Graph Neural Networks Approach for Cyberattack Detection in Smart Grids

Xiao Yue
Department of Computer Science and Engineering
Oakland University
Rochester, MI, USA
xiaoyue@oakland.edu

Guangzhi Qu
Department of Computer Science and Engineering
Oakland University
Rochester, MI, USA
gqu@oakland.edu

2024 Cyber Awareness and Research Symposium (CARS) | 979-8-3503-8641-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/CARS61786.2024.10778828

***Abstract*—Due to the development and applications of cyber-physical systems (CPS) and industrial Internet of Things (IoT), smart grids have significantly evolved the traditional power grids. Their increased utilization also exposes the vulnerabilities in smart grids, such as cyberattacks. Because of the impressive performance of machine learning models across various application domains, they are increasingly being applied for detecting cyberattacks in smart grids. However, most existing machine learning-based methods do not utilize the grids' topology information leading to inefficiency in detecting network-level attacks in the smart grids. In this paper, we propose a new approach to cyberattack detection by utilizing the graph topology of smart grids. We define the network topology of smart grids as graphs to exploit spatial features of smart grids' topological information, enabling detection of cyberattacks using a graph neural network (GNN)-based approach. Experimental results indicate that GNNs outperform other traditional machine learning models, highlighting the potential of exploiting GNNs on cybersecurity-related works for smart grids.**

***Index Terms*—Smart Grids, Cybersecurity in CPS, Graph Neural Networks**

I. Introduction

The emergence of smart grids has significantly evolved the traditional power grids, due to the development and applications of cyber-physical systems (CPS) and industrial Internet of Things (IoT) [1]–[3]. These digital communication technologies improve the efficiency, reliability, and sustainability of electricity production and distribution on smart grids. However, the integration of digital and network technologies introduces vulnerabilities in smart grids that can be exploited by malicious cyberattacks. Smart grids contain various components such as smart meters, automated control systems, and communication networks that are vulnerable to both traditional cyberattacks and novel attacks stemming from the smart technology advancements [4]. In general, smart grids contain two transmission lines, a power transmission line and a data transmission line, which are vulnerable to cyberattacks. Cyberattacks on smart grids aim to damage grid systems for different purposes such as cyber warfare, economic benefit, and terrorism. A common cyberattack on smart grids is false data injection, which sabotages the grid's communication network by exploiting various vulnerabilities in smart grids to insert false or manipulated data, compromising the integrity and reliability of the grids. For example, hackers can modify electric power usage by injecting false data into smart meters, leading to financial losses in the electricity markets. In order to prevent cyberattacks from causing both physical and virtual damage to smart grids, valuable countermeasures have been proposed in various studies [5]–[7].

With the development of machine learning techniques, machine learning models have been widely deployed and exploited in both academical and industrial fields in recent years. They present promising performance in various application domains, such as computer vision [8], natural language processing [9], and graph learning [10]. Machine learning techniques are also extensively adopted in the context of cybersecurity in smart grids. For example, Kwon et al. [11] proposed exploiting Bidirectional Recurrent Neural Networks (BRNN) to detect attacks in power systems such, as false data injection or malware. Machine learning techniques related to cyberattack detections, power quality disruption, and dynamic security assessment for cybersecurity and stability of power systems are discussed in [12]. Because of the excellent ability of machine learning models to learn complex patterns from data and make accurate predictions, they outperform traditional methods in classification tasks. Consequently, the detection of cyberattacks by exploiting machine learning models can be treated as classification tasks of normal data and tampered data. Even though topology of smart grids can be modeled as graphs [13], [14], most of the existing machine learning models for cyberattack detection in smart grids do not utilize topology information of grids leading to inefficiency of network level attacks detection in the grids. Due to the number of smart devices in a smart grid, it increases the burden on the system to monitor all devices individually. In this paper, we propose a new approach of detections on cyberattacks by exploiting spatial features of smart grids' topological information. The propagation-aggregation strategy, a prevailing strategy adopted by majority of graph neural networks (GNNs), is able to build a comprehensive representation of the entire graph by iteratively updating representations of nodes by aggregating node representations from neighboring nodes. Consequently, we are able to apply detections on the entire smart grid by exploiting graph learning techniques on smart grids. Take false data injection attacks on smart meters as an example, we can define the network topology of smart grids as graphs where

each node represents an electric usage unit such as a device or a house. Data derived from each node is assigned as the feature of that node. We train GNN models, such as a graph convolutional network (GCN) or a graph isomorphism network (GIN), to produce the classification on smart grids graphs, determining if any false data injection attack happened in a smart grid in one shot, instead of identifying attack node by node. To evaluate the proposed approaches, we conducted experiments on two smart grids whose topology can be defined as graphs. GCN, GIN, Multilayer Perceptron (MLP) classifier and Decision Tree Classifier were exploited to produce detection results on simulated data for smart meters. In addition, we also validated the effectiveness of autoencoder for dimension reduction. Experimental results indicate that both GCN and GIN outperform two traditional machine learning models, the MLP classifier and the Decision Tree Classifier.

The rest of this paper is organized as follows. In Section II, we provide an overview of the related works on attack and detection methods for smart grids, as well as GNNs. Approaches that we utilized for detections are presented in section III. Section IV demonstrates our experimental results and comparisons with two other machine learning models. Finally, we draw conclusions briefly in Section V.

II. Related works

A. Attacks on smart grids

Traditional cyberattacks on smart grids can be broadly categorized into the following types based on the target of the attacks:

- Data attacks: attacks on grid data or information that manipulate the data generated within grids and mislead control systems into making incorrect decisions, thereby disrupting normal grid operations. A few well-known types of these attacks include:
  - False Data Injection attacks (FDIAs): attackers inject false data into the grid's control system, which will respond incorrectly. This can disrupt grid operations and potentially lead to a blackout [15].
  - Replay attacks: previously legitimate data is captured and retransmitted to create incorrect actions or responses in the grid system [16].
  - Denial of Service (DoS) attacks: excessive amounts of data are sent over the grid's network that exhausts the control system's response to legitimate data from being processed [17].
  - Privacy attacks: private and sensitive information of users and the grid's infrastructure is accessed without authorization, leading to data breaches [18].
- Firmware/Software manipulation: firmware or software of the grid devices is tampered with or compromised so that attackers can acquire privilege escalation that allows them to manipulate data (altering intercepted data, collecting transmitted data, spoofing data, etc.) and control operations more extensively [17].
- Phishing and Social Engineering attacks: individuals are deceived by trusted entities into granting unauthorized access or disclosing confidential information.
- Ransomware attacks: critical data or control systems are encrypted, causing grid operations to halt. Attackers then demand a ransom to restore access or their operations.

B. Attack detection methods for smart grids

In the literature, traditional attack detection methods for smart grids can be broadly categorized into two types: signature-based detection and anomaly-based detection. We also briefly introduce deep learning-based detection methods in this section.

1) Signature-based Detection: In signature-based detection, predefined patterns or signatures of known attacks are utilized to compare the incoming data to identify threats or attacks. This method can effectively detect a wide range of attacks, including DoS attacks, malware infections, and replay attacks. The signature can include specific sequences of events or thresholds defined on individual features. However, this approach is less effective at detecting novel attacks, since it relies on existing knowledge of attack patterns.

2) Anomaly-based Detection: On the other hand, anomaly detection systems can identify deviations from normal operational behavior that may signal cyberattacks. Statistical methods [19] are employed to model normal behavior and detect deviations in grid operations to alert to expected load changes or voltage fluctuations. Supervised and unsupervised machine learning algorithms, such as Support Vector Machine (SVM) and k-means clustering, are used to differentiate normal and abnormal behaviors. These techniques are used to detect intrusions and false data injection attacks. Furthermore, deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), that can capture complex patterns and temporal dependencies, have been applied to time-series data from smart grids to detect anomalies.

3) Deep learning-based Detection: Machine learning techniques are extensively adopted for cybersecurity in smart grids, as networking technologies and condition monitoring are critical for normal operation. Compared to classic defense methods such as residue-based modeling and blockchain digital ledgers, machine learning models are preferred because of their higher accuracy, lower deployment costs, ability to handle dynamic data, and ease of direct deployment. For example, Kwon et al. [11] proposed detecting attacks such as false data injection or malware by exploiting deep learning models. MENSA [20] was proposed detecting operational anomalies and cyberattacks by combining autoencoders and generative adversarial networks simultaneously. Arshia et al. [21] proposed a hybrid framework to detect and eliminate cyberattacks that disrupt the process of dynamic state estimation by exploiting unsupervised hierarchical clustering. Wu et al. [22] trained a series of feedforward networks by utilizing extreme learning machine models for detection of false data injection attacks. Additionally, the machine learning techniques for detecting

false data injection attacks in smart grids are well-discussed in [23]. Berghout et al. [24] provide a comprehensive study on cybersecurity-related models and further discuss them from a categorical perspective, covering both traditional and advanced machine learning techniques. A deep reinforcement learning-based method is proposed in [25] to detect false data injection attacks with a combined dynamic-static detection mechanism. [26] proposed to utilize GNN detectors for false data injection attacks , and compared them to classical ML-based methods. A scalable and real-time detection mechanism for false data injection attacks by integrating graph topology of the power grid and GNN layers was presented in [27].

C. Graph neural networks

As we mentioned in Section I, machine learning techniques are also extensively adopted in the context of cybersecurity in smart grids. However, most of the existing machine learning models for cyberattack detections in smart grids do not utilize the topology information of grids leading to inefficiency of network level attacks detection in the grids, therefore may increase the burden on the system to monitor all devices individually. To address this issue, as well as spatial features of smart grids' topological information, GNNs are selected to apply graph learning task on smart grids. Due to the property of lacking Euclidean structure in graph data, GNNs have been proposed to apply machine learning techniques to graph data. As a prominent graph neural network variant, GCNs are generally divided into two main categories [28]: spectral-based and spatial-based, according to their algorithmic properties. Bruna et al. [29] proposed to perform convolutional operations on graphs in the Fourier domain by a notable spectrally-based GCN. After that, input smoothing kernels and parametric spectral filters for graph convolutions were proposed in [30]. To overcome the obstacle of low computational efficiency in creating spatial localization for graphs, and inspired by the first-order approximation of spectral graph convolutions, Kipf et al. [10] improved GCNs by exploiting a layer-wise propagation rule. In addition, K-polynomial filters were also proposed in ChebNet [31] to address the limitations. However, spectral-based GCNs suffer from relying on the fixed spectrum of the graph Laplacian, making transferring learned models to another graph challenging.

On the other hand, to bypass expensive convolutional operations in the Fourier domain, which are restricted to a fixed graph structure, spatial-based GCNs apply convolution operations by alternatively aggregating node representations from neighboring nodes. Duvenaud et al. presented a notable convolutional neural network that propagated and aggregated features between neighboring nodes [32]. The PATCHY-SAN [33] model was proposed to locally extract connected regions and order nodes based on structural information. The Deep Graph Convolutional Neural Network [34] was developed to directly work on graph data without any preprocessing, using novel spatial graph convolution layers and SortPooling layers. To overcome limitations of the number of layers in GCNs, DeeperGCN [35] was proposed to extend models to more than 100 layers.

In addition to GCNs, Graph Attention Networks (GATs) [36] exploit attention mechanisms with a convolution-style approach, by leveraging masked self-attentional layers. In order to explore the discriminative power of GNNs based on the propagation-aggregation strategy, Graph Isomorphism Networks [37] were proposed to achieve maximum discriminative power. Xu et al. proposed GraphSAGE [38], which employs inductive representation learning by sampling neighborhoods and aggregating information. GNNs have shown promising performance in various graph-related tasks, such as node classification and graph classification.

III. Approach

A. Graph topology of smart grids

In this work, we use a false data injection attack as the threat model to illustrate cyberattacks on smart grids due to its prevalence. Unlike traditional false data injection attacks that manipulate measurements to disrupt the state estimation and mislead control center operations, attackers focus on devices that are particularly vulnerable to false data injection attacks in this threat model, such as smart meters. By altering data from these devices, attackers can cause substantial economic losses directly. Therefore, we exclude less vulnerable components from the graphs constructed from smart grids to focus on the pertinent components. We represent a smart grid as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}_V)$, where $\mathcal{V}$ represents the set of smart devices and $\mathcal{E}$ is a set of directed edges indicating connectivity between different nodes. We denote $\mathcal{X}_V$ as the set of all node features, such as electric usage data or any other sensitive data from the nodes. Therefore, each node $n_i \in \mathcal{V}$ has a feature vector denoted as $x_i \in \mathcal{X}_V$. In the case of smart meters, we assign the electric usage at each hour in the entire day to the node, creating a 24-dimensional node feature vector. For any intermediate substation connected to houses without any smart meter data, we assign a zero value to each dimension. Note that we assume all edges have no features in this work. However, some GNNs can also handle graphs with edge features, such as EGCN [39] and EGIN [40].

B. Propagation-Aggregation strategy in graph neural networks

A brief introduction of the propagation-aggregation strategy, which is adopted by the majority of GNNs, is presented in this section. Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}_V)$, which is assumed to have no self-loops or isolated nodes. Here, $\mathcal{V}$ represents the set of all nodes, and $\mathcal{E}$ represents the set of all edges in $\mathcal{G}$. The set $\mathcal{X}_V$ contains node features. Each node $n_i \in \mathcal{V}$ has an associated feature vector $x_i \in \mathcal{X}_V$. And $e_{i,j} \in \mathcal{E}$ denotes the edge connecting nodes n_i and n_j. This strategy iteratively updates node representations by aggregating information from neighboring nodes. Let $a_i^{(k)}$ represent the aggregated information from the neighbors of node n_i, and let $h_i^{(k)}$ denote the latent representation of node n_i at the k^{th} GNN layer, with $h_i^{(0)} = x_i$. Let $\mathcal{N}_{(i)}$ represent the set of all neighboring nodes of node n_i. The AGGREGATE

and UPDATE functions are defined by Equations 1 and 2 at the k^{th} GNN layer, respectively.

$$a_i^{(k)} = AGGREGATE^{(k)}(\{\{h_j^{(k-1)} | n_j \in \mathcal{N}_{(i)}\}\}) \quad (1)$$

$$h_i^{(k)} = UPDATE^{(k)}(a_i^{(k)}, h_i^{(k-1)}) \quad (2)$$

GNNs output a multiset of representations of all nodes $\{\{h_i^{(k)} | n_i \in \mathcal{V}\}\}$ in the final layer as node-level embeddings. For graph-level task, a READOUT function is utilized to construct a graph presentation h_G by synthesizing representations of all nodes, as shown in Equation 3.

$$h_G = READOUT(\{\{h_i^{(k)} | n_i \in \mathcal{V}\}\}) \quad (3)$$

C. Weisfeiler-Lehman algorithm and graph isomorphism networks

Similar to the propagation-aggregation methodology being exploited in GNNs, the Weisfeiler-Lehman algorithm [41] tests isomorphism based on an aggregation methodology according to color representations of a node and its neighboring nodes. The Weisfeiler-Lehman algorithm is one of the most well-known methods for isomorphism testing. Consider the same graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}_V)$ given in Section III-B, the color representation of a node n_i at the l^{th} iteration is denoted as $c_i^{(l)}$. Note that $\mathcal{X}_V$ is a multiset in this case, which is a set allowing for multiple instances of each of its elements, noted as $\{\{\cdot\}\}$. Additionally, we assume all node features x_i are discrete. Initially, the node feature x_i is assigned to $c_i^{(0)}$ as the start of iterations. Based on Equation 4, Weisfeiler-Lehman algorithm keeps updating the color representation $c_i^{(l)}$ iteratively until convergence. In Equation 4, *HASH* is an injective mapping that maps $c_i^{(l-1)}$ to $c_i^{(l)}$ and the set of all neighboring nodes of node n_i is denoted as $\mathcal{N}_{(i)}$. Therefore, $c_i^{(l)}$ is unique for each unique pair $(c_i^{(l-1)}, \{\{c_j^{(l-1)} | n_j \in \mathcal{N}_{(i)}\}\})$.

$$c_i^{(l)} = HASH(c_i^{(l-1)}, \{\{c_j^{(l-1)} | n_j \in \mathcal{N}_{(i)}\}\}) \quad (4)$$

The Weisfeiler-Lehman algorithm updates the color representations of nodes $c_i^{(l)}$ by considering each node's previous color representation $c_i^{(l-1)}$ and aggregating the color representations from the multiset of its neighboring nodes $\{\{c_j^{(l-1)} | n_j \in \mathcal{N}_{(i)}\}\}$. The final color representations of all nodes are established when the color representations remain unchanged between iterations. The representation for the entire graph is obtained by synthesizing final color representations of all nodes. The Weisfeiler-Lehman algorithm determines that two graphs are not isomorphic if they have different graph representations. It has proven successful in passing isomorphism tests for most graphs [42], although some failure cases do exist [43]. Inspired by the Weisfeiler-Lehman algorithm, Xu et al. developed the Graph Isomorphism Network [37] to maximize the capabilities of GNNs. They introduced the concept of deep multisets, which parameterizes universal multiset functions with neural networks. GINs are designed to map isomorphic graphs to the same representation while non-isomorphic graphs are mapped to distinct representations. Therefore, the aggregation and update in each EGIN layer are represented in Equation 5, where $h_v^{(k)}$ denotes the latent representation of node n_v at the k^{th} GIN layer.

$$h_v^{(k)} = MLP^{(k)}\{(1 + \epsilon^{(k)}) \cdot h_v^{(k-1)}, + \sum_{u \in \mathcal{N}_{(v)}} h_u^{(k-1)})\} \quad (5)$$

The final embedding of the entire graph is then obtained by a READOUT function, similar to the process described in Section III-B. Therefore, GINs can be utilized to create graph embeddings of graphs constructed from smart grids.

D. Graph convolutional networks

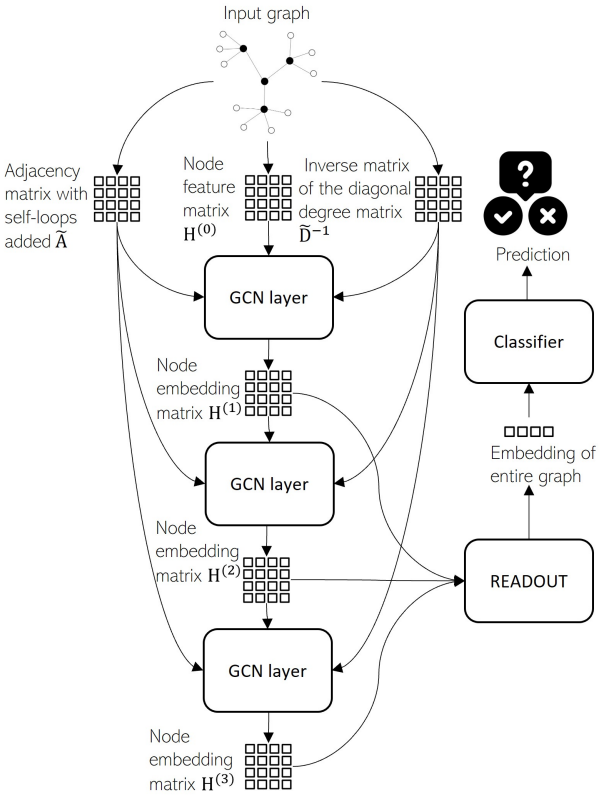

Fig. 1. Structure of a 3-layer GCN

As we construct graphs from smart grids, we can also exploit the GCN layers to aggregate information from neighboring nodes, as shown in Figure 1. The spatial-based GCNs presented here perform convolution operations by alternatively aggregating node representations from neighboring nodes. The basic rule of aggregating node features along the edges in a graph convolutional layer is shown in Equation 6.

$$H^{(l+1)} = \sigma(\widetilde{D}^{-1}\widetilde{A}H^{(l)}W^{(l)}) \quad (6)$$

Where matrix $H^{(l)}$ represents embeddings of all nodes in the l^{th} graph convolution layer, i.e. output from $(l-1)^{th}$ iteration. It can be regarded as a matrix format of all nodes' latent representations $h_i^{(k)}$ in Section III-B or $h_v^{(k)}$ in Section III-C. The node feature matrix is utilized as $H^{(0)}$ for the first iteration. $\widetilde{A} = A + I$ is the adjacency matrix of the graph

with self-loops added, where A denotes the adjacency matrix and I denotes an $n \times n$ size diagonal matrix to represent self loops in this graph. $\widetilde{D}^{-1}$ is an inverse matrix of the diagonal degree matrix $\widetilde{D}$ calculated by formula $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$. In this case, $\widetilde{D}^{-1}$ serves as a normalization constant. $W^{(l)}$ represents a learned weight matrix of the l^{th} GCN layer. The σ is denoted to be a non-linear activation function such as ReLU. By doing so, a new embeddings matrix $H^{(l)}$ is derived by aggregating information from neighboring nodes embeddings from previous layer. The final embeddings of all nodes are derived from the last graph convolutional layer. As we aim to detect any attack in an entire smart grid, a comprehensive embedding of the whole graph is required. Therefore, similar to Section III-B and Section III-C, a READOUT function is utilized to construct a graph embedding by synthesizing representations of all nodes for a graph-level task. Three common choices of READOUT functions are: *Max*, *Mean*, and *Sum*, with *Sum* demonstrating the highest expressive power [37]. The classification tasks are then completed by a downstream linear layer which produces the final prediction result.

IV. Experiments

A. Experiment settings

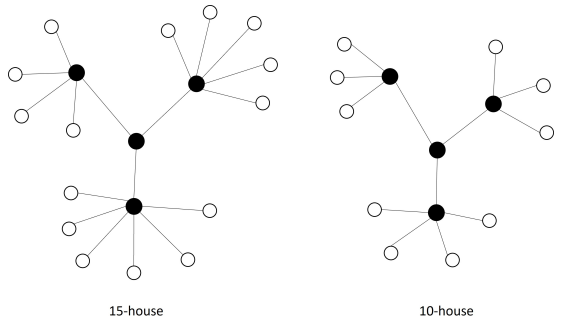

Fig. 2. Graph topology of experimental smart grids

We conducted experiments on two grids depicted in Figure 2: one with 15 houses and another with 10 houses. In this figure, white nodes indicate houses equipped with smart meters while black nodes represent intermediate substations without smart meters. We utilized real-life electric usage data as normal data for our proposed model. This dataset includes two types of usage patterns: single-house and apartment-room patterns. Each day's electric usage data from all smart meters in a grid forms one sample, which consists of 24 hours of usage data. To address the absence of tampered data, we exploit two simulated attack methods on normal data to generate tampered data for our experiments.

- Attack on a single smart meter (single): We randomly pick one smart meter, and reduce all electric usage readings from this smart meter by 20% for each day. This attack aims to preserve the original trends of electric usage, making detection more challenging.
- Attack on all smart meters (all): As a smart meter records 24 numerical data points for the 24-hour period, we randomly select each data point from all smart meters with a 10% probability for each day. All selected data is lowered by 20%. In this case, the attacks are launched randomly, leading to difficulties in detections.

After generating tampered data, the dataset contains 120 normal data and 120 tampered data of the entire smart grid for each experiment.

Our framework was implemented in PyTorch using Python version 3.9. We trained a three-layer GCN with hidden dimensions [128,32,16] and three-layer GIN with hidden dimensions [128,128,128] for each task. We employed the SUM READOUT function for both GIN and GCN models. The Adam optimizer with a learning rate of 0.001 was selected for training the GCNs and GINs. To evaluate model performance, we performed 10-fold cross-validation for each task to obtain the average and standard deviation of overall accuracies, true positive rates (detection rates) and true negative rates, to measure the ability of models on identifying both normal and tampered data. As we utilized the 10-fold cross validation where the dataset is divided into 10 equal parts, and the model is trained on 9 parts while being tested on the remaining 1 part, there is no need to specify the ratio of the training, validation, and test datasets.

However, the dimension size is significantly large due to the concatenation. Therefore, we also conducted experiments using autoencoder to project the representations into a lower-dimensional space. An autoencoder comprises an encoder and a decoder: the encoder projects the input Z_n into a lower-dimensional space, while the decoder is utilized to build a reconstructed input, denoted as $\hat{Z}_n$. The autoencoder is pre-trained using the reconstruction error as the loss function $\mathcal{L} = (Z_n, \hat{Z}_n)$. We refer to this training strategy as *w/ AE*. In our experiments, hidden dimension of autoencoders were set to [64,32], indicating that the latent representation of each node is reduced into 32 dimensions. In our experiments, autoencoders were used to preprocess the input representation of each day. An initial input data with dimension size $24 \times \#houses$ was first projected into lower dimensional space to produce an latent representation. Then latent representations of all data were then fed into MLP classifiers and Decision Tree Classifiers for final prediction. Note that the autoencoder preprocessing was only applied for Decision Tree classifiers and MLP classifiers, while the feature dimensions of nodes in the GNNs remained unchanged throughout the experiments.

B. Experimental results

Table I and Table II show the experimental results of GCNs and GINs, as well as comparisons with MLP and Decision Tree models on 10-house and 15-house, respectively. The training datasets were subjected to two types of attacks: *Attack on a single smart meter* (single) and *Attack on all smart meters* (all). From the experimental results, we observed that performance of GNN models (both GCNs and GINs) consistently outperformed the other two models, where data is treated as

TABLE I
EXPERIMENTAL RESULTS ON 10-HOUSE

	10-house (one)			10-house (all)		
	Accuracy	True positive rate	True negative rate	Accuracy	True positive rate	True negative rate
GCN	**86.58 ± 7.54**	**89.26 ± 13.91**	**83.90 ± 12.26**	**64.65 ± 3.77**	**67.45 ± 11.75**	**61.85 ± 9.57**
GIN	72.57 ± 5.28	74.65 ± 10.98	70.49 ± 13.82	59.77 ± 4.29	58.35 ± 8.12	61.19 ± 7.04
MLP	55.16 ± 7.47	56.48 ± 11.88	54.25 ± 9.37	38.29 ± 9.36	39.85 ± 7.86	36.73 ± 8.06
MLP w/ AE	15.99 ± 4.59	16.10 ± 6.07	16.50 ± 6.78	30.33 ± 4.36	27.44 ± 4.48	33.41 ± 10.53
Decision Tree	68.42 ± 5.63	73.98 ± 7.29	62.86 ± 6.32	55.33 ± 7.52	54.66 ± 7.21	55.96 ± 13.12
Decision Tree w/ AE	27.49 ± 6.09	30.26 ± 7.16	26.43 ± 11.10	34.00 ± 6.53	34.19 ± 9.46	34.35 ± 9.90

TABLE II
EXPERIMENTAL RESULTS ON 15-HOUSE

	15-house (one)			15-house (all)		
	Accuracy	True positive rate	True negative rate	Accuracy	True positive rate	True negative rate
GCN	**78.08 ± 8.85**	**80.91 ± 10.21**	**75.25 ± 9.24**	**57.29 ± 4.10**	**59.15 ± 6.97**	**55.43 ± 6.01**
GIN	71.16 ± 5.81	73.52 ± 7.79	68.81 ± 7.52	54.07 ± 4.94	55.18 ± 7.41	52.96 ± 5.89
MLP	55.16 ± 10.45	59.02 ± 12.66	52.89 ± 11.40	22.33 ± 4.72	19.63 ± 6.05	25.03 ± 11.32
MLP w/ AE	12.66 ± 4.39	14.23 ± 8.13	11.71 ± 11.19	14.66 ± 5.01	13.22 ± 8.80	17.38 ± 12.52
Decision Tree	69.66 ± 8.81	69.20 ± 11.87	71.01 ± 11.16	32.61 ± 8.13	34.10 ± 12.30	31.06 ± 10.61
Decision Tree w/ AE	26.66 ± 8.60	27.52 ± 12.48	25.33 ± 9.37	19.52 ± 5.76	15.96 ± 11.79	24.11 ± 12.04

pure numerical data. Notably, GCNs and GINs demonstrated better performance in detecting attacks from *Attack on a single smart meter*, as *Attack on all smart meters* posed greater challenges due to its strong randomness. Moreover, due to the ability of GNNs to capture the properties of an entire graph, the need of deploying detections on all smart meters individually is eliminated, which reduces the burden on system monitoring. GNN models can exploit the topological feature of a smart grid to better build the embedding of the whole smart grid, particularly since the feature dimension of each node in the graph of a smart grid is relatively small. However, for the traditional methods, since we can't utilize these topological features, we have to build the representation of the entire smart grid by concatenating feature vectors of all devices, resulting in a representation dimension of $24 \times \#houses$. The resulting input dimension is relatively large, and this issue cannot be mitigated by exploiting the autoencoder.

Decision Tree classifiers generally exhibited better performance than MLP classifiers. However, the introduction of autoencoders that are typically employed to address dimensionality issues, deteriorated the performance of both Decision Tree classifiers and MLP classifiers. This experimental result reveals that autoencoders may not be suitable for all scenarios, and their utilization requires careful consideration. Additionally, all models performed better on the 10-house smart grid compared to the 15-house smart grid, as the large number of nodes in 15-house increases the difficulty of predictions. Considering that a large amount of preprocessing techniques and deep learning models available on numerical data without topology information, GNNs may not always be the best choice for cyberattack detections in all cases. However, our work presents a promising alternative for tackling similar tasks.

V. CONCLUSION

In this paper, we propose a new approach to cyberattack detection by utilizing graph topology of smart grids. We begin by defining the network topology of smart grids as graphs. Graphs with featured nodes of smart grids are utilized for graph classification task to identify if any node is under attack. This approach eliminates the necessity for monitoring and analyzing all devices individually. Experimental results based on real-world data indicate that GNN models significantly outperformed MLP classifiers and Decision Tree Classifiers, highlighting the potential of exploiting a GNN-based approach on cybersecurity-related works for smart grids.

ACKNOWLEDGMENT

This material is based upon work supported by the Department of Energy Office of Cybersecurity, Energy Security, and Emergency Response (CESER) under Award Number(s) DE-CR0000023.

REFERENCES

[1] S. Tan, D. De, W.-Z. Song, J. Yang, and S. K. Das, "Survey of security advances in smart grid: A data driven approach," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 397–422, 2016.

[2] A. Ghasempour, "Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges," *Inventions*, vol. 4, no. 1, p. 22, 2019.

[3] A. Khattak, S. Mahmud, and G. Khan, "The power to deliver: Trends in smart grid solutions," *IEEE Power and Energy Magazine*, vol. 10, no. 4, pp. 56–64, 2012.

[4] P. I. R. Grammatikis, P. G. Sarigiannidis, and I. D. Moscholios, "Securing the internet of things: Challenges, threats and solutions," *Internet of Things*, vol. 5, pp. 41–70, 2019.

[5] J. Sakhnini, H. Karimipour, A. Dehghantanha, R. M. Parizi, and G. Srivastava, "Security aspects of internet of things aided smart grids: A bibliometric survey," *Internet of things*, vol. 14, p. 100111, 2021.

[6] P. Kumar, Y. Lin, G. Bai, A. Paverd, J. S. Dong, and A. Martin, "Smart grid metering networks: A survey on security, privacy and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2886–2927, 2019.

[7] R. Leszczyna, "Cybersecurity and privacy in standards for smart grids–a comprehensive survey," *Computer Standards & Interfaces*, vol. 56, pp. 62–73, 2018.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[9] K. Chowdhary and K. Chowdhary, "Natural language processing," *Fundamentals of artificial intelligence*, pp. 603–649, 2020.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[11] S. Kwon, H. Yoo, and T. Shon, "Ieee 1815.1-based power system security with bidirectional rnn-based network anomalous attack detection for cyber-physical system," *IEEE Access*, vol. 8, pp. 77 572–77 586, 2020.

[12] O. A. Alimi, K. Ouahada, and A. M. Abu-Mahfouz, "A review of machine learning approaches to power system security and stability," *IEEE Access*, vol. 8, pp. 113 512–113 531, 2020.

[13] B. Klaer, Ö. Sen, D. van der Velde, I. Hacker, M. Andres, and M. Henze, "Graph-based model of smart grid architectures," in *2020 International conference on smart energy systems and technologies (SEST)*. IEEE, 2020, pp. 1–6.

[14] J. Wang, X. Wang, C. Ma, and L. Kou, "A survey on the development status and application prospects of knowledge graph in smart grids," *IET Generation, Transmission & Distribution*, vol. 15, no. 3, pp. 383–407, 2021.

[15] G. Liang, J. Zhao, F. Luo, S. R. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1630–1638, 2016.

[16] A. Hoehn and P. Zhang, "Detection of replay attacks in cyber-physical systems," in *2016 American control conference (ACC)*. IEEE, 2016, pp. 290–295.

[17] X. Li, X. Liang, R. Lu, X. Shen, X. Lin, and H. Zhu, "Securing smart grid: cyber attacks, countermeasures, and challenges," *IEEE Communications Magazine*, vol. 50, no. 8, pp. 38–45, 2012.

[18] F. G. Mármol, C. Sorge, O. Ugus, and G. M. Pérez, "Do not snoop my habits: preserving privacy in the smart grid," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 166–172, 2012.

[19] S. Banik, S. K. Saha, T. Banik, and S. M. Hossain, "Anomaly detection techniques in smart grid systems: A review," in *2023 IEEE World AI IoT Congress (AIIoT)*. IEEE, 2023, pp. 0331–0337.

[20] I. Siniosoglou, P. Radoglou-Grammatikis, G. Efstathopoulos, P. Fouliras, and P. Sarigiannidis, "A unified deep learning anomaly detection and classification approach for smart grid environments," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1137–1151, 2021.

[21] A. Aflaki, M. Gitizadeh, R. Razavi-Far, V. Palade, and A. A. Ghasemi, "A hybrid framework for detecting and eliminating cyber-attacks in power grids," *Energies*, vol. 14, no. 18, p. 5823, 2021.

[22] T. Wu, W. Xue, H. Wang, C. Chung, G. Wang, J. Peng, and Q. Yang, "Extreme learning machine-based state reconstruction for automatic attack filtering in cyber physical power system," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1892–1904, 2020.

[23] L. Cui, Y. Qu, L. Gao, G. Xie, and S. Yu, "Detecting false data attacks using machine learning techniques in smart grid: A survey," *Journal of Network and Computer Applications*, vol. 170, p. 102808, 2020.

[24] T. Berghout, M. Benbouzid, and S. Muyeen, "Machine learning for cybersecurity in smart grids: A comprehensive review-based study on methods, solutions, and prospects," *International Journal of Critical Infrastructure Protection*, vol. 38, p. 100547, 2022.

[25] X. Lin, D. An, F. Cui, and F. Zhang, "False data injection attack in smart grid: Attack model and reinforcement learning-based detection method," *Frontiers in Energy Research*, vol. 10, p. 1104989, 2023.

[26] A. Takiddin, R. Atat, M. Ismail, O. Boyaci, K. R. Davis, and E. Serpedin, "Generalized graph neural network-based detection of false data injection attacks in smart grids," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 3, pp. 618–630, 2023.

[27] O. Boyaci, A. Umunnakwe, A. Sahu, M. R. Narimani, M. Ismail, K. R. Davis, and E. Serpedin, "Graph neural networks based detection of stealth false data injection attacks in smart grids," *IEEE Systems Journal*, vol. 16, no. 2, pp. 2946–2957, 2021.

[28] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.

[29] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[30] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[31] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, pp. 3844–3852, 2016.

[32] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," *arXiv preprint arXiv:1509.09292*, 2015.

[33] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *International conference on machine learning*. PMLR, 2016, pp. 2014–2023.

[34] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[35] G. Li, C. Xiong, A. Thabet, and B. Ghanem, "Deepergcn: All you need to train deeper gcns," *arXiv preprint arXiv:2006.07739*, 2020.

[36] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[37] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[38] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," *arXiv preprint arXiv:2002.07962*, 2020.

[39] L. Gong and Q. Cheng, "Exploiting edge features for graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9211–9219.

[40] X. Yue, B. Liu, F. Zhang, and G. Qu, "Edged weisfeiler-lehman algorithm," in *International Conference on Artificial Neural Networks*. Springer, 2024, pp. 93–109.

[41] B. Weisfeiler and A. Leman, "The reduction of a graph to canonical form and the algebra which appears therein," *NTI, Series*, vol. 2, no. 9, pp. 12–16, 1968.

[42] L. Babai and L. Kucera, "Canonical labelling of graphs in linear average time," in *20th annual symposium on foundations of computer science (sfcs 1979)*. IEEE, 1979, pp. 39–46.

[43] R. Sato, "A survey on the expressive power of graph neural networks," *arXiv preprint arXiv:2003.04078*, 2020.