against poisoning attacks in industrial IoT systems," *IEEE Trans. on Industrial Informatics*, vol. 18, no. 9, pp. 6388–6397, Sept. 2022.

[24] S. H. Elyas and Z. Wang, "Improved synthetic power grid modeling with correlated bus type assignments," *IEEE Trans. on Power Systems*, vol. 32, no. 5, pp. 3391–3402, Sept. 2017.
[25] Y. Ahn and H. Yeo, "An analytical planning model to estimate the optimal density of charging stations for electric vehicles," *PLOS ONE*, vol. 10, pp. 1–26, Nov. 2015.
[26] F. Baouche *et al.*, "Efficient allocation of electric vehicles charging stations: Optimization model and application to a dense urban network," *IEEE Intel. Transportation Sys. Mag.*, vol. 6, no. 3, pp. 33–43, Jul. 2014.
[27] R. Atat *et al.*, "Stochastic geometry-based model for dynamic allocation of metering equipment in spatio-temporal expanding power grids," *IEEE Trans. on Smart Grid*, vol. 11, no. 3, pp. 2080–2091, May. 2020.
[28] D. Deka *et al.*, "Analytical models for power networks: The case of the western u.s. and ercot grids," *IEEE Trans. on Smart Grid*, vol. 8, no. 6, pp. 2794–2802, Nov. 2017.
[29] G. Rolim *et al.*, "Modelling the data aggregator positioning problem in smart grids," in *IEEE Int. Conf. on Comp. and Info. Tech.* Liverpool, UK, 26–28 Oct. 2015, pp. 632–639.
[30] T. Courtat *et al.*, "Mathematics and morphogenesis of cities: A geometrical approach," *Physical Review E*, vol. 83, p. 036106, Mar. 2011.
[31] Z. Wang *et al.*, "Generating statistically correct random topologies for testing smart grid communication and control networks," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 28–39, 2010.
[32] R. D. Zimmerman *et al.*, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb. 2011.
[33] "The Electric Reliability Council of Texas (ERCOT). Backcasted (actual) load profiles-historical (2020, May 10)." [Online]. Available: https://www.ercot.com/mktinfo/loadprofile/alp
[34] A. Takiddin, M. Ismail, and E. Serpedin, "Detection of electricity theft false data injection attacks in smart grids," in *30th European Signal Processing Conference (EUSIPCO)*. Belgrade, Serbia, 29 Aug.–2 Sept. 2022, pp. 1541–1545.
[35] M. A. Hasnat *et al.*, "A graph signal processing framework for detecting and locating cyber and physical stresses in smart grids," *IEEE Transactions on Smart Grid*, vol. 13, no. 5, pp. 3688–3699, Sept. 2022.
[36] C. Sammut and G. I. Webb, Eds., *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2011, pp. 600–601.
[37] C. Stamile *et al.*, *Graph Machine Learning: Take graph data to the next level by applying machine learning techniques and algorithms*. Birmingham, United Kingdom: Packt Publishing, Jun. 2021.
[38] Y. Li *et al.*, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Int. Conf. on Learning Rep. (ICLR)*. Vancouver, BC, Canada, 30 Apr.–3 May. 2018, pp. 1–16.
[39] L. Wu *et al.*, *Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer, Jan. 2022.
[40] L. Ruiz *et al.*, "Invariance-preserving localized activation functions for graph neural networks," *IEEE Trans. on Signal Processing*, vol. 68, pp. 127–141, Nov. 2020.
[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.
[42] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids," *IEEE Systems Journal*, vol. 16, no. 3, pp. 4106–4117, Sept. 2022.
[43] V. Krishna *et al.*, "ARIMA-Based modeling and validation of consumption readings in power grids," in *Critical Information Infrastructures Security*. Springer Intern. Publishing, May 2016, pp. 199–210.
[44] A. Takiddin *et al.*, "Deep autoencoder-based detection of electricity stealth cyberattacks in AMI networks," in *Intl. Symposium on Signals, Circuits and Systems (ISSCS)*. Iasi, Romania, 15–16 Jul. 2021, pp. 1–6.
[45] A. Takiddin, M. Ismail, and E. Serpedin, "Robust data-driven detection of electricity theft adversarial evasion attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 14, no. 1, pp. 663–676, Jan. 2023.
[46] J. Zhou *et al.*, "Sparsity-induced graph convolutional network for semisupervised learning," *IEEE Trans. on Artificial Intelligence*, vol. 2, no. 6, pp. 549–563, Dec. 2021.
[47] A. Takiddin, M. Ismail, M. Nabil, M. M. E. A. Mahmoud, and E. Serpedin, "Detecting electricity theft cyber-attacks in AMI networks using deep vector embeddings," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4189–4198, Sept. 2021.
[48] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572v3*, Mar. 2015.
[49] S.-M. Dezfooli *et al.*, "Deepfool: A simple and accurate method to fool deep neural networks," in *IEEE Conf. on Comp. Vision and Pattern Recognition*. Las Vegas, NV, USA, 27–30 Jun. 2016, pp. 2574–2582.
[50] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533v4*, Feb. 2017.