Graph Autoencoder-based Detection of Unseen False Data Injection Attacks in Smart Grids

Abdulrahman Takiddin[1], Muhammad Ismail[2], Rachad Atat[3], Katherine R. Davis[1], and Erchin Serpedin[1⋆]

[1] Texas A&M University, College Station TX 77843, USA,
abdulrahman.takiddin@tamu.edu, katedavis@tamu.edu, eserpedin@tamu.edu
[2] Tennessee Tech University, Cookeville TN 38505, USA,
mismail@tntech.edu
[3] Texas A&M University at Qatar, Doha, Qatar,
rachad.atat@qatar.tamu.edu

Abstract. A major concern in smart power grids is when malicious or manipulated data is injected into measurement data due to malicious activities. Several approaches have been investigated to counter such false data injection attacks (FDIAs). However, such data-driven detectors present two major limitations. First, they neglect capturing the grid's spatial characteristics. Second, they offer limited attack identification to familiar types of FDIAs since they are present within the model's train sets. To conquer such limitations, we propose the use of an artificial intelligence-based graph autoencoder (GAE) for FDIAs detection. Our proposed detector offers three main advantages compared to existing detectors. First, it employs the operation of graph convolution to apprehend the grid's spatial characteristics. Second, it offers an unsupervised autoencoder-based anomaly detection that requires only benign samples under normal operation for training. Third, it outperforms existing detectors by $16-47\%$ in FDIAs detection rate (DR) when tested against unseen FDIAs on an IEEE 39-bus system.

Keywords: Cyberattacks, Graph Neural Network, Machine Learning, Smart Grid.

1 Introduction

The decision making within smart power grids is highly dependent on measurement data collected from several components among the power grid for proper operation [1]. Therefore, the integrity of the collected data is critical to ensure the reliability of the system and for stable operation. However, malicious entities may carry out stealthy attacks (e.g., false data injection attacks (FDIAs)) to manipulate measurement data from sensors and hence jeopardize the integrity of the power grid data [2]. As a result, decision making will be based on inaccurate measurement values, which might lead to instabilities or overload in the system

⋆ This work was supported by NSF EPCN Awards 2220346 and 2220347.

[3]. Unfortunately, such stealthy attacks can bypass existing bad data detectors [4]. Thus, more complex attempts have been proposed to detect such FDIAs employing multiple data-driven-based approaches.

1.1 Related Work and Limitations

Several approaches have been investigated to counter such FDIAs. We divide these approaches into three main categories, namely, shallow machine learning (ML), deep learning (DL), and graph-based models. Next, we report the performance of relevant studies along with their limitations.

Shallow ML Models Relevant shallow ML-based FDIAs detection schemes employ the following. Support vector machines (SVMs) provided 82% in F1-Score [5]. A decision trees model offered an F1-Score of 88% [6]. A random forest model reported an attack detection rate (DR) of 93% [7]. Nevertheless, such shallow models do not apprehend the patterns and spatial characteristics of the data [8]. They also present supervised learning that offers detection limited to the familiar attacks that are seen the models' train sets. Hence, they are susceptible to new types of attacks (i.e., not present in the train sets).

DL Models DL-based detectors have been proposed to apprehend the pattern characteristics within the data [9]. To achieve this, DL-based detectors employ the following models. A feedforward neural network (FNN) model provided an accuracy (ACC) score of 90% [10]. A convolutional neural network (CNN) model offered an ACC score of 93% [11]. A recurrent neural network (RNN) model offered a DR of 96% [12]. Although these DL-based detectors are able to apprehend the data patterns that are sophisticated using deep neural networks, they still fail to detain the system's spatial characteristics [4]. Also, they still offer limited detection performance against new types of attacks that are not present in the train sets [13].

Graph Models Graph-based detectors have been proposed to capture the grid spatial information. In particular, graph-based detectors employ graph signal processing (GSP) and graph neural network (GNN) models. GPS models employ spectral filters that are manually designed [14], [15], [16] and provided DRs of nearly 90%. However, the custom design of the filter limits the scalability of the model [4]. To overcome this, GNN models have been proposed. Specifically, a convolutional GNN (CGNN) model that incorporates the GSP operation automatically and utilizes undirected graphs offered DRs of $83-96\%$ [4]. Despite the provided advantages, existing GNN-based FDIAs detection schemes still offer attack detection only against seen attack types that are part of their training sets due to their supervised learning nature. However, in practice, the system might encounter new unseen attack types (i.e., zero-day attacks) that are different than the types the detector has been trained on.

According to the presented limitations, there is a need to improve the attack identification performance of existing state-of-the-art models. This could be achieved by proposing an artificial intelligence-based robust detection strategy that apprehends the system's sophisticated patterns as well as the spatial characteristics while offering robust identification against new types of attacks that are not present in the train sets.

1.2 Contributions

We conquer the drawbacks of existing FDIAs detection schemes by proposing a graph-based unsupervised anomaly detector. The proposed detector employs a graph autoencoder (GAE) providing three major benefits. The advantages of the proposed detector are highlighted by comparing it to various data-driven detection strategies. Specifically, the GAE model offers the following.

- It presents a deep structure with stacked graph encoder and decoder layers that detain the complex patterns of the measurement data. It is also able to detain the grid's spatial characteristics as it employs a graph Chebyshev convolution operation.
- It offers detection of new FDIA types (i.e., types that are not present during training) as it represents an unsupervised anomaly detection scheme that relies only on benign data during training. During testing, it marks unseen malicious samples of under-attack operation according to the presented dissimilarity from the features of normal operation (benign samples) that were learned during the training stage.
- It offers a superior DR of 90.2% against unseen attacks in an IEEE 39-bus system, providing DR enhancements of $16-47\%$ compared to a comprehensive list of benchmarks including shallow, deep, and graph-based detectors.

The layout of the paper is as follows. Section 2 describes the data preparation and the investigated attack types. Section 3 presents the architecture of the GAE model. Section 4 introduces the benchmark detectors and reports the detection performance. Section 5 concludes the outcomes of this work.

2 Data Preparation

For the training and testing of the studied models, we employ an IEEE 39-bus system. To generate malicious samples replicating the under-attack system operation, we acquire three FDIA types [17].

2.1 System Model

In this work, we detain the spatial and temporal characteristics of the power system (i.e., IEEE 39-bus system). Specifically, we model the system via an undirected graph. Within the graph, buses are depicted by vertices (nodes) whereas

power lines are denoted by edges. Fig. 1 shows the adopted IEEE 39-bus system modeled as an undirected graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ denote the undirected graph with vertices $\mathcal{V}$, edges $\mathcal{E}$, and weighted adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$. In $\mathcal{G}$, when bus i is connected to bus j, a weight W_{ij} is allied to an edge $e = (i, j)$ according to the line admittance.

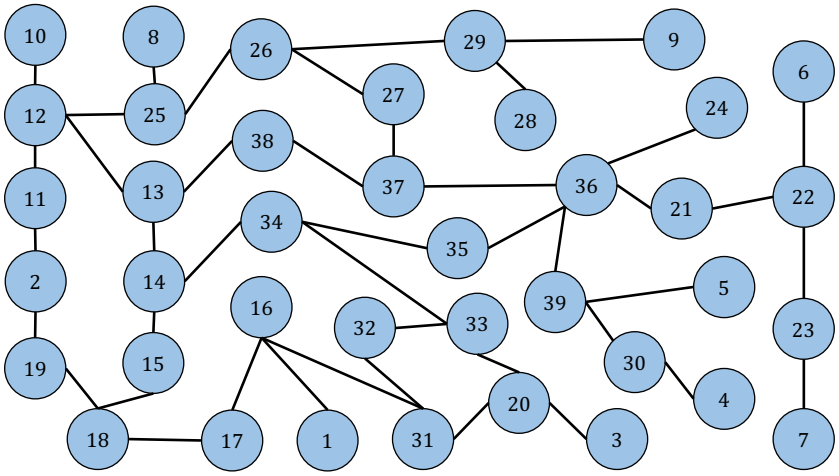

Fig. 1. IEEE 39-Bus System Graph Illustration.

In addition to the spatial aspects, temporal characteristics referring to power injections and flows are also captured where $\mathcal{V}$ and $\mathcal{E}$ are accompanied with features. To detain such features, we adopt an analysis of the power flow through Newton's method via MATLAB MATPOWER toolbox [18]. This is carried out to establish the flows of real and reactive power in the system. Specifically, the features of $\mathcal{V}$ comprise the active power (i.e., real power demand) P_i and reactive power demand Q_i in MW and MVAr, respectively. The features of $\mathcal{E}$ comprise the real power flow P_{ij} from bus i to bus j in MW and the reactive power flow Q_{ij} from bus i to bus j in MVAr.

2.2 Benign Data

We adopt the feature values discussed above to represent measurement data of normal operation, which results in generating benign samples denoted as $\boldsymbol{x}_{\mathrm{b}}(t, i)$ at bus i and timestamp t. Specifically, over a period of half a year, each hour, we report 4 power dynamics timestamps, which leads to approximately $17,000$ timestamps in total.

2.3 Malicious Data

To constitute the malicious data representing the under-attack system operation, we adopt three FDIA functions, namely, direct, replay, and general attacks. These attacks are applied to $\boldsymbol{x}_{\mathrm{b}}(t,i)$ and bypass existing bad data detectors since they present similar data patterns as benign samples [19]. The generated malicious samples are denoted as $\boldsymbol{x}_{\mathrm{m}}(t,i)$ at bus i and timestamp t. The three FDIA types are described next.

The direct attack applies specific perturbations bounded by a scaling factor $|\alpha| \leq 0.05$ that are injected into benign samples. For instance,

$$\boldsymbol{x}_{\mathrm{m}}(t,i) = \boldsymbol{x}_{\mathrm{b}}(t,i) + \alpha.\boldsymbol{x}_{\mathrm{b}}(t,i). \tag{1}$$

The replay attack generates malicious samples throughout a false repetition of readings from a prior timestamp $t-1$. As a result, the true reading of a present timestamp t is replaced as follows

$$\boldsymbol{x}_{\mathrm{m}}(t,i) = \boldsymbol{x}_{\mathrm{b}}(t-1,i). \tag{2}$$

The general attack [20] uses a true measurement value interval along with a binary β and uniform random $0 \leq \gamma \leq 1$ variables to create malicious samples where

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

2.4 Dataset Splitting

The generated sample types (benign $\boldsymbol{x}_{\mathrm{b}}$ and malicious $\boldsymbol{x}_{\mathrm{m}}$) are equal in number. Supervised models are required to be trained and tested on both sample types ($\boldsymbol{x}_{\mathrm{b}}$ and $\boldsymbol{x}_{\mathrm{m}}$). On the other hand, unsupervised models necessitate to be trained on $\boldsymbol{x}_{\mathrm{b}}$, but they are still tested on both samples types. To carry out the experiments, we split the samples into three sets where the training $\boldsymbol{X}_{\mathrm{TR}}$, validation $\boldsymbol{X}_{\mathrm{VA}}$, and testing $\boldsymbol{X}_{\mathrm{TS}}$ sets represent 80%, 10%, and 10% of samples, respectively. To avoid any bias, all sets have equal numbers of samples of both types.

3 GAE-Based Detector

To detect FDIAs, we put forward an unsupervised GAE-based anomaly detector. For training, the proposed detector relies on graph expressions of data during normal operation (benign samples) [21]. Hence, it offers identification ability against new types of attacks that are not present in the train sets.

3.1 GAE Model Architecture

The GAE model employs an autoencoder that utilizes graph encoding and decoding layers. Such layers help when it comes to apprehending the graph expressions

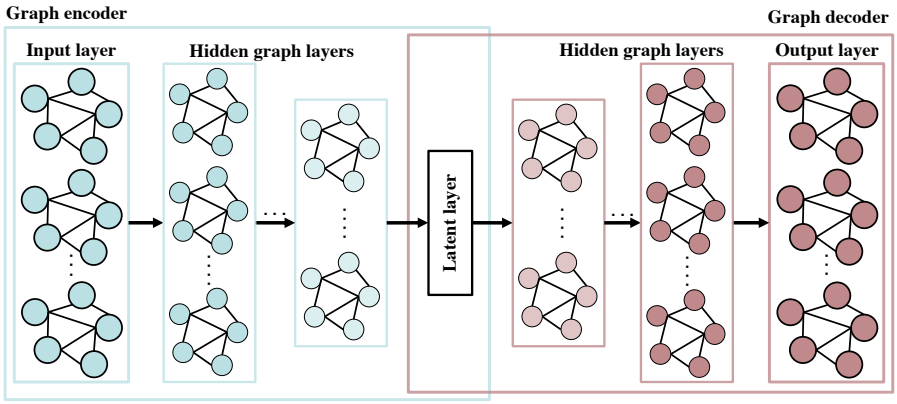

Fig. 2. Proposed Unsupervised GAE Model Architecture.

of normal operation (i.e., benign data) via a data reconstruction process [22], [23]. The overall process is illustrated in Fig. 2.

The GAE model operates as follows. First, it takes benign samples' $\boldsymbol{x}_{\mathrm{b}}$ with $[P_i, Q_i] \in \mathbb{R}^{n\times 2}$ measurements as input. Then, following the input layer, the encoder $\boldsymbol{E}$ with $\mathcal{L}_{\mathrm{E}}$ hidden graph encoding convolution layers are placed. Applying the convolution operation to graph signals is essential when detaining the grid's spatial characteristics. The graph encoding block is responsible for compressing the data. On the graph encoder side, the number of channels depicting what is fed to the convolution layers is denoted by $c_{l_{\mathrm{E}}}$ in a hidden encoding layer l_{E}. The input and output of l_{E} are $\boldsymbol{X}^{l_{\mathrm{E}}-1} \in \mathbb{R}^{n\times c_{l_{\mathrm{E}}-1}}$ and $\boldsymbol{X}^{l_{\mathrm{E}}} \in \mathbb{R}^{n\times c_{l_{\mathrm{E}}}}$, respectively. The presence of the encoding layers helps in capturing the spatial characteristics as well as constructing the output tensor, which is expressed next

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

In (4), $\boldsymbol{\theta}^{l_{\mathrm{E}}} \in \mathbb{R}^{K\times c_{l_{\mathrm{E}}-1}\times c_l}$, $\boldsymbol{b}^{l_{\mathrm{E}}} \in \mathbb{R}^{c_{l_{\mathrm{E}}}}$, and $*_{\mathcal{G}}$ depict the Chebyshev coefficients, bias, graph convolution operator. The role of the added bias and ReLU activation function is to improve the model's non-linearity ability [24].

The encoding block is followed by a latent layer, which is responsible for holding the representations of the compressed data throughout the encoding process. Thus, the presence of the latent layer enhances the features' learning process and helps in learning simpler data representations. The graph decoder block is placed after the latent layer. The role of the decoder $\boldsymbol{D}$ is decompressing the data and reconstructing the input. The graph decoder consists of $\mathcal{L}_{\mathrm{D}}$ hidden decoding graph Chebyshev convolution layers with $c_{l_{\mathrm{D}}}$ channels. Each graph decoding layer l_{D} has the input and output of $\boldsymbol{X}^{l_{\mathrm{D}}-1} \in \mathbb{R}^{n\times c_{l_{\mathrm{D}}-1}}$ and $\boldsymbol{X}^{l_{\mathrm{D}}} \in \mathbb{R}^{n\times c_{l_{\mathrm{D}}}}$, respectively. Finally, $\tilde{\boldsymbol{X}}$ denotes the reconstructed input by the graph decoder.

3.2 Training and Testing the GAE Model

Our GAE model is trained on benign samples and tested on both, benign and malicious samples. Specifically, it recognizes malicious samples of under-attack operation based on the presented dissimilarity from the learned normal patterns during training. Since the model is familiar with the patterns of normal operation during training, the dissimilarity is expected to be small during testing. This means that under-attack operation samples are expected to present higher dissimilarity during testing. Marking samples during testing is carried out based on a reconstruction error ζ of the data regeneration procedure. The graph encoder and decoder are denoted as $\boldsymbol{E} = f_{\Phi}(\boldsymbol{X})$ and $\boldsymbol{D} = g_{\Phi}(\boldsymbol{X})$, respectively, where Φ depicts the GAE model parameters, which are expressed as follows

$$\min_{\Phi} C(\boldsymbol{X}, g_{\Phi}(f_{\Phi}(\boldsymbol{X}))), \qquad \boldsymbol{X} \in \boldsymbol{X}_{\text{TR}}. \tag{5}$$

In (5), $C(\boldsymbol{X}, g_{\Phi}(f_{\Phi}(\boldsymbol{X})))$ represents a mean squared error (MSE) cost function that imposes a penalty on $g_{\Phi}(f_{\Phi}(\boldsymbol{X}))$ for the presented dissimilarity from $\boldsymbol{X}$. In other words, (5) estimates the MSE between the original input $\boldsymbol{X}$ and the reconstructed output $\tilde{\boldsymbol{X}}$. The proposed GAE model is trained with the goal of identifying parameters Φ with the aim of optimizing (5). Using the iterative gradient descent approach, the minimization of (5) is accomplished where we divide the training samples $\boldsymbol{X} \in \boldsymbol{X}_{\text{TR}}$ into small batches. Following (5), the value of ζ, which indicates the level of the model's familiarity against $\boldsymbol{X} \in \boldsymbol{X}_{\text{TST}}$, is anticipated to be small and large for benign and malicious samples, respectively. When the value of ζ becomes higher than a threshold value ψ, a malicious sample $\boldsymbol{x}_{\text{m}}$ reflecting an attack is flagged with $y = 1$, otherwise, the sample is considered benign $\boldsymbol{x}_{\text{b}}$ with a $y = 0$ label.

4 Experimental Results

This section assesses the performance of the GAE model compared to several benchmark detectors. Also, we present the used hyperparameters for each model that are selected based on a grid-search selection process. We then analyze the performance of the examined detectors.

4.1 Benchmark Detectors

For an exhaustive analysis, we include multiple data-driven benchmark detectors with shallow, deep, and graph models that are either supervised (trained on benign samples only and tested on benign and attack samples) or unsupervised (trained and tested on benign and attack samples). The adopted shallow models are listed next. The unsupervised auto-regressive integrated moving average (ARIMA) model is trained to predict future data patterns [25]. The supervised SVM model classifies samples using a hyperplane that separates both sample types [5]. The adopted deep models include the FNN, which is a supervised

model that employs feedforward layers that are fully-connected to classify samples [10]. The supervised RNN model exploits temporal correlations via utilizing recurrent cells [26]. The supervised CNN model performs the convolution operation to classify samples [11]. We also adopt a classical stacked autoencoder (SAE), which is an unsupervised model that identifies samples using a reconstruction process using fully-connected feedforward layers [22] without employing a graph convolution operation. Finally, we adopt a graph-based detector, which is a supervised CGNN model that utilizes vertices and edges for modeling the spatial aspects of the data [4].

4.2 Hyperparameter Selection

To select the most suitable hyperparameters for each of the adopted models, we utilize a grid-search selection process that is carried out on multiple stages. The best hyperparameter option is picked from a pool of options according to the offered DR calculated against $\boldsymbol{X}_{\text{VA}}$ by that value. The selected hyperparameters are listed next. ARIMA uses 1 and 0 as the differencing and moving averages, respectively. SVM uses scale and sigmoid as the kernel and gamma values, respectively. FNN employs 4 layers, 32 units, Adamax optimizer, and ELU activation without dropout. RNN employs 3 layers, 16 units, Adam optimizer, and ReLU activation with a dropout rate of 0.2. CNN employs 4 layers, 32 units, 5 neighborhood order, Adam optimizer, and ReLU activation. SAE employs 3 encoding layers with $(32, 16, 8)$ units, 3 decoding layers with $(8, 16, 32)$ units, Adam optimizer and Simoid activation without dropout. CGNN employs 4 layers, 32 units, 3 neighborhood order, Adam optimizer, and ReLU activation. The proposed GAE-model employs 3 encoding layers with $(32, 16, 8)$ units, 3 decoding layers with $(8, 16, 32)$ units, 4 neighborhood order, Rmsprop optimizer, and ReLU activation.

4.3 Evaluation Metrics

To evaluate the models, we adopt the following assessment metrics. First, DR = TP/(TP + FN) reflects the model's ability to correctly mark malicious samples, where TP and FN denote true positive and false negative, respectively. Second, false alarm rate (FAR) = FP/(FP + TN) reflects the quantity of benign samples that the model incorrectly marks as malicious, where FP and TN depict false positive and true negative, respectively. Third, ACC = (TP + TN)/(TP + TN + FP +FN) reflects the model's ability to mark both sample types.

4.4 Detection Performance

Tables 1 and 2 present the results of the investigated detectors. Table 1 reports the evaluation results when the supervised models (SVM, FNN, RNN, CNN, and CGNN) encounter seen attacks and when unsupervised detectors (ARIMA, AE, and GAE) encounter unseen (new) attacks. The GAE model demonstrates

superior DR by 33.1−36.3%, 17.8−29.2%, and 2.5% compared to the graph, deep, and shallow ML-based detectors, respectively. The reason behind the superior performance of the proposed detector is that it learns the graph representations and captures spatial aspects of normal operation of the power system without the need of being trained on malicious samples. Hence, unlike existing supervised detectors, the offered attack identification results by the GAE model are not limited to a set of FDIAs, which highlights its superiority.

Table 1. Evaluation Results Against FDIAs (%).

Detector	**DR**	**FAR**	**ACC**
ARIMA	53.9	53.6	53.1
SVM	57.1	45.8	56.1
FNN	61	39.3	60
RNN	66.6	32.8	65.4
CNN	71.6	25.9	71.3
SAE	72.4	24.7	72
CGNN	84.4	13.6	83.5
GAE	**90.2**	**9.3**	**89.8**

In real-life, attackers might launch unseen new FDIA types. These attack types might not be present during training stage of the model. To reflect such a scenario, in Table 2, we present the evaluation results of supervised models when encountering new unseen attacks compared to the unsupervised GAE model. The detection performances of such detectors significantly degrade when they are tested against unseen FDIA types. Specifically, the DRs of shallow, deep, and graph-based detectors deteriorate by 13.9%, $10.1 - 12.7\%$, and 9.8% compared to encountering seen attacks. This means that the proposed GAE-based detector provides an increase of $47 - 15.6\%$ in DR compared to supervised benchmark detectors against unseen FDIA types. The reason behind such enhancements is that the proposed GAE-based detector offers unsupervised training that learns the graph representations of benign behavior through the encoder-decoder benign data reconstruction process, which increases its robustness against unseen FDIA types.

5 Conclusions

This work proposed adopting an artificial intelligence-based GAE unsupervised anomaly detector that provides three major advantages compared to existing FDIA detectors. First, it employs Chebyshev graph convolution operation. Thus, it captures the grid's spatial characteristics. Second, it offers an unsupervised learning strategy using an autoencoder that relies only on benign samples of

Table 2. Evaluation Results Against Unseen FDIAs (%).

Detector	DR	FAR	ACC
SVM	43.2	59.9	42.2
FNN	48.3	52.1	47.9
RNN	54.8	44.6	53.6
CNN	61.5	36.5	61.7
CGNN	74.6	24.3	73.7
GAE	**90.2**	**9.3**	**89.8**

normal operation for training and hence offers robust detection of unobserved types of FDIAs that do not take part of the training process. Third, due to its structure that is equipped with stacked graph layers and its unsupervised learning nature, employing the proposed detector leads to a superior detection performance as opposed to benchmark detectors by $16 - 47\%$ in DR against unseen FDIA types.

References

1. D. An *et al.*, "Data integrity attack in dynamic state estimation of smart grid: Attack model and countermeasures," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1631–1644, Jul. 2022.
2. Z. Zhang *et al.*, "Cyber-physical coordinated risk mitigation in smart grids based on attack-defense game," *IEEE Transactions on Power Systems*, vol. 37, no. 1, pp. 530–542, Jan. 2022.
3. K. Huang *et al.*, "False data injection attacks detection in smart grid: A structural sparse matrix separation method," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2545–2558, Jul. 2021.
4. 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.
5. 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.
6. 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.
7. 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.
8. A. S. Musleh *et al.*, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2218–2234, May 2020.
9. 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.

10. 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.
11. 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.
12. 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.
13. 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.
14. 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.
15. 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.
16. 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.
17. A. Takiddin *et al.*, "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.
18. 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.
19. A. Takiddin *et al.*, "A graph neural network multi-task learning-based approach for detection and localization of cyberattacks in smart grids," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*. Rhodes Island, Greece, 4–10 Jun. 2023, pp. 1–5.
20. M. Hasnat *et al.*, "Detection and locating cyber and physical stresses in smart grids using graph signal processing," *arXiv:2006.06095*, Jun. 2020.
21. 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.
22. 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.
23. L. Wu *et al.*, *Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer, Jan. 2022.
24. A. Takiddin *et al.*, "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.
25. 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.
26. A. Takiddin *et al.*, "Data-driven detection of stealth cyber-attacks in dc microgrids," *IEEE Systems Journal*, vol. 16, no. 4, pp. 6097–6106, Dec. 2022.