A GRAPH NEURAL NETWORK MULTI-TASK LEARNING-BASED APPROACH FOR DETECTION AND LOCALIZATION OF CYBERATTACKS IN SMART GRIDS

Abdulrahman Takiddin[1], *Rachad Atat*[2], *Muhammad Ismail*[3], *Katherine Davis*[1], *Erchin Serpedin*[1*]

[1] Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX, USA
[2] Electrical & Computer Engineering Program, Texas A&M University at Qatar, Doha, Qatar
[3] Department of Computer Science, Tennessee Technological University, Cookeville, TN, USA

ABSTRACT

False data injection attacks (FDIAs) on smart power grids' measurement data present a threat to system stability. When malicious entities launch cyberattacks to manipulate the measurement data, different grid components will be affected, which leads to failures. For effective attack mitigation, two tasks are required: determining the status of the system (normal operation/under attack) and localizing the attacked bus/power substation. Existing mitigation techniques carry out these tasks separately and offer limited detection performance. In this paper, we propose a multi-task learning-based approach that performs both tasks simultaneously using a graph neural network (GNN) with stacked convolutional Chebyshev graph layers. Our results show that the proposed model presents superior system status identification and attack localization abilities with detection rates of $98.5-100\%$ and $99-100\%$, respectively, presenting improvements of $5-30\%$ compared to benchmarks.

***Index Terms*—** Multi-task learning, localization, false data injection attacks, cyberattacks, smart grids.

1. INTRODUCTION

Information security and forensics are critical when exchanging measurement data among smart grid components. Smart grids rely on measurement data to ensure proper supply/demand management and system stability [1]. The cyber-physical nature of smart power grids makes them vulnerable to false data injection attacks (FDIAs) where malicious entities manipulate power system measurement data (e.g., sensor data). This leads to making wrong decisions because of the altered data, which may result in system instability. Such attacks may also bypass traditional bad data detectors [2]. Thus, research works proposed alternative detection strategies.

1.1. Related Works

Existing detectors perform one of two tasks, detection or localization. For system status identification (detection), decision tree [3] and support vector machine (SVM) [4] detectors offered 82% and 88% F1-scores, respectively. Random forests presented a detection rate (DR) of 93% [5]. Multi-layer perceptron (MLP) [6] and convolutional neural network (CNN) [7] detectors offered accuracy (ACC) of 90% and 93%, respectively. Recurrent neural network (RNN) [8] and auto-encoder (AE) [9] detectors exhibited DRs of 96%. Convolutional graph neural network (CGNN) [10] and graph auto-encoder (GAE) [11] detectors offered DRs of $83-92\%$ in IEEE 14 and 118-bus systems. For attack localization, a graph signal processing-based localization approach offered 77% in DR [12]. Also, a GNN-based localization approach reported 93% in F1-Score [13]. However, such detectors are inefficient since the system status and attack localization tasks are performed separately, and hence, offer limited detection performance as decisions are made based on features learned from only one task.

1.2. Contributions

We overcome these limitations by proposing a multi-task learning-based detector that offers the following features:

- It performs two tasks: graph classification to determine the system status (under attack/normal operation) and node classification to localize the attack (the attacked node). This is performed efficiently using a three-stage GNN with joint, task-specific, and fusing layers.
- It captures the complex patterns of measurement data and spatial aspects of power grids using convolutional Chebyshev graph layers. We examine its robustness against FDIAs on IEEE 14, 39, and 118-bus systems.
- It makes decisions based on features learned throughout both tasks. Hence, it offers enhanced system status identification and attack localization abilities with DRs of $98.5-100\%$ and $99-100\%$, respectively, which outperforms system status and localization task-specific benchmarks by $10-30\%$ and $5-27\%$, respectively.

This paper is organized as follows. Section 2 describes the data preparation along with the FDIA functions. Section 3 in-

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

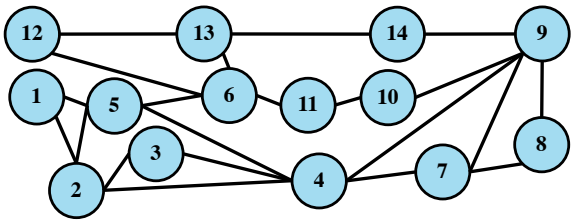

Fig. 1. A graph representation of the IEEE 14-bus system.

troduces the proposed model. Section 4 presents the detection performance results. Conclusions are presented in Section 5.

2. DATA PREPARATION

To train and test the investigated models, we use three IEEE bus systems, namely, IEEE 14, 39, and 118-bus systems. Also, we adopt three FDIA functions to produce malicious samples that mimic the power grid operation under attack.

2.1. Power System Modeling

To properly model a power system, spatial and temporal aspects need to be captured. A power system can be modeled using an undirected graph where buses and power lines are represented by nodes and edges, respectively. To model the spatial aspect of different power grids, we adopt IEEE 14, 39, and 118-bus systems. For example, Fig. 1 illustrates the IEEE 14-bus system represented using an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$, where buses, power lines, and line admittance are represented by nodes $\mathcal{V}$, edges $\mathcal{E}$, and weighted adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$, respectively. If buses i and j are connected, a weight W_{ij} is associated to edge $e = (i, j)$ based on the line admittance.

The temporal aspect refers to the electric power injections and flows. $\mathcal{V}$ and $\mathcal{E}$ are associated with features. Power flow analysis using Newton's method is carried out using MATLAB MATPOWER toolbox [14] to determine the real and reactive power flows in the system. The node features include active power P_i in megawatts of real power demand and reactive power demand Q_i in megavolt amperes of reactive power.

2.2. Benign Samples

The aforementioned features include measurement data denoting benign samples $\boldsymbol{x}_{\text{b}}(t, i)$ at bus i and timestamp t during normal operating of the power system. We have 96 daily power dynamics timestamps over six months, resulting in around $17,000$ timestamps.

2.3. Malicious Samples

We adopt three FDIA functions that are used to mimic the power system's operation under attack. The attacks are applied in a stealthy manner as they bypass traditional bad data detectors by presenting similar patterns to benign data [15]. The malicious sample $\boldsymbol{x}_{\text{m}}(t, i)$ at bus i and timestamp t is generated by altering $\boldsymbol{x}_{\text{b}}(t, i)$.

The direct attack generates $\boldsymbol{x}_{\text{m}}$ by randomly injecting perturbations bounded in magnitude by a scaling factor α ($|\alpha| \leq 0.05$) into a benign sample such that

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

The replay attack involves generating $\boldsymbol{x}_{\text{m}}$ using a false repetition of a reading from a previous timestamp $t - 1$ to replace the reading of a current timestamp t such that

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

The general attack [16] generates $\boldsymbol{x}_{\text{m}}$ using a range of true measurement values such that

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

where β and $0 \leq \gamma \leq 1$ are binary and uniform random variables, respectively.

2.4. Dataset Splitting

The investigated supervised models are trained and tested on $\boldsymbol{x}_{\text{b}}$ and $\boldsymbol{x}_{\text{m}}$, whereas the unsupervised models are trained only on $\boldsymbol{x}_{\text{b}}$ and tested on $\boldsymbol{x}_{\text{b}}$ and $\boldsymbol{x}_{\text{m}}$. To produce unbiased comparison results, we (i) use equal number of benign and malicious samples and (ii) report the average of multiple carried out experiments. The train $\boldsymbol{X}_{\text{TR}}$, validate $\boldsymbol{X}_{\text{VA}}$, and test $\boldsymbol{X}_{\text{TS}}$ sets present equal numbers of benign and malicious samples. Samples are split into the three sets, where 80%, 10%, and 10% of samples present $\boldsymbol{X}_{\text{TR}}$, $\boldsymbol{X}_{\text{TS}}$, and $\boldsymbol{X}_{\text{VA}}$, respectively.

3. MULTI-TASK LEARNING DETECTION

The proposed multi-task learning GNN-based model performs two tasks: graph classification, which determines the system status (i.e., under attack or normal operation), and node classification, which localizes the attack (i.e., identifies the attacked node).

3.1. Model Architecture

As shown in Fig. 2, the proposed model is divided into three stages. The first stage is inspired from [17, 18] and presents the joint graph layers, which are used to extract preliminary features from the data that are needed for the two successive tasks. The second stage consists of task-specific graph layers that are designated to capture relevant features for a specific task. The third stage presents the final decision of the two tasks according to the learned features in the previous stages, which boosts the detection performance. The rationale behind such a structure is to calculate the initial weights and parameters (shared parameters) once, which are then transmitted to the next stage for further processing.

The proposed GNN-based detector has an input layer that takes as inputs, graphs with $[P_i, Q_i] \in \mathbb{R}^{n\times 2}$ measurement samples that are either $\boldsymbol{X}_{\text{b}}$ or $\boldsymbol{X}_{\text{m}}$. The input layer is followed by hidden joint convolutional Chebyshev graph layers L_j for joint optimization [19]. Each layer l_j has c_{l_j} channels and an input and output of $X^{l_j-1} \in \mathbb{R}^{n\times c_{l_j-1}}$ and $X^{l_j} \in \mathbb{R}^{n\times c_{l_j}}$, respectively. Each layer l_j is responsible for capturing the spatial aspects from the graph [20] through the operations of graph convolution, bias insertion, and activation function (i.e., ReLU). ReLU produces the output tensor X^{l_j}:

$$X^{l_j} = \text{ReLU}(\boldsymbol{\mu}^{l_j} *_{\mathcal{G}} X^{l_j-1} + \boldsymbol{b}^{l_j}), \tag{4}$$

where $\boldsymbol{\mu}^{l_j} \in \mathbb{R}^{K\times c_{l_j-1}\times c_{l_j}}$ denotes the Chebyshev coefficients, $\boldsymbol{b}^{l_j} \in \mathbb{R}^{c_{l_j}}$ stands for the bias, and $*_{\mathcal{G}}$ depicts the graph convolution operator.

The joint layers L_j are followed by the task-specific convolutional Chebyshev graph layers L_s (i.e., L_{s1} and L_{s2} for the first and second tasks, respectively). Each task-specific layer l_s captures more relevant features for the specific task it is assigned to. For each task, the first layer l_s takes the output $X^{L_j} \in \mathbb{R}^{n\times c_{L_j}}$ of the last joint layer as an input and produces $X^{l_s} \in \mathbb{R}^{n\times c_{l_s}}$ as an output. The rest of the task-specific layers take $X^{l_s-1} \in \mathbb{R}^{n\times c_{l_s-1}}$ as inputs, then output $X^{l_s} \in \mathbb{R}^{n\times c_{l_s}}$. Both sets of L_s layers are followed by a dense layer that determines the probability of an attack presence (i.e., attack sample in the graph level for the first task and attacked node for the second task). The dense layers take the output of the last task-specific graph layer ($X^{L_{s1}} \in \mathbb{R}^{n\times c_{L_{s1}}}$ and $X^{L_{s2}} \in \mathbb{R}^{n\times c_{L_{s2}}}$ for the first and second tasks, respectively) as an input. The dense layers produce the outcome of a sigmoid function expressed as

$$\text{sigmoid}(W^{L_s} X^{L_s} + \boldsymbol{b}^{L_s}), \tag{5}$$

where $W^{L_s} \in \mathbb{R}^{n\times c_{L_s}}$ depicts the task-specific feature weights and $\boldsymbol{b}^{L_s} \in \mathbb{R}$ denotes the task-specific bias. Employing bias and the activation functions (ReLU and sigmoid) leads to an improved model non-linearity [21]. The task-specific decision (i.e., probability of an under attack status and a malicious node for the first and second tasks, respectively) is then presented in the task-specific output layer. The last layer (final output) presents a fusing layer that provides an improved decision on both tasks based on the information obtained during the previous steps.

3.2. Model Training

The training of the proposed multi-task GNN model and the calculations of the model parameters are carried out using the cross-entropy loss function:

$$C(y_p, \Theta) = \frac{-1}{N} \sum_{\boldsymbol{X}_{\text{TR}}} \{y \log(y_p) + (1-y)\log(1-y_p)\}, \tag{6}$$

where Θ denotes the trainable parameters ($\boldsymbol{\mu}^l$, $\boldsymbol{b}^l$, W^L, and $\boldsymbol{b}^L$) $\forall\, l_{(\cdot)}$, and N depicts the number of training samples $\boldsymbol{X}_{\text{TR}}$.

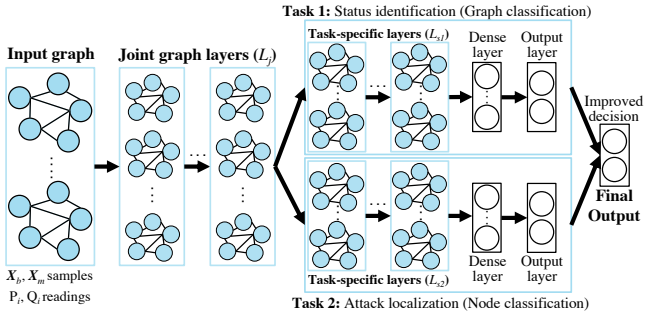

Fig. 2. Illustration of the proposed multi-task GNN model.

y and y_p represent the actual and predicted labels, respectively. Training is performed end-to-end sequentially, based on the final output, and not the task specific outputs [22]. This boosts the detection performance compared to the output of the task-specific layers or training via separate structures for each task. The model training is based on an iterative gradient descent-based optimization, where we split $\boldsymbol{X}_{\text{TR}}$ into equally-sized mini batches that are fed to the model in 128 epochs.

4. EXPERIMENTAL RESULTS

4.1. Benchmark Detectors

We adopt benchmark detectors that offer different characteristics, including structure (shallow/deep/graph), training nature (unsupervised/ supervised), and mechanism (static/dynamic). Autoregressive integrated moving average (ARIMA) is an unsupervised dynamic model trained on benign data to predict data patterns [23]. SVM is a supervised static model trained on both sample types and classifies samples via an equal-distance hyperplane between them [4]. MLP is a supervised static model that utilizes fully-connected feedforward layers to capture features from data to classify samples [6]. RNN is a supervised dynamic model that uses recurrent cells to capture temporal aspects [8]. CNN is a supervised model that classifies samples using convolution [7]. CGNN [10] and GAE [11] are supervised and unsupervised graph models, respectively. They model spatial aspects using nodes and edges.

4.2. Optimal Hyperparameters

We adopt a multi-stage grid-search hyperparameter selection procedure where each hyperparameter is selected at a stage [24]. The optimal value is selected from a predefined list of possible values based on the DR against $\boldsymbol{X}_{\text{VA}}$ associated with that value. For ARIMA, the differencing and moving averages are 1 and 0, respectively. For SVM, the kernel and gamma are scale and sigmoid, respectively. For MLP, there are 4 layers with 32 units, Adamax optimizer, no dropout rate, and ELU activation. For RNN, there are 3 layers with 16 units, Adam optimizer, dropout rate of 0.2, and ReLU activation. For CNN, there are 4 layers with 32 units, neighborhood order of 5, Rmsprop optimizer, and ReLU activation. For the

graph models, there are 4 layers (in each stage for the proposed model) with 32 units, neighborhood order of 3, Adam optimizer, and ReLU activation. The adopted grid-search hyperparameter selection procedure enhances the detection performance by 4–6% compared to the default hyperparameters.

4.3. Evaluation Metrics

We consider three performance evaluation metrics: DR = TP/(TP + FN) indicates the amount of truly detected malicious samples, where TP and FN depict true positive and false negative, respectively; False alarm rate (FAR) = FP/(FP+TN) indicates the amount of benign samples falsely detected as malicious, where FP and TN denote false positive and true negative, respectively; ACC = (TP+TN)/(TP+TN+FP+FN) indicates how well samples are classified.

4.4. Simulation Results

According to Tables 1 and 2, for the system status classification task, the proposed multi-task GNN detector offers superior DR by $23.2-30.1\%$, $15-21.2\%$, and $9.4-11.5\%$ compared to shallow, deep, and graph-based benchmarks, respectively. For the attack localization task, the proposed multi-task GNN detector offers superior DR by $20.2-26.6\%$, $9.6-17.3\%$, and $4.1-5.8\%$ compared to shallow, deep, and graph-based benchmarks, respectively. Such enhancements are because the spatial aspects are learned using a multi-stage GNN model with stacked convolutional Chebyshev graph layers. The detector captures general details about the data using the joint layers. Then, it captures task-specific features in the task-specific stage. It then makes final decisions about both tasks using the learned features in the previous stages. Larger systems (118-bus) present slightly better DR by $0.5-3.4\%$ compared to smaller ones (14 and 39-bus) since they offer more data to detectors to capture distinctive features. Detection performance gradually improves by 6–7.5% as we train on all samples compared to $50\%-75\%$ of available samples.

5. CONCLUSIONS

This paper proposed implementing a GNN-based multi-task learning scheme by smart grid system operators to efficiently classify the system status and effectively localize FDIAs in smart power grids. The proposed structure, consisting of a three-stage GNN (joint, task-specific graph layers, and an output fusing layer), helped in boosting the detection performance since the final decision is based on the outputs of the two tasks (graph and node classification). The proposed detector captured spatial aspects of the power system using convolutional Chebyshev graph layers. It yielded system status detection and attack localization DRs of $98.5-100\%$ and $99-100\%$, respectively, offering improvements of $10-30\%$ and $5-27\%$ compared to task-specific system status identification and attack localization benchmarks, respectively. Our future work will focus on classifying multiple attack types to design mitigation strategies accordingly.

Table 1. Detection performance of system status (%).

Detector	Metric	IEEE System Size		
		14-bus	39-bus	118-bus
ARIMA [23]	**DR**	68.4	70.2	73.5
	FAR	33.2	30.9	28.7
	ACC	67.8	69.8	72.8
SVM [4]	**DR**	72.1	74.3	76.8
	FAR	30.2	28.3	26.4
	ACC	71.6	73.8	76.0
MLP [6]	**DR**	77.3	79.2	81.3
	FAR	24.7	23.0	21.3
	ACC	76.9	78.5	80.4
RNN [8]	**DR**	80.6	82.4	84.5
	FAR	21.1	19.8	18.4
	ACC	80.5	81.9	83.8
CNN [7]	**DR**	82.5	83.7	85.5
	FAR	19.3	18.1	16.9
	ACC	81.8	83.1	84.3
AE [9]	**DR**	84.3	85.4	86.4
	FAR	17.9	16.8	15.7
	ACC	83.9	85.1	86.1
CGNN [10]	**DR**	87.2	87.8	88.5
	FAR	15.7	15.2	14.6
	ACC	86.9	87.5	88.1
GAE [11]	**DR**	89.1	89.6	90.0
	FAR	10.4	9.9	9.4
	ACC	88.6	89.0	89.6
Proposed multi-task GNN	**DR**	**98.5**	**99.3**	**100.0**
	FAR	**0.92**	**0.54**	**0.12**
	ACC	**98.1**	**98.9**	**99.8**

Table 2. Detection performance of attack localization (%).

Detector	Metric	IEEE System Size		
		14-bus	39-bus	118-bus
ARIMA [23]	**DR**	72.6	73.9	75.5
	FAR	28.5	27.2	26.0
	ACC	71.9	73.1	74.4
SVM [4]	**DR**	77.5	78.8	79.8
	FAR	24.6	23.5	22.4
	ACC	76.9	78.0	79.2
MLP [6]	**DR**	81.9	83.1	84.4
	FAR	20.1	19.1	18.0
	ACC	76.9	78.5	80.4
RNN [8]	**DR**	85.6	86.7	87.9
	FAR	17.6	16.4	17.6
	ACC	84.9	86.0	87.2
CNN [7]	**DR**	88.3	89.3	90.4
	FAR	16.8	15.6	14.3
	ACC	87.6	88.4	89.7
AE [9]	**DR**	90.7	91.7	92.6
	FAR	13.8	12.7	11.8
	ACC	89.7	90.6	91.4
CGNN [10]	**DR**	93.4	94.5	95.7
	FAR	11.8	10.7	9.6
	ACC	92.7	93.7	94.8
GAE [11]	**DR**	93.6	94.7	95.9
	FAR	10.5	9.6	8.5
	ACC	94.1	94.8	95.4
Proposed multi-task GNN	**DR**	**99.2**	**99.7**	**100.0**
	FAR	**0.72**	**0.42**	**0.08**
	ACC	**98.7**	**99.2**	**99.7**

6. REFERENCES

[1] Zhimei 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.

[2] Xuefei 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 Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1957–1967, Mar. 2022.

[3] Xiao 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)*. 15–17 Oct. 2020, pp. 133–136, Manchester, United Kingdom.

[4] Mohammad Esmalifalak et al., "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Sys. J*, vol. 11, no. 3, pp. 1644–1652, Sept. 2017.

[5] Defu Wang et al., "Detection of power grid disturbances and cyber-attacks based on machine learning," *Journal of Info. Sec. and Apps.*, vol. 46, pp. 42–52, Jun. 2019.

[6] Erik 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)*. 14–16 Dec. 2017, Las Vegas, NV, USA.

[7] Shuoyao 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.

[8] Yufeng 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.

[9] Ying 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.

[10] Osman 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.

[11] Abdulrahman 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*, pp. 1–13, 2023.

[12] Ezzeldin Shereen et al., "Detection and localization of pmu time synchronization attacks via graph signal processing," *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 3241–3254, Jul. 2022.

[13] Osman Boyaci et al., "Infinite impulse response graph neural networks for cyberattack localization in smart grids," *arXiv preprint arXiv:2206.12527*, Jun. 2022.

[14] "R. D. Zimmerman and C. E. Murillo-sanchez, "MATPOWER." [online] available: https://matpower.org," .

[15] Abdulrahman Takiddin et al., "Detection of electricity theft false data injection attacks in smart grids," in *30th European Signal Processing Conference (EUSIPCO))*. 29 Aug.–2 Sept. 2022, pp. 1541–1545, Belgrade, Serbia.

[16] Md Abul Hasnat et al., "Detection and locating cyber and physical stresses in smart grids using graph signal processing," *arXiv:2006.06095*, 2020.

[17] Yu Xie et al., "A multi-task representation learning architecture for enhanced graph classification," *Frontiers in Neuroscience*, vol. 13, pp. 1395, Jan. 2020.

[18] Hong Huang et al., "Multitask representation learning with multiview graph convolutional networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 983–995, Mar. 2022.

[19] Jae-Jin Jeon and Eesung Kim, "Multitask learning and joint optimization for transformer-rnn-transducer speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 21–24 June 2021, pp. 6793–6797, Toronto, ON, Canada.

[20] Jiayu Li et al., "Adversparse: An adversarial attack framework for deep spatial-temporal graph neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 23–27 May 2022, pp. 5857–5861, Singapore, Singapore.

[21] Luana Ruiz et al., "Invariance-preserving localized activation functions for graph neural networks," *IEEE Trans. on Signal Proc.*, vol. 68, pp. 127–141, Nov. 2020.

[22] Abdulrahman 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.

[23] William H V. Badrinath *et al.*, "ARIMA-Based modeling and validation of consumption readings in power grids," in *Critical Information Infrastructures Security*. 2016, pp. 199–210, Springer International Publishing.

[24] Abdulrahman 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.