# Transferable Learning of GCN Sampling Graph Data Clusters from Different Power Systems

Tong Wu, Anna Scaglione, Daniel Arnold and Tianyi Chen

September 24, 2024

# Transferable Learning of GCN Sampling Graph Data Clusters from Different Power Systems

Tong Wu, *Member*, Anna Scaglione, *Fellow, IEEE*, Daniel Arnold, *Member, IEEE*, Tianyi Chen, *Member, IEEE*

***Abstract*—Contemporary neural network (NN) detectors for power systems face two primary challenges. First, each power system requires individual training of NN detectors to accommodate its unique configuration and base demands. Second, significant changes within the power system, such as the introduction of new substations or new generators, necessitate retraining. To overcome these issues, we introduce a novel architecture, the Nodal Graph Convolutional Neural Network (NGCN), which utilizes graph convolutions at each bus and its neighborhoods. This approach allows the training process to encompass multiple power systems and include all buses, thereby enhancing the transferability of the method across different power systems. The NGCN is particularly effective for detection tasks, such as cyber-attacks on smart inverters and false data injection attacks. Our tests demonstrate that the NGCN significantly improves performance over traditional NNs, boosting detection accuracy from approximately 85% to around 97% for the aforementioned task. Furthermore, the transferable NGCN, which is trained by samples from multiple power systems, performs considerably better in evaluations than the NGCN trained on a single power system.**

## I. Introduction

### A. Background and Motivation

Power systems are safety-critical infrastructures highly susceptible to cyber-attacks [1]. These attacks, targeting devices like smart inverters, can trigger destabilizing oscillations within the system [2]. Moreover, cyber-attacks on power generators may lead to even more severe consequences, such as widespread blackouts [3]. It is, therefore, imperative to detect such attacks promptly as they occur. Early detection allows for the necessary control actions to be taken to mitigate or completely avert these disruptive events [4].

Recent research has increasingly concentrated on utilizing machine learning methods, especially deep learning, to detect cyber-attacks and false data injection (FDI) attacks in power systems. A thorough review of these advancements can be found in [5]. Deep learning leverages neural networks to train detectors based on available labeled data. In grid applications these data can be generated through simulations, since historical labeled data from cyber-attacks are often not available. The neural networks trained are highly effective in discerning the complex relationships between PMU (Phasor Measurement Unit) or AMI (Advanced Metering Infrastructure) measurements in the presence of attacks, identifying not only if an attack is occurring within the power system [6] but also pointing out the specific buses or devices under attack [7]. Neural network-based detectors are typically trained on data simulated for a specific power system, meaning they rely on samples generated using the target grid's unique configuration, including factors like the number of buses, branches, and base demand. Since these configurations can differ substantially between systems, the model parameters often need to be retrained for each new environment. Minor perturbations in the system do not pose significant issues and generally do not necessitate retraining. In the case of significant changes within the power system, such as the introduction of new solar PV inverters, load points, substations or generators, the retraining of these detectors is, unfortunately, necessary. Each retraining process is time-consuming, creating a significant operational challenge and associated financial and manpower costs. Moreover, the resources and expertise needed for effective training are not commonly available to power system operators, particularly at the distribution level, where this issue is especially pronounced.

### B. Related Work

Existing machine learning methods for detecting cyber-attacks in power grids are broadly classified into two categories: physics-agnostic deep learning [8–11] and physics-aware graph convolutional network (GCN) methods [7, 12, 13]. For physics-agnostic approaches, [10, 11] employed fully connected neural networks to detect cyber-attacks targeting power systems and transmission protective relays. [8] applied convolutional neural networks to leverage the local correlations of voltage phasors. Additionally, [9] utilized recurrent neural networks to capture the temporal correlations in time-series measurements of power systems. In the physics-aware category, [7, 13] implemented spatio-temporal GCNs for detecting FDI in power systems. Furthermore, [12] extended GCNs into the complex domain, considering the power system's admittance matrix as the graph shift operator. This method significantly outperforms other neural networks and GCNs in detecting false data injection attacks due to its ability to extract different graph spectral components of voltage phasors to analyze, exploiting the low-pass properties of normal voltage phasors and the high-pass properties of attacked measurements [14].

Tong Wu is with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, 32816 USA (e-mail: tong.wu@ucf.edu). Anna Scaglione is with the Department of Electrical and Computer Engineering, Cornell Tech, Cornell University, New York City, NY, 10044 USA (e-mail: as337@cornell.edu). Daniel Arnold is with Lawrence Berkeley National Laboratory (e-mail: dbarnold@lbl.gov). Tianyi Chen is with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, 12180 USA (chent18@rpi.edu).

This research was supported in part by the NSF under Grant NSF ECCS # 2210012, in part by the Director and in part by the CESER, Risk Management and Tools and Technologies (RMT) program of the U.S. Department of Energy via Mitigation via Analytics for Grid-Inverter Cybersecurity (MAGIC) project under contract DE-AC02-05CH11231. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors of this work.

### C. Contributions and Organization

This paper introduces what we refer to as Nodal Graph Convolutional Neural Networks (NGCNs), a new graph neural network architecture designed for scalable training across buses from different power systems. Rather than training on data that simulate label date from a single system our method includes two components: 1) a neural network architecture that can be adapted to different topologies and 2) a training process involves sampling features data from sub-graphs that are part of various power systems and smart inverters configuration and attack scenarios, to develop a universal detector that can be adapted without retraining for another arbitrary system. The main contributions of our research are outlined below:

- We have developed a novel NGCN framework capable of performing node convolutions for each bus and its neighboring areas. This framework has been further enhanced to perform both graph and temporal convolutions, capturing the spatio-temporal correlations of voltage phasors. Our NGCN effectively extracts graph spectral features that are critical for distinguishing between normal and attacked states. This distinction is based on the observation that normal voltage phasors typically result from the outputs of low-pass graph filters [14], which primarily exhibit smooth components, whereas attacked states often resemble high-pass or band-pass graph signals.
- We show how one can curate a training set for the NGCN that encompasses a variety of power grids simulations, with subsets of buses equipped with smart inverters of different sizes. The NGCN is trained by subsampling different power grids and buses, with labels indicating the presence or absence of a cyber attack. Once trained, the NGCN can be deployed across any power system and bus to perform detection tasks. Additionally, we extend the transferable NGCN to detect FDI attacks across buses in different power systems.

### D. Notation

The grid network is modeled as an undirected weighted graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, where the set of nodes $\mathcal{V}=\{1,\ldots,N\}$ represents the buses, and the set of edges $\mathcal{E}\subsetneq\mathcal{V}\times\mathcal{V}$ corresponds to the transmission lines, which can be overhead or underground. The subset of buses equipped with smart inverters is denoted by $\mathcal{V}_s$, with $N_s$ representing its cardinality. For each node $n\in\mathcal{V}$, let $v_n\in\mathbb{C}$ be the complex line-to-ground voltage, and $\boldsymbol{v}=[v_1,\ldots,v_N]^\top$ the vector of voltage phasors across all buses, where $v_n$ has magnitude $|v_n|$ and phase angle $\theta_n$. Similarly, the vectors $\boldsymbol{i}$, $\boldsymbol{s}$, $\boldsymbol{p}$, and $\boldsymbol{q}$ represent current injection phasors, apparent power injections, active power, and reactive power injections, respectively, all having the same dimension as $\boldsymbol{v}$. The apparent power vector is defined as $\boldsymbol{s}=\boldsymbol{p}+\mathrm{j}\boldsymbol{q}$, where $\mathrm{j}=\sqrt{-1}$ is the imaginary unit.

## II. Spatio-Temporal Nodal Graph Convolutional Neural Networks

In this section, we first explore the concept of graph filters and reveal the essential mechanisms for adapting their application to data, guided by the underlying physics of the grid. In this study, we extend our previous work as detailed in [12] to the development of NGCN sampling graph data clusters from different power systems.

### A. Physics-Aware Grid GCN over One Graph

To enhance the feature extraction layers in comparison to traditional neural networks, graph filters can be leveraged for improved representation capabilities. To define these filters, we first introduce some key concepts, starting with the Graph Shift Operator (GSO). A graph signal, denoted by $\boldsymbol{x}\in\mathbb{C}^N$, is a vector indexed by the nodes of a graph; for example, it represents the state vector of voltage phasors at each bus in a power grid. The neighborhood of node $i$, represented as $\mathcal{V}_{\delta(i)}$, refers to the set of nodes directly connected to node $i$, where $\delta(i)$ denotes the neighborhoods of node $i$. The GSO, represented by the matrix $\mathbf{S}\in\mathbb{C}^{N\times N}$, is a neighborhood operator that only combines the values from neighboring nodes. We consider complex symmetric GSOs, i.e., $\mathbf{S}=\mathbf{S}^\top$. Typically designed to mimic a differential operator, the GSO is commonly selected as a graph weighted Laplacian matrix. A graph filter is a linear matrix operator $\mathcal{H}(\mathbf{S})$, which is a function of the GSO. It acts on graph signals as follows:

$$\boldsymbol{x}^1=\mathcal{H}(\mathbf{S})\boldsymbol{x}^0, \tag{1}$$

where $\boldsymbol{x}^0$ denotes the input features. In power systems applications, $\boldsymbol{x}^0$ is typically the state vector $\boldsymbol{v}$.

The key characteristic of the graph filter $\mathcal{H}(\mathbf{S})$ is that it must be shift-invariant with GSO, similar to the time-invariant filters in the time domain. This means that $\mathcal{H}(\mathbf{S})$ must satisfy the condition $\mathcal{H}(\mathbf{S})\mathbf{S}=\mathbf{S}\mathcal{H}(\mathbf{S})$. This property holds only if $\mathcal{H}(\mathbf{S})$ can be expressed as a matrix polynomial:

$$\mathcal{H}(\mathbf{S})=\sum_{k=0}^{K}h_k\mathbf{S}^k, \tag{2}$$

where the graph filter order $K$ can potentially be infinite. Based on Eq. (2), we can construct the graph neural network perceptron, which is expressed as

$$\boldsymbol{x}^\ell=\sigma\left[\sum_{k=0}^{K-1}h_k\mathbf{S}^k\boldsymbol{x}^{\ell-1}\right],\ell=1,\cdots,N \tag{3}$$

where the input feature vector is $\boldsymbol{x}^0=\boldsymbol{v}\in\mathbb{C}^N$ or $\boldsymbol{x}^0=[|\boldsymbol{v}|,\angle\boldsymbol{v}]\in\mathbb{R}^{N\times 2}$. The GSO raised to the power $k$, denoted as $\mathbf{S}^k$, is a matrix of size $\mathbb{C}^{N\times N}$. The resulting graph signal $\boldsymbol{x}$ also belongs to $\mathbb{C}^N$, while $h_k\in\mathbb{C}$ represents a scalar coefficient. The function $\sigma(\cdot)$ denotes the activation function, typically chosen as ReLU for the hidden layers. The final layer of the GCN for a node-level task is given by:

$$\boldsymbol{y}=\sigma\left[\boldsymbol{\Theta}\begin{pmatrix}\Re(\boldsymbol{x}^L)\\ \Im(\boldsymbol{x}^L)\end{pmatrix}\right] \tag{4}$$

where $\boldsymbol{\Theta}\in\mathbb{R}^{N\times 2N}$ contains the trainable parameters of the output layer, and $\boldsymbol{x}^L$ is the feature representation from the last hidden layer. However, a major limitation of traditional GCNs is that for different graphs, a separate set of $h_k$ must be trained for each distinct $\mathbf{S}$, since the dimensions of both $\mathbf{S}$ and the graph signal $\boldsymbol{x}$ can vary across different graphs. As a result, the dimensionality of $\mathbf{\Theta}$ also changes, making it impossible to design a single neural network that can process all graphs with varying $\mathbf{S}$ and $\boldsymbol{x}$.

### *B. Graph Convolution across Different Graphs*

To make GCNs transferable across different graphs with varying $\mathbf{S}$ and $\boldsymbol{x}$, one potential method is to employ the decomposition of full graph convolutions into nodal graph convolutions. Unlike [15], where the graph is divided into subgraphs containing several neighborhoods, we decompose the entire graph into individual nodes neighborhoods and mix nodes from different graphs during training. By sampling data of neighbohoods that are part of multiple power systems, By sampling neighborhood data from multiple power systems, a single GCN can be trained and applied to any bus in different power grids that is statistically similar to the ensemble used for training. This uniqueness allows the model to be transferable across different systems. In the following sections, we introduce new settings that address general subgraph convolutions. As a special case, these settings enable us to develop nodal-graph convolutional neural networks.

More specifically, consider $Q$ distinct power systems, each represented as a connected component within a large, unconnected graph. Each component corresponds to a specific power system. Each component corresponds to a specific power system, and from this point, we refer to each as a subgraph. This unconnected graph can be thought of as being sampled from the set $\mathcal{Q} = [1, 2, \dots, Q]$. For each power system $q \in \mathcal{Q}$, the graph $\mathcal{G}_q$ consists of a node set $\mathcal{V}_q$, where $N_q = |\mathcal{V}_q|$ is the number of nodes, a weighted Laplacian matrix $\mathbf{S}_q \in \mathbb{C}^{N_q \times N_q}$, and a node data matrix $\mathbf{X}_q \in \mathbb{C}^{N_q \times m_0}$, where $m_0$ represents the number of node features. To represent all $Q$ power systems together, we define a global, "big" Laplacian matrix for the combined system. Let $N = \sum_{q=1}^{Q} N_q$ be the total number of nodes across all infrastructures. With a slight abuse of notation, this global Laplacian matrix captures the relationships across the nodes of all power systems combined:

$$\mathbf{S} := \begin{bmatrix} \mathbf{S}_1 & \vdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{S}_Q \end{bmatrix} \in \mathbb{C}^{N \times N}. \tag{5}$$

Similarly, we can define the global "big" node feature matrix $\mathbf{X}$ and the global graph shift operator $\mathbf{S}$ to represent all infrastructures combined. For each subgraph $\mathcal{G}_m$, we introduce the matrix $\mathbf{P}_{\mathcal{G}_m} \in \mathbb{R}^{N \times N}$ (or simply $\mathbf{P}_m$), a diagonal matrix where the $n$-th diagonal entry is zero if the $n$-th node is not part of the node set $\mathcal{V}_m$ of subgraph $\mathcal{G}_m$. More formally, $[\mathbf{P}_m]_{nn} = 0$ if $n \notin \mathcal{V}_m$, and $[\mathbf{P}_m]_{n'n'} = 1$ if $n' \in \mathcal{V}_m$. For each subgraph $\mathcal{G}_m = \{\mathcal{V}_m, \mathcal{E}_m\}$, sampled from any of the $Q$ infrastructures $\{\mathcal{G}_q\}$, we define the corresponding "expanded" Laplacian matrix as $\mathbf{S}_m = \mathbf{P}_m \mathbf{S} \mathbf{P}_m \in \mathbb{C}^{N \times N}$. This expanded Laplacian captures the structure of the $m$-th subgraph in the global context as follows:

$$\mathbf{S}_m = \mathbf{P}_m \mathbf{S} \mathbf{P}_m = \tag{6}$$

$$\begin{bmatrix} 0 & \dots & 0 \\ \vdots & \mathbf{I}_{N_m} & \vdots \\ 0 & \dots & 0 \end{bmatrix} \times \begin{bmatrix} \mathbf{S}_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{S}_Q \end{bmatrix} \times \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \mathbf{I}_{N_m} & \vdots \\ 0 & \dots & 0 \end{bmatrix} \tag{7}$$

where $\mathbf{I}_{N_m}$ is an $N_m \times N_m$ identity matrix with $N_m = |\mathcal{V}_m|$.

Since $N$ is very large, training the entire $N$-dimensional vector $\mathbf{S}_m$ may appear to be a daunting task. In reality, the large matrix $\mathbf{S}_m$ has only one active subgraph within itself, which is the one selected by $\mathbf{P}_m$. To clarify, while we express the operations and loss function in terms of the larger $\mathbf{S}_m$ for the mathematical formulation, in our practical implementation of the training, we only consider the non-zero elements that correspond to the sampled sub-graph, i.e. the neighborhoods we sampled. Importantly, let $\mathcal{V}_m^k$ be the set of nodes that includes the $m$-th node and all its neighbors up to $k$-hops. The matrix $\mathbf{P}_m^k$ is a diagonal matrix that selects the nodes in $\mathcal{V}_m^k$, while zeroing out all other nodes in the global system. The diagonal entries of $\mathbf{P}_m^k$ can be defined as:

$$[\mathbf{P}_m^k]_{nn} = \begin{cases} 1 & \text{if } n \in \mathcal{V}_m^k \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Given $\mathbf{P}_m^k$, the expanded Laplacian matrix $\mathbf{S}_m^k$ can be defined as:

$$\mathbf{S}_m^k = \mathbf{P}_m^k \mathbf{S}^k \mathbf{P}_m^k, \tag{9}$$

where $\mathbf{S}$ is the global Laplacian matrix representing the entire system of subgraphs. The matrix $\mathbf{S}_m^k$ captures the interactions between nodes within the $k$-hop neighborhood of the $m$-th node.

**Remark 1** *A special case arises when $k = 0$. In this case, $\mathcal{V}_m^0$ consists of only the $m$-th node, and $\mathbf{S}^0 = \mathbf{I}$, the identity matrix. The matrix $\mathbf{P}_m^0$ selects only the $m$-th node from the global graph. In matrix form, $\mathbf{P}_m^0$ can be written as:*

$$\mathbf{P}_m^0 = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & 1 & \vdots \\ 0 & \dots & 0 \end{bmatrix} \tag{10}$$

*Thus, the expanded Laplacian matrix for $k = 0$, denoted $\mathbf{S}_m^0$, is:*

$$\mathbf{S}_m^0 = \mathbf{P}_m^0 \mathbf{I} \mathbf{P}_m^0 = \mathbf{P}_m^0 \tag{11}$$

*In this case, $\mathbf{S}_m^0$ effectively isolates the $m$-th node and treats it independently because $\mathbf{S}^0 = \mathbf{I}$ does not introduce any interaction between the nodes.*

### *C. Nodal-Graph Convolution Neural Network*

For multi-feature input, let $\mathbf{X} = [\boldsymbol{x}^1; \dots; \boldsymbol{x}^{m_0}] \in \mathbb{R}^{N \times m_0}$ represent the input feature set, and $\mathbf{H} \in \mathbb{R}^{m_l \times m_{l+1}}$ denote the multi-channel outputs. Here, $m_l$ refers to the total number of input features, and $m_{l+1}$ represents the total number of output channels. The filtering operation can then be expressed as:

$$\mathbf{X}^{l+1} = \sigma\left(\sum_{k=0}^{K-1} \mathbf{S}_m^k \mathbf{X}^l \mathbf{H}_k^l\right), \tag{12}$$

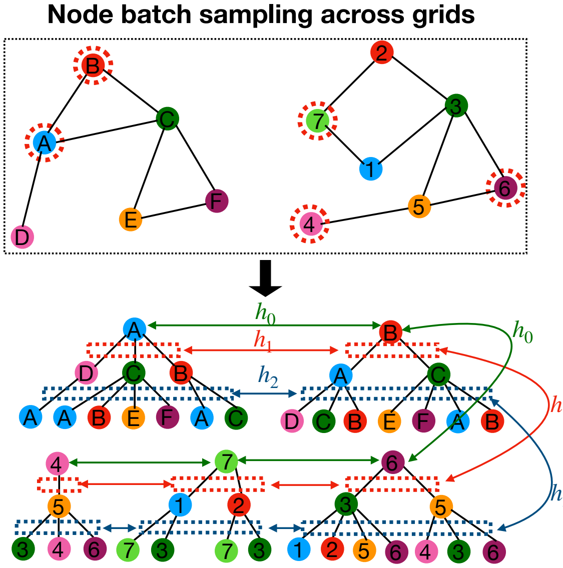

Fig. 1: General Graph AI node sampling across different power systems. The two distinct power system graphs at the top illustrate how nodes are selected for sampling from different grids. The dashed circles highlight the nodes being sampled. The lower section of the figure shows the corresponding computational graph, where node features are propagated through different layers ($h_0$, $h_1$, $h_2$), representing hierarchical node embeddings over various depths of the graph. The dashed boxes indicate that the parameters are shared among nodes in the same layer, while the color-coded edges signify the use of identical trainable parameters across different graphs.

where $\sigma(\cdot)$ is a non-linear activation function (e.g., sigmoid or ReLU), and $\mathbf{H}_k^l \in \mathbb{R}^{m_l \times m_{l+1}}$ are the $k$-th trainable weight matrices that map the node features from $m_l$ to $m_{l+1}$. Each element of $\mathbf{H}_k^l$ corresponds to $h_k$ in (3). Importantly, the operation $\mathbf{S}_m^k$ should first perform the graph convolution, followed by the selection process using $\mathbf{P}_m$.

For all $Q$ graphs $\{\mathcal{G}_q\}_{q=1}^{Q}$, we define the corresponding sampling matrices $\{\mathbf{P}_m\}_{m=1}^{M}$. The training objective function is then expressed as:

$$\min_{\mathbf{H}} \quad \mathsf{L}(\mathbf{H};\mathbf{S},\mathbf{X}) := \frac{1}{M}\sum_{m=1}^{M} \underbrace{\ell(\mathbf{H};\mathbf{P}_m\mathbf{S}\mathbf{P}_m,\mathbf{X},y_m)}_{\mathsf{L}(\mathbf{H};\mathbf{S},\mathbf{X},\mathbf{P}_m)}, \quad (13)$$

where $\{y_m\}$ denotes the set of labeled nodes across the union of all subgraphs, represented as $\bigcup_m \mathcal{V}_m$.

In (13), the goal is to learn a GCN model $\mathbf{H}$ that is shared across multiple subgraphs from different infrastructures. One advantage of this subgraph-based training objective is that it trains the GCN model using only local information from subgraphs. For example, during stochastic gradient descent, for a selected subgraph $\mathcal{G}_m$ and GCN weights $\{\mathbf{H}^l\}$, the forward pass computes the embedding of (12). Because the sampling matrix $\mathbf{P}_m\mathbf{S}\mathbf{P}_m$ selects a node $i \in \mathcal{G}_m$, the embedding (row) vector $\mathbf{x}_i^{l+1}$ at layer $l+1$ follows the recursion:

$$\mathbf{x}_i^{l+1} = \sigma\left(\sum_{j\in\mathcal{N}_i(\mathcal{G}_m)} \sum_{k=0}^{K-1} [\mathbf{S}_m^k]_{ij} (\mathbf{x}_i^l)^\top \mathbf{H}_k^l\right) \in \mathbb{R}^{1\times m_{l+1}},$$

where $\mathcal{N}_i(\mathcal{G}_m)$ represents the neighbors of node $i$ in the subgraph $\mathcal{G}_m$, and $\mathbf{x}_i$ is its corresponding node feature. The node selection by $\mathbf{P}_m$ ensures that only one diagonal element is set to 1, corresponding to node $i$. After this selection, each data sample has a dimension of $1 \times m_L$.

To illustrate this, consider the toy example in Fig. 1, which features two graphs with $K = 3$. Each nodal graph consists of a node along with its first-order and second-order neighborhoods, which we define as the *computational nodal graph*. These graphs use local data to train the parameters $\mathbf{H}_k$. It is important to emphasize that, in this NGCN framework, the training of $\mathbf{H}_k$ is performed independently by each node and its neighborhoods. However, all nodal graphs share the same set of parameters $\mathbf{H}_k$. Additionally, these computational nodal graphs from different power systems must converge to a unified set of $\mathbf{H}_k$. Therefore, each training batch will include computational nodal graphs sampled from different power systems to ensure consistency across the network.

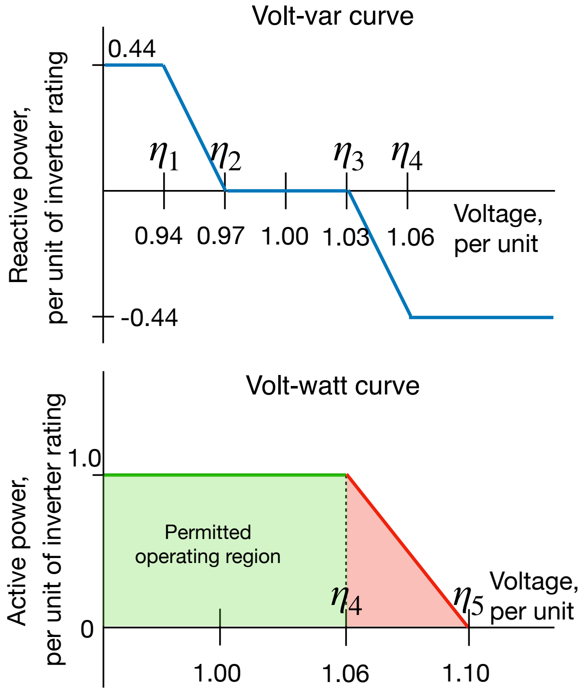

Fig. 2: Volt-Var (VV) and Volt-Watt (VW) piece-wise functions.

## III. Transferable NGCN Applications in Power Systems

This section presents two applications of the Transferable NGCN for enhancing security in power systems: detecting cyber-attacks on smart inverters and identifying FDI attacks.

### A. Cyber-Attack Detection on Smart Inverters

The proposed model uses voltage phasor from each inverter (node) and its neighboring nodes as input. The output of the NGCN will predict whether an inverter is under attack, denoted as $y_i = 1$, or functioning normally, denoted as $y_i = 0$. This approach allows for a scalable, system-agnostic solution that enhances the security of smart inverters in varying power network configurations. To provide a clear context, we first outline the operational model of smart inverters below:

*1) Smart Inverter Control Model:* The control of power injection for smart inverters is determined by two piecewise linear functions, known as Volt-Var (VV) and Volt-Watt (VW) curves. These droop curves dictate the reactive and active power outputs based on the voltage magnitude. As shown in Fig. 2, these curves are parameterized by five key values, denoted as $\boldsymbol{\eta} = [\eta_1, \eta_2, \eta_3, \eta_4, \eta_5]^\top \in \mathbb{R}^5$, which define the segments of the curves. The Volt-Var curve is defined as:

$$f_n^q(|\tilde{v}_i|) = \begin{cases} \bar{q} & |\tilde{v}_i| \leq \eta_1, \\ \left(\frac{\eta_2 - |\tilde{v}_i|}{\eta_2 - \eta_1}\right)\bar{q} & \eta_1 < |\tilde{v}_i| \leq \eta_2, \\ 0 & \eta_2 < |\tilde{v}_i| \leq \eta_3, \\ -\left(\frac{\eta_4 - |\tilde{v}_i|}{\eta_4 - \eta_3}\right)\bar{q} & \eta_3 < |\tilde{v}_i| \leq \eta_4, \\ -\bar{q} & |\tilde{v}_i| > \eta_4, \end{cases} \tag{14}$$

and the Volt-Watt curve is described by:

$$f_i^p(|\tilde{v}_i|) = \begin{cases} \tilde{p} & |\tilde{v}_i| \leq \eta_4, \\ \left(\frac{\eta_5 - |\tilde{v}_i|}{\eta_5 - \eta_4}\right)\bar{p} & \eta_4 < |\tilde{v}_i| \leq \eta_5, \\ 0 & |\tilde{v}_i| > \eta_5, \end{cases} \tag{15}$$

where $\bar{p}$ and $\bar{q}$ represent the active and reactive power outputs, respectively, based on the filtered voltage magnitude $|\tilde{v}_i|$. The filtered voltage, $|\tilde{v}_i|$, is obtained by applying a low-pass filter to the measured voltage $|v_{i,t}|$ at bus $i$, thereby reducing noise. The filtering process is given by:

$$|\tilde{v}_{i,t}| = |\tilde{v}_{i,t-1}| + \tau_n^c(|v_{i,t}| - |\tilde{v}_{i,t-1}|), \tag{16}$$

where $\tau_n^c$ is the time constant of the low-pass filter. The operation of the smart inverter is constrained by its capacity limit $\bar{s}$, which bounds the active and reactive power outputs:

$$\bar{q}^2 + f^p(|\tilde{v}_{i,t}|)^2 \leq \bar{s}^2. \tag{17}$$

To ensure smooth transitions in power injection, the dynamics of active and reactive power evolve gradually over time by

$$\begin{aligned} p_{i,t} &= p_{i,t-1} + \tau^o(f_i^p(|\tilde{v}_{i,t}|) - p_{i,t-1}), \\ q_{i,t} &= q_{i,t-1} + \tau^o(f_n^q(|\tilde{v}_{i,t}|) - q_{i,t-1}), \end{aligned} \tag{18}$$

where $\tau^o$ is a time constant that regulates the rate of change. The resulting complex power injected into the system at each time step is expressed as $s_{i,t} = p_{i,t} + jq_{i,t}$.

*2) Cyber-Attacks on Smart Inverters:* Cyber-attacks targeting smart inverters can destabilize the power system by manipulating the VV and VW control settings. To frame this scenario, we assume each node in the network is equipped with a smart inverter capable of VV/VW control, making the total number of inverters $N_s$. The set of inverters, $\mathcal{V}_s$, can be divided into two subsets: $\mathcal{H}$, representing compromised inverters, and $\mathcal{U}$, representing uncompromised ones, such that $\mathcal{H} \cup \mathcal{U} = \mathcal{V}_s$.

For compromised inverters, an attacker can maliciously alter the VV and VW setpoints, denoted by the vector $\boldsymbol{\eta} = [\eta_1, \eta_2, \eta_3, \eta_4, \eta_5]^\top \in \mathbb{R}^5$. Typically, these setpoints are configured as $\boldsymbol{\eta} = [0.94, 0.97, 1.03, 1.06, 1.10]$ under normal operating conditions. However, an adversary can manipulate them to a more detrimental configuration, such as $\boldsymbol{\eta} = [0.98, 0.99, 1.01, 1.02, 1.10]$, potentially destabilizing the system. These changes can trigger instability by reducing the deadband of the VV control curves, which leads to oscillation in power injections, particularly in the reactive power component, $q_{i,t}$. This causes the reactive power to oscillate between negative and positive values, resulting in oscillations in the voltage profile across the system. To detect such attacks, we employ a loss function based on Binary Cross-Entropy. The model's input consists of time-series voltage phasor data, including both voltage magnitudes and phase angles.

### *B. False Data Injection Detection*

The second task for the transferable NGCN is the detection of FDI attacks. Unlike cyber-attacks, which alter input features while maintaining power system states that satisfy power flow equations, FDI attacks violate these equations. FDI detection is commonly formulated as a binary hypothesis testing problem, where the null hypothesis assumes no false data, and the alternative hypothesis represents the presence of compromised measurements. This subsection focuses on localizing these compromised measurements (the FDI localization problem), which can be framed as a multi-label classification task. Each measurement is classified as either valid or false.

Stealth FDI attacks are particularly challenging, as they are designed to escape detection by residual-based state estimation methods. To effectively detect such attacks, it is crucial to account for the physical laws that govern power systems, namely Kirchhoff's and Ohm's laws. These relationships can be expressed as follows:

$$\boldsymbol{i} = \boldsymbol{Y}\boldsymbol{v}, \quad v_n = |v_n|e^{j\varphi_n^v}, \quad i_n = |i_n|e^{j\varphi_n^i}, \quad \forall n \in \mathcal{V}, \tag{19}$$

where $\boldsymbol{v} \in \mathbb{C}^{N\times 1}$ and $|\boldsymbol{v}| \in \mathbb{R}_+^{N\times 1}$ represent the vectors of bus voltage phasors and magnitudes, respectively, and $\boldsymbol{i} \in \mathbb{C}^{N\times 1}$ and $|\boldsymbol{i}| \in \mathbb{R}_+^{N\times 1}$ denote the vectors of bus current phasors and magnitudes. Let $\mathcal{A}$ denote the set of available measurements, and $\mathcal{U}$ represent the set of unavailable measurements. Using this setup, we can express the observed data $\boldsymbol{z}_t$ as follows:

$$\underbrace{\begin{bmatrix} \hat{\boldsymbol{i}}_{\mathcal{A}} \\ \hat{\boldsymbol{v}}_{\mathcal{A}} \end{bmatrix}}_{\boldsymbol{z}_t} = \underbrace{\begin{bmatrix} \boldsymbol{Y}_{\mathcal{A}\mathcal{A}} & \boldsymbol{Y}_{\mathcal{A}\mathcal{U}} \\ \mathbb{I}_{|\mathcal{A}|} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{H}} \underbrace{\begin{bmatrix} \boldsymbol{v}_{\mathcal{A}} \\ \boldsymbol{v}_{\mathcal{U}} \end{bmatrix}}_{\boldsymbol{x}_t} + \boldsymbol{\varepsilon}_t, \tag{20}$$

where $\boldsymbol{\varepsilon}_t$ is the measurement noise vector.

A stealth FDI attack manipulates voltage and current phasor measurements at specific buses, denoted as $\mathcal{C}$, by introducing a perturbation vector $\delta\boldsymbol{x}_t$. This perturbation is defined as:

$$\delta\boldsymbol{x}_t^\top = \begin{bmatrix} \delta\boldsymbol{x}_{\mathcal{C}}^\top & \boldsymbol{0}_{\mathcal{P}+\mathcal{U}}^\top \end{bmatrix}, \quad \text{where} \quad \boldsymbol{Y}_{\mathcal{P}+\mathcal{C}}\delta\boldsymbol{x}_{\mathcal{C}} = \boldsymbol{0}, \quad \mathcal{C} \subset \mathcal{A}. \tag{21}$$

The set $\mathcal{C} \subset \mathcal{A}$ consists of randomly sampled buses that have been compromised by the FDI attack. Here, $\mathcal{P}$ represents the set of unaffected (honest) nodes. The condition $\boldsymbol{Y}_{\mathcal{P}+\mathcal{C}}\delta\boldsymbol{x}_{\mathcal{C}} = \boldsymbol{0}$ ensures that the perturbation remains undetected, as it satisfies the power flow equations at honest nodes. As a result, the observable data under an FDI attack is given by:

$$\boldsymbol{z}_t = \boldsymbol{H}(\boldsymbol{x}_t + \delta\boldsymbol{x}_t) + \boldsymbol{\varepsilon}_t. \tag{22}$$

To detect these attacks, we construct a ground-truth label vector $\boldsymbol{y}$, where each entry is defined using the logit function:

$$\boldsymbol{y}_i = \text{logit}(\delta\boldsymbol{x}_i), \tag{23}$$

where $\text{logit}(\cdot)$ is an indicator function such that $[\boldsymbol{y}]_i = 1$ if $[\delta\boldsymbol{x}]_i \neq 0$, and $[\boldsymbol{y}]_i = 0$ otherwise.

## IV. Case Studies

In this section, we evaluate the performance of the transferable NGCN across different power systems by testing two applications: cyber-attack detection and FDI detection. These case studies aim to demonstrate the NGCN's ability to generalize across various grid configurations.

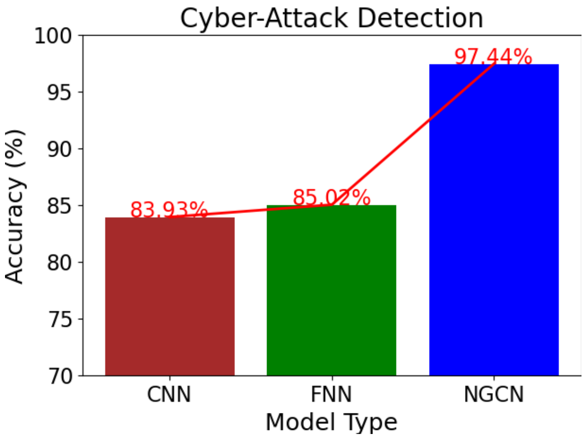

Fig. 3: Comparison of Cyber Attack Detection Using Spatiotemporal Features.

### A. Simulation Settings

For the simulation, we train a universal NGCN using data from the IEEE 17-bus, 18-bus, 22-bus, and 28-bus radial distribution systems. These systems are equipped with 5, 5, 7, and 8 smart inverters, respectively. The power demand data is sourced from real-world measurements in Texas [1], while photovoltaic (PV) power data for the smart inverters is obtained from the National Renewable Energy Laboratory (NREL) [2].

For the cyber-attack detection task, we utilize 25 computational nodal graphs, with each graph corresponding to a specific smart inverter. In these cases, cyber-attacks modify the VV-VW control functionalities of the smart inverters, leading to oscillatory events. We select $K = 3$ to define the neighborhood size for constructing the computational graphs. The input to the NGCN consists of time-series voltage phasor data, with $m_0 = 20$ channels, representing the multi-channel input format. The network architecture for cyber-attack detection comprises a NGCN layer for feature extraction, followed by two fully connected layers, each containing 256 neurons. For the FDI detection task, the goal is to identify anomalous data across all nodes in the IEEE 17-bus, 18-bus, 22-bus, and 28-bus systems. In this case, we focus exclusively on spatial information, excluding time-series data from the inputs. The network architecture for FDI detection mirrors that of the cyber-attack detection task, with a NGCN layer for spatial feature extraction, followed by two fully connected layers, each containing 256 neurons.

### B. Cyber-Attack Detection

The simulation results, shown in Fig. 3, illustrate a comparison of different detector models for cyber-attack detection using spatiotemporal features. Three models were evaluated: a feedforward neural network (FNN), a convolutional neural network (CNN), and a NGCN. As depicted, the NGCN significantly outperforms both the FNN and CNN, achieving an accuracy of 98.91%, compared to 78.92% for the FNN and 81.21% for the CNN. This demonstrates the effectiveness of incorporating spatiotemporal features into the NGCN for cyber-attack detection tasks.

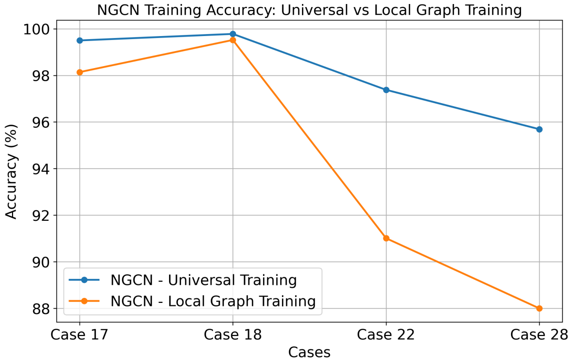

Fig. 4: Comparison of NGCN Training Accuracy: Universal vs. Local Graph Training.

Fig. 4 presents a performance comparison between universal NGCN training and local graph training. In the universal training approach, the model is trained using data from all four power systems and validated individually on each one. Conversely, the local training approach involves training and validating the model on each system's data independently. For the IEEE 22-bus system (referred to as "Case 22" in Fig. 4), universal NGCN training achieves an accuracy of 97.38%, significantly higher than the 91.01% achieved by local graph training. Similarly, in "Case 28" (the IEEE 28-bus system), universal training achieves 95.69%, outperforming the 88.00% of local training. These results highlight how universal training, by sampling across multiple power systems, enhances the model's generalization across different grid configurations.

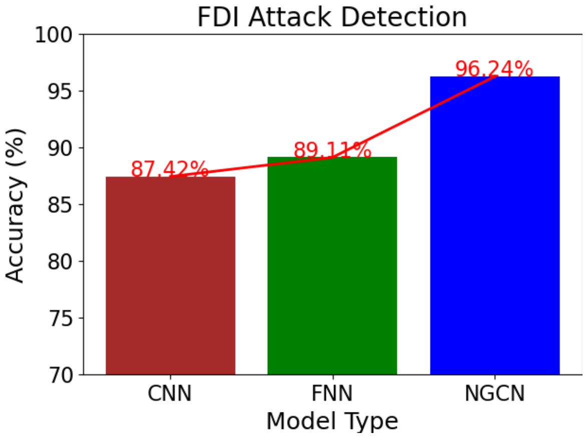

Fig. 5: Comparison of FDI Detection Using Only Spatial Features.

### C. FDI Detection

The results shown in Fig. 5 illustrate the performance of different models in detecting FDI attacks using spatial features. Although the accuracy of the NGCN decreases slightly to 95.21%, it still significantly outperforms both the FNN, which

[1] https://www.ercot.com/gridinfo/load/load_hist
[2] https://www.nrel.gov/grid/solar-power-data.html

achieves 74.27%, and the CNN, which reaches 77.56%. These findings underscore the effectiveness of the NGCN for FDI detection, even when relying solely on spatial data.

## V. Conclusions

In conclusion, this study demonstrates the effectiveness of the NGCN in addressing the limitations of traditional neural network detectors for power systems. By leveraging graph convolutions at each bus and its neighborhoods, the NGCN architecture enables the generalization of the detection model across multiple power systems, reducing the need for system-specific training and retraining in the face of system changes. The results from our experiments, particularly in the detection of cyber-attacks on smart inverters, show a significant improvement in detection accuracy, increasing from around 85% to approximately 97%. These findings highlight the NGCN's potential to provide a more robust and scalable solution for power system monitoring and cyber-attack detection.

## References

[1] S. Paul, F. Ding, K. Utkarsh, W. Liu, M. J. O'Malley, and J. Barnett, "On vulnerability and resilience of cyber-physical power systems: A review," *IEEE Systems Journal*, vol. 16, no. 2, pp. 2367–2378, 2021.

[2] T. Wu, A. Scaglione, and D. Arnold, "Reinforcement learning using physics inspired graph convolutional neural networks," in *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2022, pp. 1–8.

[3] P. Donti, A. Agarwal, N. V. Bedmutha, L. Pileggi, and J. Z. Kolter, "Adversarially robust learning for security-constrained optimal power flow," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 677–28 689, 2021.

[4] T. Wu, Y.-J. A. Zhang, and X. Tang, "Online detection of events with low-quality synchrophasor measurements based on $i$ forest," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 168–178, 2020.

[5] N. Tatipatri and S. Arun, "A comprehensive review on cyber-attacks in power systems: Impact analysis, detection and cyber security," *IEEE Access*, 2024.

[6] M. Ismail, M. F. Shaaban, M. Naidu, and E. Serpedin, "Deep learning detection of electricity theft cyber-attacks in renewable distributed generation," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3428–3437, 2020.

[7] T. Wu, I. L. Carreño, A. Scaglione, and D. Arnold, "Spatio-temporal graph convolutional neural networks for physics-aware grid learning algorithms," *IEEE Transactions on Smart Grid*, 2023.

[8] S. Wang, S. Bi, and Y.-J. A. Zhang, "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, 2020.

[9] W.-C. Hong, D.-R. Huang, C.-L. Chen, and J.-S. Lee, "Towards accurate and efficient classification of power system contingencies and cyber-attacks using recurrent neural networks," *IEEE Access*, vol. 8, pp. 123 297–123 309, 2020.

[10] D. Wilson, Y. Tang, J. Yan, and Z. Lu, "Deep learning-aided cyber-attack detection in power transmission systems," in *2018 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2018, pp. 1–5.

[11] Y. M. Khaw, A. A. Jahromi, M. F. Arani, S. Sanner, D. Kundur, and M. Kassouf, "A deep learning-based cyberattack detection system for transmission protective relays," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2554–2565, 2020.

[12] T. Wu, A. Scaglione, and D. Arnold, "Complex-value spatio-temporal graph convolutional neural networks and its applications to electric power systems ai," *IEEE Transactions on Smart Grid*, 2023.

[13] W. Xia, Y. Li, L. Yu, and D. He, "Locational detection of false data injection attacks in the edge space via hodge graph neural network for smart grids," *IEEE Transactions on Smart Grid*, 2024.

[14] R. Ramakrishna and A. Scaglione, "Grid-graph signal processing (grid-gsp): A graph signal processing framework for the power grid," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2725–2739, 2021.

[15] A. Campbell, H. Liu, A. Scaglione, and T. Wu, "A federated learning approach for graph convolutional neural networks," in *2024 IEEE 13rd Sensor Array and Multichannel Signal Processing Workshop (SAM)*. IEEE, 2024, pp. 1–5.