# Generalized Graph Neural Network-Based Detection of False Data Injection Attacks in Smart Grids

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

*Abstract*—**False data injection attacks (FDIAs) pose a significant threat to smart power grids. Recent efforts have focused on developing machine learning (ML)-based defense strategies against such attacks. However, existing strategies offer limited detection performance since they (a) lack the capability of embedding the spatial aspects of the power system topology in the detection mechanism, (b) offer topology-specific detection that does not generalize well to practical systems with seasonal reconfigurations in their topology, or (c) offer detection based on only seen types of FDIAs present in the training set. Therefore, in this paper, we aim to develop a defense strategy that offers an improved generalization ability and detection performance against unseen attacks. Towards this objective, we propose a graph autoencoder (GAE)-based detection strategy that (a) captures spatio-temporal features of power systems, hence, offering improved detection performance, (b) is trained on comprehensive graphs reflecting various realizations of power system topologies, hence, offering better generalization abilities, and (c) works effectively against unseen FDIAs. The proposed detector is trained and tested on various topological configurations from 14, 39, and 118-bus systems offering detection rates (DRs) of $93.6\%$, $95.7\%$, and $99.1\%$, respectively, when tested against unseen FDIAs and unseen topologies. This presents an improvement of $11.5-30\%$ compared to existing ML-based strategies.**

*Index Terms*—**Cyberattacks, false data injection attacks, graph autoencoder, graph neural network, generalized detection, machine learning, smart power grids.**

## NOMENCLATURE

| | |
|---|---|
| $\boldsymbol{b}^l \in \mathbb{R}^{c_l}$ | Bias term in layer $l$ with $c_l$ channels. |
| $\boldsymbol{D}$ | Graph decoder of the GAE model. |
| $\boldsymbol{E}$ | Graph encoder of the GAE model. |
| $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ | Graph $\mathcal{G}$ with $\mathcal{V}$ vertices, $\mathcal{E}$ edges, and $\boldsymbol{W}$ weights. |
| $P_{ij}$, $Q_{ij}$ | Active and reactive power flow between buses $i$ and $j$. |
| $\boldsymbol{x}_{\text{BEN}}(t,i)$ | Benign sample at timestamp $t$ and bus $i$. |
| $\boldsymbol{x}_{\text{MAL}}(t,i)$ | Malicious sample at timestamp $t$ and bus $i$. |
| $\boldsymbol{\theta}^l \in \mathbb{R}^{K \times c_{l-1} \times c_l}$ | Chebyshev coefficients in layer $l$. |

## I. INTRODUCTION

Smart power grids present complex cyber-physical systems that rely on measurement data exchanged among the power system entities for situational awareness and operational decisions [1]. For proper and efficient decision-making, the integrity of the exchanged measurement data is crucial for the reliability of the power system [2]. A major concern that jeopardizes data integrity is false data injection attacks (FDIAs) where malicious entities alter the measurement data (e.g., manipulate the sensor data) [3]. FDIAs may lead to operational decisions that overload the power system [4], [5]. The challenge herein is that such attacks can be carried out in a stealthy manner and bypass the traditional bad data detection systems [6]. Hence, recent efforts have focused on developing smart defense strategies that can thwart such stealthy attacks.

### *A. Related Work and Limitations*

In the literature, machine learning (ML)-based strategies were proposed to detect FDIAs [7] with promising reported results. These defense strategies rely on either classical ML-based techniques (e.g., shallow models and deep neural networks) or graph-based techniques (e.g., graph signal processing (GSP) and graph neural networks (GNNs)). Other defense strategies were also proposed [8]. Defense strategies are reviewed next.

*1) Classical ML-Based Detection:* Classical ML-based detection includes shallow models and deep neural networks.

*a) Shallow Models:* a support vector machine (SVM)-based detector offered an $82\%$ F1-Score [9]. Furthermore, an FDIA detector based on decision trees reported an $88\%$ F1-Score [10]. Additionally, a random forest-based approach provided a $93\%$ detection rate (DR) [11]. However, these shallow detectors cannot capture the complex patterns in measurement data [8]. Thus, they offer limited detection performance.

*b) Deep Models:* to capture the complex patterns within the measurement data and possibly improve the detection performance, deep neural network-based FDIA detection strategies have been proposed. For instance, a feedforward neural network (FNN)-based detector offered a classification accuracy (ACC) of around $90\%$ [12]. Moreover, a recurrent neural network (RNN)-based detector offered a $96\%$ DR [13]. Besides, an unsupervised autoencoder-based detector provided a $96\%$ DR [14]. Also, a detector based on multi-layer perceptron offered a detection accuracy of $99\%$ [15]. Furthermore, two detectors based on convolutional neural networks (CNNs) reported detection accuracy of $93\%$ [16] and $99\%$ [17].

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

Rachad Atat is with the Electrical & Computer Engineering Department, Texas AM University Qatar, Doha 23874, Qatar (email: rachad.atat@qatar.tamu.edu).

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

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

Two remarks are highlighted herein. First, the aforementioned detectors assumed different FDIA strategies, system sizes, and performance metrics. Hence, the reported detection performances are hard to compare as these studies lack common ground. Second, the aforementioned detectors employ classical ML-based techniques that, by nature, do not capture the spatial relationships within the measurement data implied by the power system topology [6], [8]. Hence, we refer to these detectors as topology-unaware. Our results will show that these topology-unaware detectors offer a limited detection performance when compared under the same conditions with topology-aware detectors. Specifically, the DR gap is $9.3-30\%$ as will be shown in Section IV-D1.

*2) Graph-based Detection:* To exploit the grid topological information, detectors based on GSP and GNN approaches were proposed in the literature.

*a) GSP Models:* GSP-based detectors adopt manually designed spectral filters [18], [19], [20]. For example, [19] and [20] offered DRs of around $90\%$. Despite capturing the topological information, the custom filter design of these GSP-based detectors limits the model scalability [6]. Also, they are only designed and tested on a specific system (e.g., IEEE 14-bus system) with small datasets, which limits their generalization ability to bigger systems.

*b) GNN Models:* GNN-based approaches were adopted in power grids for optimal power applications [21] and large-scale systems [22] as well as for FDIAs detection [6] and localization [23]. Such detectors adopt GNN models that automatically incorporate the GSP operations. For instance, the convolutional GNN (CGNN)-based detector in [6] offered DRs of $83-96\%$. This approach utilizes undirected graphs to capture the power system topology and hence presents a topology-aware detector. Despite the offered advantages, the GNN-based detectors in [6], [21] - [23] still present two major limitations. First, they lack the generalization ability as they are trained and tested on a fixed topological configuration for a given system size; hence, we refer to them as topology-specific. In practice, the reconfiguration of the power system topology takes place regularly for various reasons [24]. Thus, a flexible and robust generalized detector that offers a stable performance even in cases where topological reconfigurations take place is required. Since the detectors in [21] - [23] are built only based on one topological configuration, their detection performance is limited to only one configuration. Specifically, our results presented in Section IV-D2 will show that a CGNN model that lacks generalization abilities offers lower DRs (by $7-8\%$) when encountering new topological reconfigurations. Second, in terms of the effectiveness against unseen (zero-day) attacks, the detectors in [21] - [23] are only trained and tested against the same (seen) attack strategy. In practice, the detector may encounter a new attack strategy (zero-day attack) different from the ones that it has been trained against [25], [26]. Our results presented in Section IV-D3 will show that the existing strategies offer a lower DR (by $8.3-13.5\%$) when they encounter unseen FDIAs.

*3) Other Defense Strategies:* Besides the aforementioned ML-based detectors, other detection strategies have been proposed in the literature. Such strategies include an estimation-based framework to identify power line outages [27] and an iterative framework to prevent cascading contingencies [28]. However, such detectors are not designed to thwart stealthy FDIAs. Other FDIA detection schemes that are based on interval observer [29], [30], random matrix theory [31], structural separation [5], and hierarchical knowledge sharing [32] strategies were also proposed. Nevertheless, stealthy FDIAs can still bypass such detection schemes [33] since they present shallow detection techniques that do not capture the complex patterns within the measurement data [8]. Additionally, these non-ML detection schemes do not present independent systems to be trained and thus introduce applicability restrictions with attack identification threshold, detection delays, and scalability aspects [5], [6]. Furthermore, such schemes are topology-unaware and topology-specific and hence they offer limited detection performance in case of topological reconfigurations. Other studies focus only on proposing attack strategies targeting specific detectors in smart grids [34], [35], without offering practical attack detection strategies or mitigation solutions.

Based on the aforementioned limitations of existing detectors, further efforts are required to develop an effective graph ML-based detection strategy that is generalized and topology-aware, which can be independently trained on datasets to detect unseen attacks in unseen topological configurations with high DR and ACC as well as low false alarm rate (FAR).

### *B. Contributions*

In this work, we overcome the limitations of existing detectors by proposing a generalized GNN-based detection strategy employing a graph autoencoder (GAE) that offers the five advantages summarized in Table I. We validate the practicality of our detector by performing a comprehensive comparative analysis against multiple topology-aware and topology-unaware (classical ML models) detection strategies in generalized and topology-specific settings. The novel contributions of this work are listed next:

- First, the proposed GAE-based detector is topology-aware as it captures the spatial topological relationships within measurement data via Chebyshev graph convolution layers with attention mechanism. Existing detectors [9] - [17] are topology-unaware as they employ classical ML methods that fail to capture spatial features.
- Second, it offers generalized detection as it is trained on multiple topological configurations from 14, 39, and 118-bus systems. Hence, it detects FDIAs on unseen configurations (i.e., not part of its training set). Existing detectors [6], [9] - [23] are only tested against seen configurations (i.e., part of their training set). This makes the proposed detector more practical and robust against FDIAs even in the presence of seasonal topological reconfigurations.
- Third, it offers unsupervised anomaly detection that requires only benign data for training. It detects unseen malicious samples based on their deviation from the learned benign patterns. Existing detectors [6], [9] - [13], [15] - [23] are supervised and require benign and malicious samples for training, which limits their detection to seen attacks that are part of their training sets.

TABLE I
ADVANTAGES OF IMPLEMENTING THE PROPOSED GAE-BASED DETECTOR COMPARED TO EXISTING DETECTORS.

| Aspect | Existing Detectors | Proposed Detector | Significance |
|---|---|---|---|
| **Topology-awarness** | **Topology-unaware:**<br>Adopt classical ML models. | **Topology-aware:**<br>Employs a GNN to capture spatial features. | Allows the model to capture essential spatial features (i.e., distribution of the nodes and their connectivity). |
| **Detection setting** | **Topology-specific:**<br>Trained and tested only on one topological configuration. | **Generalized:**<br>Trained and tested on multiple topological configurations. | Allows for FDIAs detection even when occasional reconfigurations occur. |
| **Training nature** | **Supervised:**<br>Requires benign and malicious samples.<br>Detects FDIAs present in the training set. | **Unsupervised:**<br>Requires benign samples only.<br>Offers detection of unseen FDIAs. | Allows for FDIAs detection even when malicious entities carry out unseen attacks not present in the training set. |
| **Detection performance** | **Limited:**<br>Lower DR and ACC. Higher FAR. | **Superior:**<br>Higher DR and ACC. Lower FAR. | Better detection performance reflects better robustness against FDIAs. |
| **Scalability** | **Non-linear:**<br>Takes significantly more time to make decisions as system size increases. | **Linear:**<br>Takes slightly more time to make decisions as system size increases. | Linear scalability is required to maintain the real-time taken for online decision making as system size increases. |

- Fourth, it offers superior DRs of $87-91\%$ and $94-99\%$ in the topology-specific and generalized settings, respectively, offering DR improvements of $11.5-30\%$ compared to topology-unaware benchmark detectors.
- Fifth, it is highly scalable to larger systems with DR improvements of $2-6\%$ when tested in bigger systems compared to a small system. Unlike benchmarks, it offers linear scalability in terms of the time taken to make a decision as the system size (complexity) increases.

The remaining sections of this paper are organized as follows. Section II presents the dataset generation process using stochastic geometry along with the adopted FDIA functions. Section III introduces the design of the proposed GAE-based FDIAs detector. In Section IV, we first introduce the benchmark topology-aware (CGNN-based model) and topology-unaware (classical ML models) detectors along with their hyperparameters. We then report the detection performance of the investigated detectors along with the model analysis. Our conclusions and future works are presented in Section V.

## II. DATA GENERATION

This section describes (a) how the topologies of power grids are modeled to adopt GNNs, (b) the data generation process of different bus system topologies, (c) the FDIA data created via cyberattack functions, and (d) the generalized and topology-specific settings of the investigated detectors.

### A. Graph Modeling of Smart Grid

Since the power grid can be modeled as a graph, GNN techniques can be adopted to design an optimal FDIA detection strategy. Power grids have been modeled as undirected connected weighted graphs [6], [21] - [23]. As shown in Fig. 1, we opt to utilize undirected graphs rather than directed graphs since modeling power grids using directed graphs, where the adjacency matrices are asymmetric, restricts information exchange within the network (GNN model) [36]. Specifically, information exchange with buses located at the peripheral regions of the power grid becomes restricted, which limits the learning process of GNN models, as the graph diffusion is hindered [36]. To model the power grid, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ denote an undirected weighted graph consisting of a finite set of vertices $\mathcal{V}$ with $|\mathcal{V}| = n$ denoting the number of buses. $\mathcal{E}$ depicts the set of edges (power lines) and $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ denotes the weighted adjacency matrix (line admittance). If buses $i$ and $j$ are connected, a weight $W_{ij}$ is assigned to edge $e = (i, j)$.

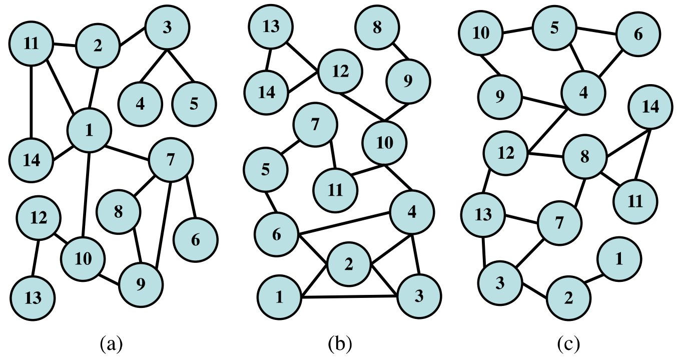


Fig. 1. Three different topological configurations for a 14-bus system.

Learning the respective patterns helps determine if the system state is either in normal operation (benign) or under cyberattack (malicious) [6]. Since the power system can be represented by a graph tree structure, GNN presents itself as a powerful learning candidate that identifies the system state by fusing topological features (i.e., spatial distribution of the nodes and their connectivity) and temporal features from the power grid (i.e., power injections and flows). To develop a generalized detection strategy that adapts well to any reconfiguration in the power system topology, the GNN-based model should be trained on large spatio-temporal datasets representing various topological configurations.

Unfortunately, the required spatial and temporal datasets that are needed to develop the attack and defense strategies are not available. Existing graph-based models are either (a) representing unrealistic random graphs, (b) relying on specific IEEE bus system topology [37], or (c) employing information restricted to specific cities (e.g., [38], [39]). Existing random graph models do not capture practical topological (spatial) and temporal features and hence cannot be relied upon [40]. While spatial features of IEEE bus test systems can be obtained via different tools such as OpenDSS, the available bus test systems do not offer various topological configurations for each system size, which is needed for the generalization ability. Moreover, the studies that are based on actual cities do not share the spatial and temporal features of their systems due to non-disclosure agreements and national security reasons [40].

### *B. Generative Spatio-temporal Model*

Given the aforementioned limitations, there is a need for a generative model that can establish many topological configurations of practical power systems with various scale sizes. To develop a generative model that can be used to generate different configurations of power grids, we resort to stochastic geometry [40]. Next, we describe why and how stochastic geometry is used to generate spatial features followed by the temporal features for the different topological configurations.

*1) Generating Spatial Features Using Stochastic Geometry:* Stochastic geometry is a powerful tool that takes into consideration the physical constraints of connecting the power elements together [41]. Also, it captures the spatial coupling and correlations of the electrical elements [42]. Such advantages follow from the fact that the stochastic geometric morphogenesis of cities utilizing iterated Poisson tessellations is in match to a high degree with the real world [43]. The stochastic geometry model has been validated in [40] against real power grids and IEEE test systems. Hence, stochastic geometry is used herein to develop various topological configurations that mimic real-world power grids. This allows us to generate numerous samples of the spatial features (node distributions and connectivities) that are used to develop the generalized detector. Specifically, stochastic geometry is used as a tool to generate Poisson lines representing roads in a region, and distribute buses/power substations along these lines. This allows the power grid model to be more realistic and practical since electrical elements do not present a fixed pattern (e.g., a simple square-shaped grid) due to geographical, physical, energy, and cost constraints [44]. To construct practical power system topological configurations using stochastic geometry based on [40], we carry out five steps: step 1: define a geographical area representing a region; step 2: generate Poisson lines representing roads; step 3: distribute bus nodes on each line; step 4: connect buses together; step 5: assign electrical parameters. Next, we elaborate on each step.

**Step 1:** A geographical region (e.g., city) is defined and modeled as a disk with radius $R$.

**Step 2:** To represent roads, $|M|$ lines are generated inside the disk from a set of points distributed according to a Poisson distribution on a representation space following a Poisson line process (PLP). Each line $m$ is characterized by $0 \leq \mu_m < 2\pi$ as the line direction, and $0 < \upsilon_m \leq R$ as the line location.

**Step 3:** On each Poisson line $m$, $|\mathcal{V}|_m$ buses/power substations are created and distributed following a one-dimensional homogeneous Poisson point process (HPPP) with density $\lambda_{\mathcal{V}_m}$. The buses in the system construct an HPPP with a density of

$$\lambda_{\mathcal{V}} = \sum_{m=1}^{|M|} \lambda_{\mathcal{V}_m}. \tag{1}$$

**Step 4:** Each bus and its neighbors are connected together according to the physical paths using a potential near-geodesic route and the shifted sum of exponential distributions of the buses' degree [41], which provides an accurate approximation to common power distribution system structures. To reduce the power losses and ensure power delivery, disconnected buses are linked according to the shortest pathways among them.

**Step 5:** Electrical parameters are assigned by statistically matching the generated values of the stochastic topology with an actual power system or test system of the same size [37].

Since this is a stochastic process, repeating the same steps while maintaining the same density allows us to create several topological configurations of a given system size, whose spatial features (nodal degree, degree centrality, eigenvalue spread, etc.) match to a high degree a practical power system of the same size. For example, Fig. 1 shows three topological configurations for a system of 14 buses whose spatial features will match to a high degree an IEEE 14-bus test system. The advantage here is that while the IEEE 14-bus system presents a single topological configuration, the stochastic geometry process described herein enables us to create a large number of topological configurations, which can be used while training the GNN-based detection model to enhance the detector's generalization ability against any reconfiguration.

*2) Temporal Features:* For each power grid topology, the temporal features (power injections and flows) are generated by simulating the power flow within the power grid topology. To do that, power flow analysis using Newton's method is carried out for each topology using MATLAB MATPOWER toolbox [45]. The toolbox allows us to determine the voltages and currents along with real and reactive power flows in the system. We generate temporal features for each developed configuration. The nodes (vertices) $\mathcal{V}$ and edges $\mathcal{E}$ of a graph $\mathcal{G}$ are labeled with features. The generated node data contains real power demand (active power $P_i$ in MW) and reactive power demand ($Q_i$ in MVAr). The edge data contains real power injected at the "from" bus $i$ end (MW) and real power injected at the "to" bus end (MW) denoted as active power $P_{ij}$ flows between bus $i$ and $j$. The edge data also comprises reactive power injected at the "from" bus $i$ end (MVAr) and reactive power injected at the "to" bus $j$ end (MVAr) depicted as reactive power $Q_{ij}$ flows between bus $i$ and $j$.

*3) Topological Configurations:* Using the aforementioned methods, large spatio-temporal datasets can be generated for power systems under normal operation and attack conditions, which can be used to develop a generalized detection strategy. We generate ten different topological configurations for three power system sizes (ten configurations for a 14-bus system, ten configurations for a 39-bus system, and ten configurations for a 118-bus system). For each topology, we capture 96 power dynamics timestamps in a day (measurements every 15 minutes) over half a year, giving a total of $17,520$ timestamps. By simulating the power flow across the system, we generate benign data reflecting normal operation. Then, we inject false data following the attack strategy described next (in Section II-C) to generate malicious data. The obtained data is used to train and test the proposed GAE model to generalize defense mechanisms against unseen attacks and unseen topological configuration. Henceforth, notations $\boldsymbol{x}_{\text{BEN}}(t,i)$ and $\boldsymbol{x}_{\text{MAL}}(t,i)$ denote the benign (true) and malicious (false) measurement

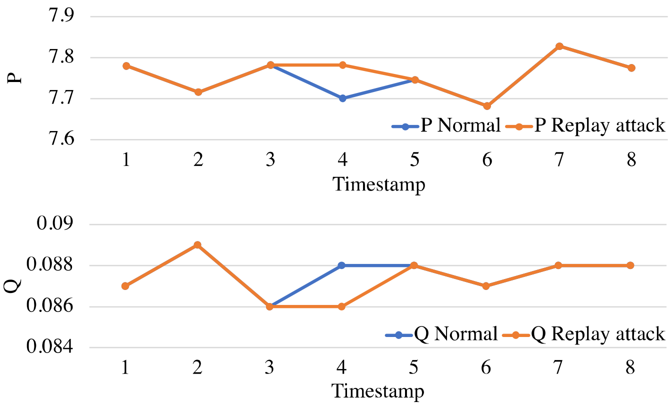


Fig. 2. Sample P and Q of normal operation and replay attack.

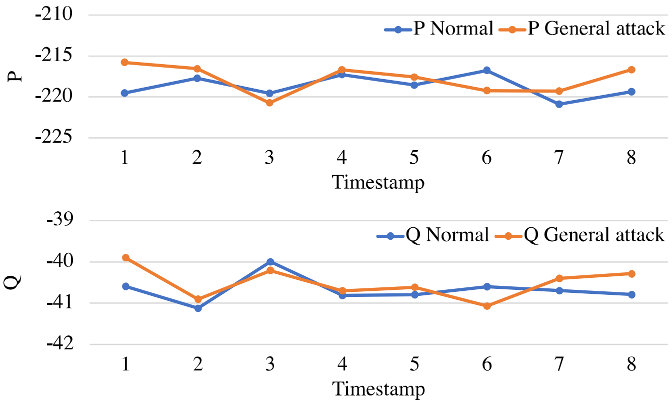


Fig. 3. Sample P and Q of normal operation and general attack.

value during normal and abnormal operation, respectively, at timestamp $t$ and bus $i$.

### C. Attack Data

We consider two FDIA types, namely, data replay attack and general attack [46]. The difference between the measurements and the corresponding manipulated data is kept below a certain threshold value that can be deemed acceptable by the power system to bypass the traditional bad data detector. For the replay attack, due to the high data correlation between two consecutive timestamps, the residual is kept at a smaller value, making the attack stealthy. Similarly, for the general attack, the parameters are selected such that the residual between the real and the manipulated data is kept small enough, yet effective.

In the data replay attack, an attack sample (i.e., bus $i$ and timestamp $t$) is selected randomly where the true measurement value of a previous timestamp $(t-1)$ is repeated such that

$$\boldsymbol{x}_{\text{MAL}}(t,i) = \boldsymbol{x}_{\text{BEN}}(t-1,i), \tag{2}$$

where $\boldsymbol{x}_{\text{MAL}}(t,i)$ denotes a malicious sample at the current timestamp $t$ at bus $i$. Fig. 2 illustrates sample $P$ and $Q$ data from a 14-bus system at a given bus $i$ comparing normal operation with a replay attack over a two-hour-period, where a replay attack takes place at timestamp $4$.

In the general attack, false data is being injected such that

$$\boldsymbol{x}_{\text{MAL}}(t,i) = \boldsymbol{x}_{\text{BEN}}(t,i) + (-1)^{\beta}\alpha.\gamma.\text{Range}(\boldsymbol{x}_{\text{BEN}}(t,i)), \tag{3}$$

where $\beta$ is a binary random variable, $\alpha$ is the attack magnitude, and $\gamma$ is a uniform random variable between $(0,1)$. The last term (Range) denotes the range of the true measurements at timestamp $t$ and bus $i$. Since it is a measurement taken by metering equipment, there could be associated uncertainties described by a margin of error. To account for these uncertainties, we define a range of $\pm 1\%$ deviation from the true measurement [47]. Fig. 3 illustrates sample $P$ and $Q$ data from a 14-bus system at bus $i$ comparing normal operation with a general attack over a two-hour period.

### D. Detection Setting

We compare the performance of the investigated detectors in two settings, namely, generalized and topology-specific settings depending on the training nature of the models. In all of the conducted experiments, the number of benign and malicious samples is balanced with a 1:1 ratio. Since supervised detectors use benign and malicious samples in the training and testing phases, we use an equal number of benign and malicious samples in both phases. Unsupervised detectors use only benign samples for training, but are tested on an equal number of benign and malicious samples. The number of training and testing topologies for each detection setting is described next.

*1) Generalized Training:* In the generalized setting, the investigated models utilize the ten topologies of each system size assembled as $\boldsymbol{\Gamma} = [1, 2, ..., 10]$ following a leave-one-out method to identify the training and testing samples [48]. Specifically, for each bus system size (14, 39, and 118-bus systems), we conduct eight different generalized experiments. Each experiment has a training set $\boldsymbol{X}_{\text{TR}}$ that consists of seven labeled topologies, a validation set $\boldsymbol{X}_{\text{VAL}}$ containing one labeled topology, and a test set $\boldsymbol{X}_{\text{TST}}$ with the remaining two unseen labeled topologies. We then report the average detection performance over these experiments. Labeling captures the temporal node and edge features as well as the classification of the system (normal operation or under attack). For example, in the first experiment, the generalized models are trained on samples from seven topologies $\boldsymbol{\Gamma} = \{1, 2, ..., 7\}$, validated on one topology $\boldsymbol{\Gamma} = \{8\}$, and tested on two unseen topologies $\boldsymbol{\Gamma} = \{9, 10\}$.

*2) Topology-Specific Training:* In addition to the generalized experiments, for comparison reasons, we also examine the detectors in a topology-specific setting. Specifically, we carry out eight topology-specific experiments. In each experiment, the investigated models are trained only on one topology (e.g., $\boldsymbol{\Gamma} = \{1\}$), validated on another one (e.g., $\boldsymbol{\Gamma} = \{2\}$), and tested on two unseen topologies (e.g., $\boldsymbol{\Gamma} = \{3, 4\}$).

## III. Generalized GAE-based Detector

This section presents our topology-aware GNN generalized detector. Specifically, we propose a GAE-based anomaly detector that is generalized, which means that it is trained on the graph representations of certain topologies and is able to detect unseen FDIA types in different unseen topologies.

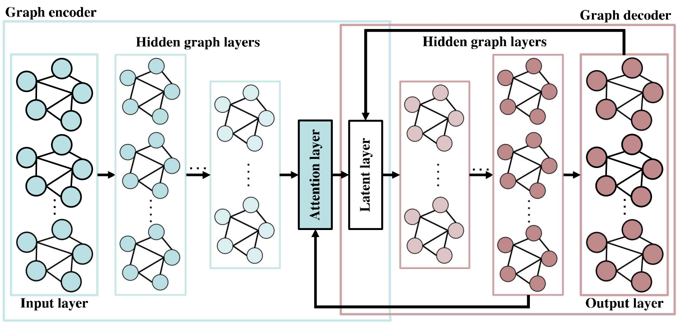


Fig. 4. Illustration of the proposed unsupervised GAE-based detection.

### A. *GAE Model Architecture*

The proposed GAE-based detector requires only benign samples $\boldsymbol{x}_{\mathrm{BEN}}$ from the normal operation measurements for training [49] and it is tested on (used against) unseen types of attacks in unseen topologies. The GAE-based detector learns the graph representation of the normal operation data throughout a data reconstruction process using an encoder and decoder [50], as shown in Fig. 4. Next, we describe the Chebyshev convolution operation as well as the components of the proposed GAE model.

*1) Chebyshev Convolution Operation:* To capture spatial aspects of the power system topology, it is essential to apply a convolution operation to graph signals. Therefore, we propose a GAE-based detector that performs a Chebyshev convolution operation. Unlike classical signal processing, in the vertex domain within graphs, an expressive translation operator does not exist [6]. Thus, performing a convolution operation to graph signals requires transforming them into the spectral domain using graph Fourier transform, then convolving them (Hadamard product) in the spectral domain, then transforming the result back into the vertex domain using inverse graph Fourier transform [51]. Typically, a graph signal is filtered by a kernel function in the spectral domain by multiplying its spectral components with the Fourier coefficients. However, these filters are parameter-free, not spatially localized, and present high computational complexity. To overcome this, the filters are parameterized as a Chebyshev polynomial function [52] computed recursively from the graph Laplacian operator, which is a fundamental operator in spectral graph theory [6].

*2) GAE Components:* The proposed GAE model takes, as input, the benign samples $\boldsymbol{x}_{\mathrm{BEN}}$ from the different topologies $\boldsymbol{\Gamma}$ with $[P_i, Q_i] \in \mathbb{R}^{n\times 2}$ measurements. Specifically, the GAE model consists of the following components.

*a) Graph Encoder $\boldsymbol{E}$:* The input layer is followed by a graph encoder $\boldsymbol{E}$ with $\mathcal{L}_{\mathrm{E}}$ hidden encoding Chebyshev graph convolution layers. Let $c_{l_{\mathrm{E}}}$ denote the number of channels (i.e., inputs to the graph convolution layers) in a hidden encoding layer $l_{\mathrm{E}}$. Each $l_{\mathrm{E}}$ has an input of $\boldsymbol{X}^{l_{\mathrm{E}}-1} \in \mathbb{R}^{n\times c_{l_{\mathrm{E}}-1}}$ and output of $\boldsymbol{X}^{l_{\mathrm{E}}} \in \mathbb{R}^{n\times c_{l_{\mathrm{E}}}}$. Such layers capture the spatial features within the graphs by applying the graph convolution operation, adding bias, and employing a ReLU function to produce the output tensor expressed as

$$\boldsymbol{X}^{l_{\mathrm{E}}} = \mathrm{ReLU}(\boldsymbol{\theta}^{l_{\mathrm{E}}} *_{\mathcal{G}} \boldsymbol{X}^{l_{\mathrm{E}}-1} + \boldsymbol{b}^{l_{\mathrm{E}}}), \tag{4}$$

where $\boldsymbol{\theta}^{l_{\mathrm{E}}} \in \mathbb{R}^{K\times c_{l_{\mathrm{E}}-1}\times c_l}$ and $\boldsymbol{b}^{l_{\mathrm{E}}} \in \mathbb{R}^{c_{l_{\mathrm{E}}}}$ denote the Chebyshev coefficients with $K$ order and bias of $l_{\mathrm{E}}$, respectively, and $*_{\mathcal{G}}$ is a graph convolution operator. The presence of bias along with the ReLU activation function enhances the nonlinear modeling capability of the detector [53].

*b) Attention Layer $L_{\mathrm{A}}$:* Following $\mathcal{L}_{\mathrm{E}}$, we place an attention layer $L_{\mathrm{A}}$ that allocates weights and scores for each instance where higher importance is assigned to instances that offer higher contribution toward obtaining the desired output [54]. As illustrated in Fig. 4, $L_{\mathrm{A}}$ receives $\boldsymbol{X}^{l_{\mathrm{E}}}$ and $\boldsymbol{X}^{l_{\mathrm{D}}-1}$, the outputs of the last encoding and first decoding graph layers, respectively. Attention is achieved via an alignment score

$$\boldsymbol{\kappa} = \xi(\boldsymbol{X}^{l_{\mathrm{E}}}, \boldsymbol{X}^{l_{\mathrm{D}}-1}) \tag{5}$$

with alignment function $\xi$, where the weight of attention is a Softmax of alignment scores as follows

$$\boldsymbol{\Omega} = \frac{\exp(\boldsymbol{\kappa})}{\sum_{|\boldsymbol{\kappa}|}\exp(\boldsymbol{\kappa})}. \tag{6}$$

The multiplication of $\boldsymbol{\Omega}$ and the weighted sum of the instances produces a context vector $X^{L_{\mathrm{A}}}$ denoting the output of $L_{\mathrm{A}}$.

*c) Latent Layer $L_{\mathrm{S}}$:* $X^{L_{\mathrm{A}}}$ along with the reconstructed output $\tilde{\boldsymbol{X}}$ from the graph decoder's side are passed to the latent layer $L_{\mathrm{S}}$, where the concatenation $\sum(X^{L_{\mathrm{A}}}, \tilde{\boldsymbol{X}})$ takes place and passed to the graph decoder $\boldsymbol{D}$.

*d) Graph Decoder $\boldsymbol{D}$:* The latent layer is followed by a graph decoder $\boldsymbol{D}$ with $\mathcal{L}_{\mathrm{D}}$ hidden decoding Chebyshev graph convolution layers with $c_{l_{\mathrm{D}}}$ channels. The aforementioned resultant concatenation presents the input to the proceeding graph decoding layer $l_{\mathrm{D}}$ with the output of $\boldsymbol{X}^{l_{\mathrm{D}}} \in \mathbb{R}^{n\times c_{l_{\mathrm{D}}}}$. The following graph decoding layers take $\boldsymbol{X}^{l_{\mathrm{D}}-1} \in \mathbb{R}^{n\times c_{l_{\mathrm{D}}-1}}$ as input and output $\boldsymbol{X}^{l_{\mathrm{D}}} \in \mathbb{R}^{n\times c_{l_{\mathrm{D}}}}$. The graph decoder is responsible for reconstructing the original input of the network where the reconstructed output $\tilde{\boldsymbol{X}}$ is produced.

### B. *GAE Model Training*

The training algorithm of the GAE model is summarized in Algorithm 1. The GAE-based detector identifies abnormal operation by assessing the deviation from the learned normal patterns. High deviation indicates the presence of a cyberattack. Defining when an anomaly occurs depends on the reconstruction error $\zeta$ during the reconstruction process. Specifically, we define the graph encoder and decoder as $\boldsymbol{E} = f_{\Phi}(\boldsymbol{X})$ and $\boldsymbol{D} = g_{\Phi}(\boldsymbol{X})$, respectively, where $\Phi$ denotes the GAE parameters that are determined such that

$$\min_{\Phi} C(\boldsymbol{X}, g_{\Phi}(f_{\Phi}(\boldsymbol{X}))), \qquad \boldsymbol{X} \in \boldsymbol{X}_{\mathrm{TR}}, \tag{7}$$

where $C(\boldsymbol{X}, g_{\Phi}(f_{\Phi}(\boldsymbol{X})))$ denotes a cost function (i.e., the mean squared error (MSE)) that penalizes $g_{\Phi}(f_{\Phi}(\boldsymbol{X}))$ for being dissimilar from $\boldsymbol{X}$. In Algorithm 1, $\boldsymbol{\theta}^{l_{\mathrm{E}}}$ and $\boldsymbol{\theta}^{l_{\mathrm{D}}}$ denote the free Chebyshev coefficients in the graph encoder and decoder, respectively. $\boldsymbol{b}^{l_{\mathrm{E}}}$ and $\boldsymbol{b}^{l_{\mathrm{D}}}$ denote the bias in the graph encoder and decoder, respectively. The training of the GAE model aims to find the model parameters $\boldsymbol{\theta}^{l_{(\cdot)}}$ and $\boldsymbol{b}^{l_{(\cdot)}}$, denoted by $\Phi$, that optimize (7). The minimization of (7) is attained through the iterative gradient descent algorithm depicted in Algorithm 1 that assumes a stochastic gradient descent execution with learning rate $\eta$. $\nabla$ denotes the partial derivative and $|\boldsymbol{X}_{\text{TR}}|$ denotes the number of training samples. The training samples $\boldsymbol{X}$ from $\boldsymbol{X}_{\text{TR}}$ are split into equally-sized mini batches and fed into the model with 128 epochs.

**Algorithm 1:** Proposed GAE Model Training

```
Input Data: $\boldsymbol{X}_{\text{TR}}$
Initialization: Chebyshev coefficients $\boldsymbol{\theta}^{l_{(\cdot)}}$ and bias
  $\boldsymbol{b}^{l_{(\cdot)}}\ \forall\ l_{(\cdot)}$, $\boldsymbol{X}^{l_{\text{D}}-1}$, and $\tilde{\boldsymbol{X}}$
while not converged do
    for each topology $\boldsymbol{\Gamma}$ do
        for each training sample $\boldsymbol{X}$ do
            Feed forward:
            Graph Encoder ($\boldsymbol{E}$):
            for hidden layers $l_{\text{E}} \in \mathcal{L}_{\text{E}}$ do
                $\boldsymbol{X}^{l_{\text{E}}} = \text{ReLU}(\boldsymbol{\theta}^{l_{\text{E}}} *_{\mathcal{G}} \boldsymbol{X}^{l_{\text{E}}-1} + \boldsymbol{b}^{l_{\text{E}}})$
            end
            Attention Layer ($L_{\text{A}}$):
            $\boldsymbol{\kappa} = \xi(\boldsymbol{X}^{l_{\text{E}}}, \boldsymbol{X}^{l_{\text{D}}-1})$
            $\boldsymbol{\Omega} = \frac{\exp(\boldsymbol{\kappa})}{\sum_{|\boldsymbol{\kappa}|} \exp(\boldsymbol{\kappa})}$
            The multiplication of $\boldsymbol{\Omega}$ and
            instances' weighted sum produces $X^{L_{\text{A}}}$
            $L_{\text{A}}$ outputs $X^{L_{\text{A}}}$
            Latent Layer ($L_{\text{S}}$):
            $\sum(X^{L_{\text{A}}}, \tilde{\boldsymbol{X}})$
            Graph Decoder ($\boldsymbol{D}$):
            for hidden layers $l_{\text{D}} \in \mathcal{L}_{\text{D}}$ do
                $\boldsymbol{X}^{l_{\text{D}}} = \text{ReLU}(\boldsymbol{\theta}^{l_{\text{D}}} *_{\mathcal{G}} \boldsymbol{X}^{l_{\text{D}}-1} + \boldsymbol{b}^{l_{\text{D}}})$
            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:
            $\nabla_{\boldsymbol{\theta}^{l_{(\cdot)}}} C$ and $\nabla_{\boldsymbol{b}^{l_{(\cdot)}}} C$
        end
        Parameters update:
          $\boldsymbol{\theta}^{l_{(\cdot)}} = \boldsymbol{\theta}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{\theta}^{l_{(\cdot)}}} C$
          $\boldsymbol{b}^{l_{(\cdot)}} = \boldsymbol{b}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{b}^{l_{(\cdot)}}} C$
    end
end
Output: Optimal parameters $\boldsymbol{\theta}^{l_{(\cdot)}}$ and bias $\boldsymbol{b}^{l_{(\cdot)}}\ \forall\ l_{(\cdot)}$
```

Following the cost function (7), the reconstruction error $\zeta$ is expected to be small for benign data and large for anomalies. $\zeta$ is used to indicate how familiar the model is with a given test instance $\boldsymbol{X} \in \boldsymbol{X}_{\text{TST}}$; whenever it exceeds a certain threshold $\psi$, an anomaly indicating an FDIA is detected. If $\zeta > \psi$, $\boldsymbol{X}$ is considered as $\boldsymbol{X}_{\text{MAL}}$. After training using $\boldsymbol{X}_{\text{TR}}$, we apply the test set $\boldsymbol{X}_{\text{TST}}$. If the cost function, which computes the MSE between $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ is greater than $\psi$, labels of $y = 1$ and $y = 0$ are assigned to a malicious and benign sample, respectively.

## IV. Experimental Results

This section introduces the investigated benchmark topology-aware and topology-unaware detectors. We then present the hyperparameter search algorithm along with the selected optimal hyperparameter accordingly. Then, we introduce the performance metrics and provide the numerical analysis of the investigated detectors. We conclude the section by examining the models in terms of the training time, scalability, and number of topologies required for a generalization ability.

### A. Benchmark detectors

We compare the performance of our proposed detector against two main types of benchmark detectors, namely, a topology-aware CGNN-based detector (based on [6]) and classical ML-based detectors (based on [12] - [17]) that are topology-unaware. Next, we introduce the CGNN-based detector in detail since it presents a GNN model that is topology-aware, which is the focus of this paper. Then, we briefly list the topology-unaware classical ML-based detectors.

*1) CGNN-based Detection:* The CGNN-based detector [6] is trained on data from the normal and malicious operations, and then tested on seen types of FDIAs.

*a) CGNN Model Architecture:* As shown in Fig. 5, the CGNN-based detector consists of an input layer representing the different topologies $\boldsymbol{\Gamma}$ with $[P_i, Q_i] \in \mathbb{R}^{n\times 2}$ measurements that could be either $\boldsymbol{X}_{\text{BEN}}$ or $\boldsymbol{X}_{\text{MAL}}$. The input layer is followed by $\mathcal{L}_{\text{V}}$ hidden Chebyshev graph convolution layers. Let $c_{l_{\text{V}}}$ denote the number of channels in a hidden layer $l_{\text{V}}$ that has as input $\boldsymbol{X}^{l_{\text{V}}-1} \in \mathbb{R}^{n\times c_{l_{\text{V}}-1}}$ and output of $\boldsymbol{X}^{l_{\text{V}}} \in \mathbb{R}^{n\times c_{l_{\text{V}}}}$. Each $l_{\text{V}}$ helps capture the spatial features within the graphs by performing the graph convolution operation, adding bias, and employing a ReLU function [6]. The ReLU function is responsible for producing the output tensor $\boldsymbol{X}^{l_{\text{V}}}$ of $l_{\text{V}}$ such that

$$\boldsymbol{X}^{l_{\text{V}}} = \text{ReLU}(\boldsymbol{\theta}^{l_{\text{V}}} *_{\mathcal{G}} \boldsymbol{X}^{l_{\text{V}}-1} + \boldsymbol{b}^{l_{\text{V}}}), \tag{8}$$

where $\boldsymbol{\theta}^{l_{\text{V}}} \in \mathbb{R}^{K\times c_{l_{\text{V}}-1}\times c_l}$ and $\boldsymbol{b}^{l_{\text{V}}} \in \mathbb{R}^{c_{l_{\text{V}}}}$ denote the Chebyshev coefficients and bias of $l_{\text{V}}$, respectively. The proceeding dense layer determines the probability of having an attacked sample, where the decision is presented in the output layer. The dense layer takes the output $\boldsymbol{X}^{L_{\text{V}}} \in \mathbb{R}^{n\times c_{L_{\text{V}}}}$ of the last hidden layer $L_{\text{V}}$ and outputs the result of the following sigmoid function

$$\text{sigmoid}(\boldsymbol{W}^{L_{\text{V}}} \boldsymbol{X}^{L_{\text{V}}} + \boldsymbol{b}^{L_{\text{V}}}), \tag{9}$$

where $\boldsymbol{W}^{L_{\text{V}}} \in \mathbb{R}^{n\times c_{L_{\text{V}}}}$ and $\boldsymbol{b}^{L_{\text{V}}} \in \mathbb{R}$ denote the feature weights and bias, respectively. The presence of bias along with ReLU and sigmod activation functions enhances the nonlinear modeling capability of the detector [53].

*b) CGNN Model Training:* To train the CGNN model and compute the free parameters, a cross-entropy loss function is adopted, i.e.,

$$C(\tilde{y}, \sigma) = \frac{-1}{|\boldsymbol{X}_{\text{TR}}|} \sum_{\boldsymbol{X}_{\text{TR}}} \{y \log(\tilde{y}) + (1-y)\log(1-\tilde{y})\}, \tag{10}$$

where $|\boldsymbol{X}_{\text{TR}}|$ denotes the number of training samples and $\sigma$ depicts the trainable parameters ($\boldsymbol{\theta}^{l_{\text{V}}}$, $\boldsymbol{b}^{l_{\text{V}}}$, $\boldsymbol{W}^{L_{\text{V}}}$, and $\boldsymbol{b}^{L_{\text{V}}}$). $y$ and $\tilde{y}$ represent the true and predicted label (i.e., benign or malicious) of a given sample, respectively. The model is trained using an iterative gradient descent-based optimization algorithm, where the training samples $\boldsymbol{X}$ from $\boldsymbol{X}_{\text{TR}}$ are split into equally-sized mini batches and fed into the model.

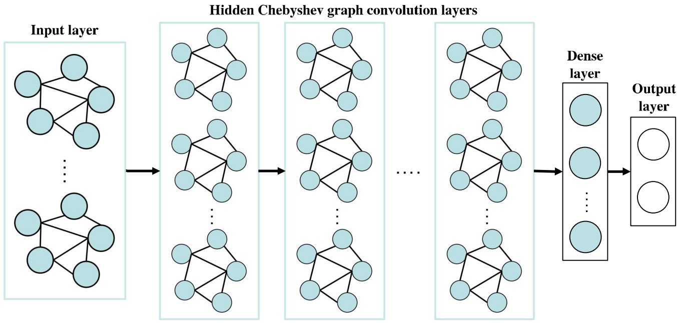


Fig. 5. Illustration of a CGNN-based detection.

*2) Classical ML-Based Benchmark Detectors:* The investigated benchmark classical ML-based detectors are topology-unaware (i.e., do not exploit information about the topology $\boldsymbol{\Gamma}$). The benchmark detectors listed below offer various characteristics including supervised/unsupervised training natures, shallow/deep structures, and static/dynamic mechanisms.

- Autoregressive integrated moving average (ARIMA) is an unsupervised shallow dynamic model that is trained on normal operation data to predict future patterns using minimum prediction MSE. When a sample exceeds a specific threshold $\psi$ during testing, it is marked as an abnormal operation [55].
- FNN is a supervised deep static model trained on normal and malicious operation data. It learns the patterns via stacked hidden layers where neurons are fully-connected and information flows in a feedforward manner [12].
- Long short-term memory (LSTM) is a supervised deep dynamic RNN-variation model. It is trained on normal and malicious data where information flows in recurrent cycles to hold previous knowledge [56].
- CNN is a supervised deep model that extracts features from the data by conducting convolutions [17].
- Autoencoder with attention (AEA) is an unsupervised deep dynamic model that learns behavioral patterns of normal operation data by reconstructing data using recurrent layers equipped with an attention mechanism [25].

### *B. Optimal Hyperparameters*

To get the best possible performance out of the investigated detectors (proposed and benchmarks), we perform a sequential grid search on the models' hyperparameters where each hyperparameter is selected separately in sequential stages [57]. The hyperparameter that offers the highest DR in the validation phase against $\boldsymbol{X}_{\text{VAL}}$ gets selected. The optimal hyperparameter value $\varrho$ is selected from a predefined search space $\mathcal{P}$ as follows. Number of layers $\mathcal{L} = \{2, 3, 4, 5, 6, 8\}$, number of units $\mathcal{U} = \{4, 8, 16, 32, 64\}$, dropout rate $\mathcal{D} = \{0, 0.2, 0.4, 0.5\}$, neighborhood order $\mathcal{K} = \{2, 3, 4, 5\}$, optimizer $\mathcal{O} = \{$SGD, Adam, Adamax, and Rmsprop$\}$, and activation function $\mathcal{A} = \{$Relu, Sigmoid, Elu, Tanh$\}$. Table II lists the optimal hyperparameters for the investigated deep and GNN models. For the autoencoder-based detectors (AEA and GAE), the reported $U$ is for the first hidden layer in the encoder. For example, a four-layer autoencoder with $U = 32$ would have $(32, 16)$ and $(16, 32)$ units in the two encoding and decoding layers, respectively. For ARIMA, the optimal values of the moving average and differencing degree turn out to be $0$ and $1$, respectively, which are selected from the search space of $\{0,\ 1,\ 2,\ 3\}$. The optimal detection threshold values $\psi$ for the unsupervised ARIMA, AEA, and proposed GAE model turn out to be $0.43$, $0.52$, and $0.54$, respectively.

TABLE II
OPTIMAL HYPERPARAMETERS OF THE INVESTIGATED DETECTORS.

| Detector | $\varrho$ | 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 |
| AEA | $L$ | 6 | 4 | 8 |
| | $U$ | 32 | 32 | 32 |
| | $D$ | 0 | 0 | 0.2 |
| | $O$ | Adam | Adam | SGD |
| | $A$ | Sigmoid | Sigmoid | Sigmoid |
| CGNN | $L$ | 5 | 4 | 5 |
| | $U$ | 16 | 16 | 32 |
| | $K$ | 3 | 5 | 4 |
| | $O$ | Adam | Rmsprop | Rmsprop |
| | $A$ | Relu | Relu | Relu |
| GAE | $L$ | 6 | 6 | 6 |
| | $U$ | 32 | 64 | 64 |
| | $K$ | 4 | 4 | 5 |
| | $O$ | Rmsprop | Adam | Adam |
| | $A$ | Relu | Relu | Relu |

### *C. Performance Metrics*

To compare the detection performance of the investigated detectors, we adopt detection rate (DR = TP/(TP + FN)) to determine the portion of correctly detected malicious samples, false alarm rate (FAR = FP/(FP + TN) to determine the portion of benign samples incorrectly marked as malicious, and accuracy (ACC = (TP + TN)/(TP + TN + FP + FN)) to determine how well the model marks both sample types. TP (correctly identified malicious samples) and TN (correctly identified benign samples) denote true positive and true negative samples, respectively. FP (incorrectly identified benign samples) and FN (incorrectly identified malicious samples) denote false positive and false negative samples, respectively.

### *D. Numerical Results*

Tables III and IV present the detection performance in generalized and topology-specific settings, respectively. They

TABLE III
DETECTION PERFORMANCE OF GENERALIZED DETECTORS (%)

| Bus System Size | Detector | Metric | | |
|---|---|---|---|---|
| | | DR | FAR | ACC |
| **14-bus** | **ARIMA** | 64.6 | 43.2 | 63.4 |
| | **FNN** | 71.6 | 28.5 | 70.1 |
| | **LSTM** | 77.0 | 22.4 | 75.2 |
| | **CNN** | 80.3 | 16.9 | 79.9 |
| | **AEA** | 82.1 | 15.3 | 81.1 |
| | **CGNN** | 91.4 | 6.3 | 90.4 |
| | **Proposed GAE** | **93.6** | **4.7** | **93.1** |
| **39-bus** | **ARIMA** | 66.1 | 41.6 | 65.2 |
| | **FNN** | 73.3 | 27.1 | 71.9 |
| | **LSTM** | 78.5 | 20.9 | 77.1 |
| | **CNN** | 82.2 | 15.3 | 80.9 |
| | **AEA** | 83.8 | 13.5 | 82.9 |
| | **CGNN** | 93.4 | 4.5 | 92.7 |
| | **Proposed GAE** | **95.7** | **2.7** | **95.0** |
| **118-bus** | **ARIMA** | 69.2 | 39.1 | 68.3 |
| | **FNN** | 77.1 | 23.9 | 75.3 |
| | **LSTM** | 82.5 | 17.8 | 80.6 |
| | **CNN** | 86.3 | 12.1 | 85.4 |
| | **AEA** | 87.2 | 10.4 | 86.4 |
| | **CGNN** | 97.6 | 1.2 | 97.3 |
| | **Proposed GAE** | **99.1** | **0.3** | **98.7** |

TABLE IV
DETECTION PERFORMANCE OF TOPOLOGY-SPECIFIC DETECTORS (%)

| Bus System Size | Detector | Metric | | |
|---|---|---|---|---|
| | | DR | FAR | ACC |
| **14-bus** | **ARIMA** | 53.0 | 54.3 | 52.3 |
| | **FNN** | 60.4 | 39.6 | 59.2 |
| | **LSTM** | 66.1 | 33.4 | 64.6 |
| | **CNN** | 70.7 | 26.7 | 70.3 |
| | **AEA** | 71.6 | 25.7 | 71.1 |
| | **CGNN** | 84.6 | 13.7 | 83.7 |
| | **Proposed GAE** | **86.9** | **12.0** | **86.4** |
| **39-bus** | **ARIMA** | 53.9 | 53.6 | 53.1 |
| | **FNN** | 61.0 | 39.3 | 60.0 |
| | **LSTM** | 66.6 | 32.8 | 65.4 |
| | **CNN** | 71.6 | 25.9 | 71.3 |
| | **AEA** | 72.4 | 24.7 | 72.0 |
| | **CGNN** | 86.3 | 11.7 | 85.5 |
| | **Proposed GAE** | **88.7** | **9.8** | **87.9** |
| **118-bus** | **ARIMA** | 56.2 | 52.7 | 54.6 |
| | **FNN** | 63.9 | 37.1 | 61.5 |
| | **LSTM** | 69.6 | 30.5 | 67.9 |
| | **CNN** | 74.7 | 23.6 | 74.0 |
| | **AEA** | 75.1 | 22.4 | 74.3 |
| | **CGNN** | 89.6 | 9.3 | 89.4 |
| | **Proposed GAE** | **91.3** | **8.3** | **90.9** |

report the detection performance of unsupervised detectors (ARIMA, AEA, and GAE) against unseen FDIAs. However, they report the performance when supervised detectors (FNN, LSTM, CNN, and CGNN) are trained on benign and both FDIA types (general and replay attacks presented in Section II-C). Thus, the reported performance for the supervised detectors in Tables III and IV is based on seen FDIA types. Later in Tables V and VI, we present the detection performance when the supervised detectors encounter unseen FDIA types that are not part of the training set.

*1) Detection Performance in Generalized Settings:* Table III reports the average detection performance results of the generalized experiments described in Section II-D1 where the models are tested against unseen topological configurations with various connectivity levels that are not part of the training phase. According to the experimental results, as the system size increases, the performance of the detectors improves. Such an improvement is due to the increase in the amount of data with larger systems with more nodes and edges and hence more measurement readings to train on. Specifically, testing the generalized detectors on the 39-bus system topologies offers average DR improvements of $1.5 - 2.1\%$ compared to the 14-bus system. Furthermore, testing on the 118-bus system topologies provides superior DRs by $4.6 - 6.2\%$ compared to the 14-bus system. Overall, the topology-aware generalized detectors offer detection improvements of about $9 - 38.8\%$ compared to generalized topology-unaware (classical ML models) benchmark detectors. Such an improvement is due to capturing the spatial features within the graphs by performing the graph convolution operation. Specifically, the detection performance improvements of the topology-aware detectors (CGNN and GAE) compared to the topology-unaware (ARIMA, FNN, LSTM, CNN, and AEA) detectors are listed as follows.

- In the 14-bus system, the CGNN-based detector offers performance improvements of $9.3 - 26.8\%$ in DR, $9 - 36.9\%$ in FAR, and $9.3 - 27\%$ in ACC. The GAE-based detector offers further improvements of $11.5 - 29\%$ in DR, $10.6 - 38.5\%$ in FAR, and $12 - 29.7\%$ in ACC.
- In the 39-bus system, the CGNN-based detector offers performance improvements of $9.6 - 27.3\%$ in DR, $9 - 37.1\%$ in FAR, and $9.8 - 27.5\%$ in ACC. The GAE-based detector offers further improvements of $11.9 - 29.6\%$ in DR, $10.8 - 38.9\%$ in FAR, and $12.1 - 29.8\%$ in ACC.
- In the 118-bus system, the CGNN-based detector offers performance improvements of $10.4 - 28.4\%$ in DR, $9.2 - 37.9\%$ in FAR, and $10.9 - 29\%$ in ACC. The GAE-based detector offers further improvements of $12 - 30\%$ in DR, $10.1 - 38.8\%$ in FAR, and $12.3 - 30.4\%$ in ACC.

*2) Detection Performance in Topology-Specific Settings:* Table IV shows the average detection performance results of the topology-specific experiments described in Section II-D2. Based on the conducted experiments, as the system size increases, the performance of the detectors slightly improves, which is due to the slight increase in the amount of data

TABLE V
DETECTION PERFORMANCE OF GENERALIZED DETECTORS AGAINST UNSEEN FDIAS (%)

| Bus System Size | Detector | Metric | Unseen Attack | | Avg |
|---|---|---|---|---|---|
| | | | Replay | General | |
| 14-bus | FNN | DR | 58.3 | 55.0 | 56.7 |
| | | FAR | 41.6 | 44.5 | 43.1 |
| | | ACC | 56.6 | 53.8 | 55.2 |
| | LSTM | DR | 65.4 | 62.3 | 63.9 |
| | | FAR | 33.6 | 36.9 | 35.3 |
| | | ACC | 64.2 | 61.2 | 62.7 |
| | CNN | DR | 69.8 | 66.5 | 68.2 |
| | | FAR | 28.5 | 31.1 | 29.8 |
| | | ACC | 69.4 | 66.0 | 67.7 |
| | CGNN | DR | 83.0 | 79.7 | 81.4 |
| | | FAR | 15.7 | 18.5 | 17.1 |
| | | ACC | 81.8 | 78.8 | 80.3 |
| | Proposed GAE | DR | **95.1** | **92.2** | **93.6** |
| | | FAR | **4.0** | **5.4** | **4.7** |
| | | ACC | **94.5** | **91.7** | **93.1** |
| 39-bus | FNN | DR | 61.0 | 58.2 | 59.6 |
| | | FAR | 40.6 | 43.7 | 42.2 |
| | | ACC | 59.1 | 56.0 | 57.6 |
| | LSTM | DR | 68.0 | 65.2 | 66.6 |
| | | FAR | 31.7 | 33.9 | 32.8 |
| | | ACC | 66.9 | 63.7 | 65.3 |
| | CNN | DR | 72.8 | 69.4 | 71.1 |
| | | FAR | 25.8 | 28.7 | 27.3 |
| | | ACC | 71.9 | 68.6 | 70.3 |
| | CGNN | DR | 86.2 | 82.6 | 84.4 |
| | | FAR | 12.7 | 16.2 | 14.5 |
| | | ACC | 85.6 | 82.7 | 84.2 |
| | Proposed GAE | DR | **96.8** | **94.6** | **95.7** |
| | | FAR | **2.2** | **3.2** | **2.7** |
| | | ACC | **95.9** | **94.0** | **95.0** |
| 118-bus | FNN | DR | 65.2 | 62.6 | 63.9 |
| | | FAR | 34.9 | 38.7 | 36.8 |
| | | ACC | 64.1 | 60.8 | 62.5 |
| | LSTM | DR | 73.2 | 70.0 | 71.6 |
| | | FAR | 26.9 | 29.9 | 28.4 |
| | | ACC | 71.4 | 68.5 | 70.0 |
| | CNN | DR | 77.4 | 74.5 | 76.0 |
| | | FAR | 20.6 | 23.5 | 22.1 |
| | | ACC | 76.7 | 73.7 | 75.2 |
| | CGNN | DR | 90.8 | 87.9 | 89.4 |
| | | FAR | 8.7 | 12.0 | 10.4 |
| | | ACC | 91.2 | 87.8 | 89.5 |
| | Proposed GAE | DR | **99.4** | **98.8** | **99.1** |
| | | FAR | **0.2** | **0.4** | **0.3** |
| | | ACC | **99.2** | **98.3** | **98.7** |

TABLE VI
DETECTION PERFORMANCE OF TOPOLOGY-SPECIFIC DETECTORS AGAINST UNSEEN FDIAS (%)

| Bus System Size | Detector | Metric | Unseen Attack | | Avg |
|---|---|---|---|---|---|
| | | | Replay | General | |
| 14-bus | FNN | DR | 44.9 | 42.0 | 43.5 |
| | | FAR | 55.3 | 58.4 | 56.9 |
| | | ACC | 43.7 | 41.1 | 42.4 |
| | LSTM | DR | 53.9 | 50.7 | 52.3 |
| | | FAR | 49.2 | 51.8 | 50.5 |
| | | ACC | 52.4 | 49.3 | 50.9 |
| | CNN | DR | 58.9 | 56.6 | 57.8 |
| | | FAR | 38.7 | 42.5 | 40.6 |
| | | ACC | 59.2 | 56.0 | 57.6 |
| | CGNN | DR | 72.6 | 69.6 | 71.1 |
| | | FAR | 30.6 | 32.9 | 31.8 |
| | | ACC | 72.7 | 69.7 | 71.2 |
| | Proposed GAE | DR | **88.2** | **85.6** | **86.9** |
| | | FAR | **12.9** | **11.1** | **12.0** |
| | | ACC | 87.6 | 85.2 | 86.4 |
| 39-bus | FNN | DR | 47.7 | 44.9 | 46.3 |
| | | FAR | 53.7 | 56.3 | 55.0 |
| | | ACC | 46.1 | 44.0 | 45.1 |
| | LSTM | DR | 56.0 | 53.6 | 54.8 |
| | | FAR | 43.3 | 45.8 | 44.6 |
| | | ACC | 54.9 | 51.8 | 53.4 |
| | CNN | DR | 63.0 | 59.9 | 61.5 |
| | | FAR | 34.9 | 38.0 | 36.5 |
| | | ACC | 63.0 | 60.1 | 61.6 |
| | CGNN | DR | 76.5 | 73.7 | 75.1 |
| | | FAR | 22.0 | 25.5 | 23.8 |
| | | ACC | 76.1 | 72.8 | 74.5 |
| | Proposed GAE | DR | **89.9** | **87.5** | **88.7** |
| | | FAR | **10.4** | **9.2** | **9.8** |
| | | ACC | **89.0** | **86.9** | **87.9** |
| 118-bus | FNN | DR | 50.3 | 47.7 | 49.0 |
| | | FAR | 50.1 | 53.9 | 52.0 |
| | | ACC | 48.4 | 45.1 | 46.8 |
| | LSTM | DR | 56.7 | 54.4 | 55.6 |
| | | FAR | 43.0 | 46.4 | 44.7 |
| | | ACC | 55.7 | 52.9 | 54.3 |
| | CNN | DR | 63.2 | 60.7 | 62.0 |
| | | FAR | 35.0 | 38.3 | 36.7 |
| | | ACC | 62.1 | 59.4 | 60.8 |
| | CGNN | DR | 81.1 | 78.4 | 79.8 |
| | | FAR | 18.4 | 21.7 | 20.1 |
| | | ACC | 81.1 | 77.7 | 79.4 |
| | Proposed GAE | DR | **91.9** | **90.7** | **91.3** |
| | | FAR | **7.9** | **8.7** | **8.3** |
| | | ACC | **91.6** | **90.2** | **90.9** |

(i.e., more measurement readings) in bigger systems, yet the generalized detectors offer better improvements due to being trained on more data from bigger systems topologies. Specifically, testing the topology-specific detectors on the 39-bus system topologies offers slight DR improvements of $0.5-1.8\%$ compared to the 14-bus system. Additionally, testing on the 118-bus system topologies provides improved DRs by $3-4.9\%$ compared to the 14-bus system. Overall, in a topology-specific setting, the topology-aware detectors still offer stable performance with DR improvements of $13-33.9\%$, $14-34.8\%$, and $14.5-35.1\%$ compared to topology-unaware topology-specific benchmark detectors when tested on the 14, 39, and 118-bus systems, respectively.

From Tables III and IV, we interpret that the generalized detectors offer superior detection performance compared to the topology-specific detectors. Such superior performance is because generalized detectors capture more data from different topologies and hence have better capabilities of detecting attacks from new topological reconfigurations, unlike topology-specific detectors that are trained on a specific topological configuration with less data. Specifically, when topology-unaware benchmark detectors are tested in topology-specific settings, the DRs deteriorate by $10.5-11.6\%$, $11.4-12.2\%$, and $12.1-13\%$, in 14, 39, and 118-bus systems, respectively, compared to generalized settings. Such high deterioration rates are because topology-unaware detectors in topology-specific settings lack the capability of capturing the spatial features [6], [8] and the topological reconfigurations. However, the topology-aware detectors still offer decent DRs in a topology-specific setting that degrade only by $6.7 - 6.8\%$, $7 - 7.1\%$, and $7.8 - 8\%$, in 14, 39, and 118-bus systems, respectively, compared to a generalized setting since they still capture the spatial features. In both, generalized and topology-specific settings, the proposed GAE-based detector outperforms the benchmark detectors, which is due to the advantages listed next (in Section IV-D3).

*3) Detection Performance Against Unseen FDIAs:* Tables III and IV reported the detection performance when supervised detectors are trained on both FDIAs, which is not always a valid assumption in practice since the detector may encounter an attack strategy (zero-day attack) different from the ones that it has been trained on [25], [26]. Therefore, to capture real-life scenarios, we also report the performance of the supervised detectors when trained on benign data as well as one attack function while being tested on the other attack function as an unseen attack. Tables V and VI report the performance of the supervised generalized and topology-specific detectors, respectively, (compared to the unsupervised GAE model) when encountering unseen attacks. Specifically, the deterioration in DR compared to detecting seen attacks is summarized next.

- According to Table V, generalized topology-unaware detectors (FNN, LSTM, and CNN) deteriorate by $12.2 - 15\%$, $11.1 - 13.7\%$, and $10.4 - 13.2\%$, in the 14, 39, and 118-bus system, respectively, compared to encountering seen attacks. Similarly, the generalized benchmark topology-aware detector (CGNN) deteriorates by $10.1\%$, $9\%$, and $8.3\%$, in the 14, 39, and 118-bus systems, respectively, compared to encountering seen attacks.
- According to Table VI, topology-specific topology-unaware detectors (FNN, LSTM, and CNN) deteriorate by $13 - 17\%$, $10.2 - 14.7\%$, and $12.8 - 14.9\%$, in the 14, 39, and 118-bus system, respectively, compared to encountering seen attacks. Similarly, the CGNN-based detector deteriorates by $13.5\%$, $11.2\%$, and $9.8\%$, in the 14, 39, and 118-bus systems, respectively, compared to encountering seen attacks.

*4) Remarks:* Next, we point out some remarks.

*a) Training Nature:* Supervised detectors fail to fully detect unseen attacks since such attacks are not part of their training set and hence the models are unfamiliar with unseen attack patterns [26]. However, unsupervised models need to be trained only on benign behavior and hence they are able to mark deviations as malicious behavior [25].

*b) Classical and Graph Autoencoders:* The unsupervised autoencoder-based AEA and GAE models offer the best detection performance compared to the classical and graph-based models, respectively. The main difference between a classical stacked autoencoder (e.g., AEA) and a graph autoencoder (e.g., GAE) is that GAEs adopt graph layers that perform the Chebyshev convolution operation to capture the spatial aspects of the power system topological configurations that classical autoencoders fail to capture. Hence, while a classical SAE works on two-dimensional (2D) data (value of measurement and timestamp), a GAE operates on data that is represented by a graph and thus accounts for the measurement value, timestamp, and the adjacency matrix describing the graph connectivity. As such, a GAE is able to capture the spatial relationships among measurement values at different nodes. Extracting such aspects is crucial herein to offer a generalized detection scheme that works well even in cases of seasonal topological reconfiguration. Therefore, AEA and GAE are considered topology-unaware and topology-aware detectors, respectively, where the proposed GAE-based detector enhances the DR by $12 - 16.2\%$ compared to the classical AEA.

*c) Resisting FDIAs:* In addition to the performance metrics reported in Tables III, IV, V, and VI, our detection performance analysis in Sections IV.D1 and IV.D2 relied primarily on DR to determine how well the detectors correctly identify FDIAs. We can further verify that the proposed GAE-based defense strategy can better resist FDIAs compared to benchmark detectors by reporting the false negative rate (miss rate (MR) = 100 - DR) that the detector offers. MR determines the portion of malicious samples (FDIAs) that are being incorrectly detected as benign. Using Tables V and VI, in generalized settings against unseen attacks, the proposed GAE-based detector offers MRs of $0.9 - 6.4\%$ whereas the best performing benchmark detector (CGNN) offers MRs of $10.6 - 18.6\%$. Thus, the proposed GAE-based detector offers improved MR by $9.7 - 12.2\%$ compared to the CGNN-based detector. Lower MRs translates to the detector's ability to resist more malicious samples (FDIAs) although such samples are not present in the training set of the proposed GAE model.

*d) Advantages of GAE:* Besides offering generalized detection against FDIAs within unseen topologies, the reported detection performance of the proposed GAE-based detector is based on completely unseen malicious data. The detection improvement that the proposed GAE detector offers is due to the (a) generalized training nature that learns the power measurements sequences from different topologies over prolonged timestamps, (b) presence of Chebyshev graph convolution layers helps capture the spatial aspect of the data since they model the grid topologies using graphs, and (c) deep stacked structure with attention mechanism, which helps extract the complex patterns and learn distinctive features from the data. Such characteristics lead to improved detection performance compared to classical ML-based detection in addition to offering linear scalability, as discussed next (in Section IV-E).

### *E. Model Analysis*

We study three aspects related to the investigated detectors. First, we present the time taken to train the models offline. Second, we study the scalability of the models in terms of the decision time and complexity. Third, we investigate the optimal number of needed topologies to develop a generalized FDIAs detector.

*1) Offline Training Time:* All models are trained offline using an NVIDIA GeForce RTX 2070 hardware accelerator utilizing Keras sequential API. The offline training process of a topology-unaware detector takes around $2 - 4$ hours, whereas it takes $4.5 - 6$ hours to train a topology-aware detector. The variation in training time is due to the amount of features to

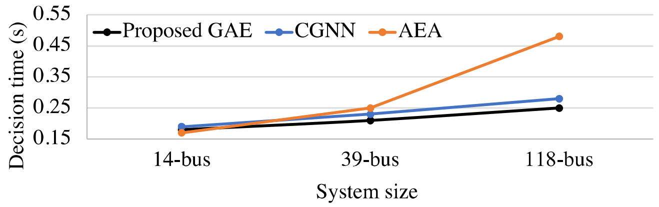
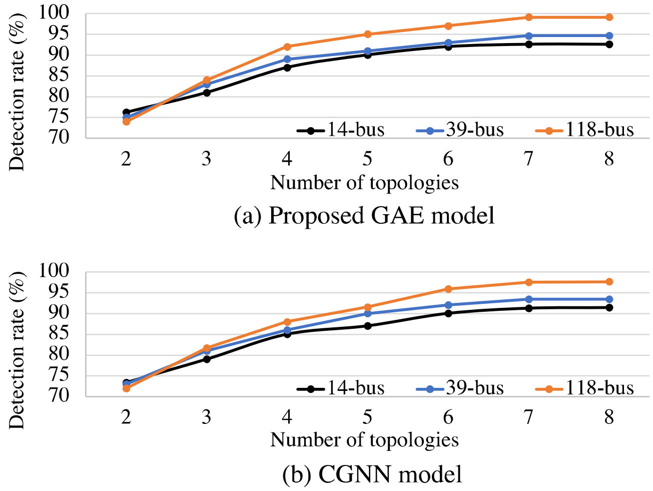


Fig. 6. Illustration of the linear scalability of the proposed model.



(a) Proposed GAE model

(b) CGNN model

Fig. 7. DR of the topology-aware models as training topologies increases.

train on. Topology-aware models take relatively more time to train since they capture more features (i.e., spatial features of topological configurations). Thus, topology-aware detectors offer superior detection performance at the expense of training time. However, training time is not a pressing issue herein since system operators may train the models on available datasets offline periodically, not in real-time.

*2) Model Scalability:* Fig. 6 demonstrates the scalability of the topology-aware detectors compared to AEA. We selected AEA in this comparison since it presents the best performing unsupervised topology-unaware benchmark detector against unseen attacks (as shown in Tables III and IV). We study scalability in terms of the real-time taken to make a decision (during online testing) on a sample as the system bus-size increases. As the system size increases, the number of trainable parameters also increases and the model becomes more complex [6]. Fig. 6 shows that, unlike the classical AEA model, the topology-aware models are linearly scalable as the system size (complexity) increases. Unlike topology-unaware detectors, topology-aware detectors capture spatial features and hence increasing the system size offers more features to capture, which helps boost the detection performance with a slight linear increase in the decision time.

*3) Number of Topologies:* Fig. 7 illustrates the improvement in DR as the number of topologies $\mathbf{\Gamma}$ used to train the generalized topology-aware models (CGNN and GAE) increases. It can be shown that the reported DRs improve as we increase the number of training topologies from 2 to 7. After that, when the model is trained on eight topologies, the DRs saturate, i.e., becomes similar to when training on seven topologies. Therefore, based on the conducted experiments, due to the saturation nature of machine learning models as sample size increases and to avoid overfitting [58], we train the generalized models on seven topologies and validate on one for optimal results. Then, we test on the remaining unseen two topologies throughout the different experiments described in Section II-D1.

## V. Conclusion and Future Work

This paper investigated the use of GNN-based (topology-aware) FDIAs detectors compared to classical ML-based (topology-unaware) ones in generalized and topology-specific settings. The generalized models utilize datasets from different topological configurations of multiple system sizes. The datasets are generated using a generative spatio-temporal model employing stochastic geometry. Subsequently, we proposed a GAE-based detector that is (a) topology-aware as it captures spatio-temporal features of power systems through Chebyshev graph convolution layers, (b) generalized as it is trained on comprehensive graphs reflecting various realizations of power system topologies, and (c) able to detect unseen FDIAs as it presents an unsupervised anomaly detection. As a result, the proposed GAE detector offered DRs of up to $99.1\%$ against unseen FDIAs in unseen topologies, reflecting superior DR by $11.5 - 30\%$ compared to existing ML-based detectors. This paper focused on detecting FDIAs at the graph level (i.e., detecting the overall system state). In our future work, we will focus on localizing FDIAs (i.e., detecting the attacked node) and classifying different FDIA types. Detection, localization, and classification of FDIAs will help mitigate such attacks in smart power grids. We will also examine the performance of detectors against coordinated cyber-physical attacks.

## 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] D. An *et al.*, “Data integrity attack in dynamic state estimation of smart grid: Attack model and countermeasures,” *IEEE Trans. on Automation Science and Engineering*, vol. 19, no. 3, pp. 1631–1644, Jul. 2022.

[3] Y. Liu *et al.*, “False data injection attacks against state estimation in electric power grids,” *ACM Trans. on Information and System Security (TISSEC)*, vol. 14, no. 1, pp. 1–33, Jun. 2011.

[4] 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.

[5] 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.

[6] 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.

[7] K. Hamedani *et al.*, “Detecting dynamic attacks in smart grids using reservoir computing: A spiking delayed feedback reservoir based approach,” *IEEE Trans. on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 253–264, Jun. 2020.

[8] A. S. Musleh *et al.*, “A survey on the detection algorithms for false data injection attacks in smart grids,” *IEEE Trans. on Smart Grid*, vol. 11, no. 3, pp. 2218–2234, May 2020.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[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] 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.

[16] 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.

[17] 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.

[18] E. Drayer *et al.*, “Detection of false data injection attacks in power systems with graph fourier transform,” in *IEEE Glbl. Conf. on Signal and Info. Proc.* Anaheim, CA, USA, 26–29 Nov. 2018, pp. 135–140.

[19] 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.

[20] 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.

[21] D. Owerko, F. Gama, and A. Ribeiro, “Optimal power flow using graph neural networks,” in *IEEE Intl. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*. Barcelona, Spain, 04–08 May. 2020, pp. 5930–5934.

[22] O. Boyaci *et al.*, “Cyberattack detection in large-scale smart grids using chebyshev graph convolutional networks,” in *Intl. Conf. on Electrical and Electronics Engr.* Alanya, Turkey, 29–31 Mar. 2022, pp. 217–221.

[23] O. Boyaci *et al.*, “Joint detection and localization of stealth false data injection attacks in smart grids using graph neural networks,” *IEEE Trans. on Smart Grid*, vol. 13, no. 1, pp. 807–819, Jan. 2022.

[24] S. Jazebi *et al.*, “Distribution network reconfiguration in the presence of harmonic loads: Optimization techniques and analysis,” *IEEE Trans. on Smart Grid*, vol. 5, no. 4, pp. 1929–1937, 2014.

[25] A. Takiddin *et al.*, “Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids,” *IEEE Systems Journal*, vol. 16, no. 3, pp. 4106–4117, Sept. 2022.

[26] A. Takiddin *et al.*, “Robust data-driven detection of electricity theft adversarial evasion attacks in smart grids,” *IEEE Trans. on Smart Grid*, vol. 14, no. 1, pp. 663–676, Jan. 2023.

[27] W.-T. Li *et al.*, “Location identification of power line outages using pmu measurements with bad data,” *IEEE Trans. on Power Systems*, vol. 31, no. 5, pp. 3624–3635, Sept. 2016.

[28] T. Aziz *et al.*, “A methodology to prevent cascading contingencies using bess in a renewable integrated microgrid,” *Intl. Journal of Electrical Power & Energy Systems*, vol. 110, pp. 737–746, Sept. 2019.

[29] X. Wang *et al.*, “Detection and isolation of false data injection attacks in smart grid via unknown input interval observer,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3214–3229, Apr. 2020.

[30] X. Luo *et al.*, “Interval observer-based detection and localization against false data injection attack in smart grids,” *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 657–671, Jan. 2021.

[31] S. Lakshminarayana *et al.*, “Data-driven false data injection attacks against power grids: A random matrix approach,” *IEEE Trans. on Smart Grid*, vol. 12, no. 1, pp. 635–646, Jan. 2021.

[32] B. Li *et al.*, “Detection of false data injection attacks on smart grids: A resilience-enhanced scheme,” *IEEE Trans. on Power Systems*, vol. 37, no. 4, pp. 2679–2692, Jul. 2022.

[33] M. Mohammadpourfard *et al.*, “Identification of false data injection attacks with considering the impact of wind generation and topology reconfigurations,” *IEEE Trans. on Sustainable Energy*, vol. 9, no. 3, pp. 1349–1364, Jul. 2018.

[34] H.-M. Chung *et al.*, “Local cyber-physical attack for masking line outage and topology attack in smart grid,” *IEEE Trans. on Smart Grid*, vol. 10, no. 4, pp. 4577–4588, Jul. 2019.

[35] Y. Liu and L. Cheng, “Relentless false data injection attacks against kalman-filter-based detection in smart grid,” *IEEE Trans. on Control of Network Systems*, vol. 9, no. 3, pp. 1238–1250, Sept. 2022.

[36] J. B. Hansen *et al.*, “Power flow balancing with decentralized graph neural networks,” *IEEE Trans. on Power Systems*, pp. 1–11, Aug. 2022.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] D. Deka, S. Vishwanath, and R. Baldick, “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.

[42] 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.

[43] T. Courtat *et al.*, “Mathematics and morphogenesis of cities: A geometrical approach,” *Physical Review E*, vol. 83, p. 036106, Mar. 2011.

[44] S. A. R. Zaidi and M. Ghogho, “Stochastic geometric analysis of black hole attack on smart grid communication networks,” in *IEEE Int. Conf. Smart Grid Comm.* Tainan, Taiwan, 5–8 Nov. 2012, pp. 716–721.

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

[46] M. A. Hasnat and M. Rahnamay-Naeini, “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.

[47] E. D. Dunlop *et al.*, “Guidelines for PV power measurement in industry,” *European Commission*, 2010.

[48] C. Sammut and G. I. Webb, Eds., *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2011, pp. 600–601.

[49] 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.

[50] L. Wu *et al.*, *Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer, Jan. 2022.

[51] A. Ortega *et al.*, “Graph signal processing: Overview, challenges, and applications,” *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[52] M. Defferrard *et al.*, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *30th Int. Conf. Neural Inf. Process. Syst.* Barcelona, Spain, 5–10 Dec. 2016, p. 3844–3852.

[53] L. Ruiz, F. Gama, A. G. Marques, and A. Ribeiro, “Invariance-preserving localized activation functions for graph neural networks,” *IEEE Trans. on Signal Processing*, vol. 68, pp. 127–141, Nov. 2020.

[54] A. Takiddin *et al.*, “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.

[55] V. Krishna *et al.*, “ARIMA-Based modeling and validation of consumption readings in power grids,” in *Critical Information Infrastructures Security*. Springer Intl. Publishing, May 2016, pp. 199–210.

[56] 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, Jul. 2021, pp. 1–6.

[57] A. Takiddin *et al.*, “Detecting electricity theft cyber-attacks in AMI networks using deep vector embeddings,” *IEEE Systems Journal*, vol. 15, no. 3, pp. 4189–4198, Sept. 2021.

[58] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.