Complex-Value Spatiotemporal Graph Convolutional Neural Networks and Its Applications to Electric Power Systems AI

Tong Wu®, Member, IEEE, Anna Scaglione®, Fellow, IEEE, and Daniel Arnold®, Member, IEEE

Abstract—The effective representation, processing, analysis, and visualization of large-scale structured data over graphs, especially power grids, are gaining a lot of attention. So far most of the literature considered exclusively real-valued graph signals. However, graph signals are often sparse in the Fourier domain, and more informative and compact representations for them can be obtained using the complex envelope of their spectral components, as opposed to the original real-valued signals. This is the case for the AC carrier, and motivates its complex phasor representation. Focusing on applications in power systems, in this paper we generalize graph convolutional neural networks (GCN) to the complex domain, incorporating a complex-valued graph shift operators (GSO) based on the admittance matrix of power grids in the definition of graph filters (GF) and process complex-valued voltage phasor graph signals (GS). The theory developed is generalized to handle spatio-temporal complex network processes. We prove that complex-valued GCNs can be stable with respect to perturbations of the underlying graph support, by bounding of the error propagation through multiple NN layers. The paper showcases the benefits of complex GCN relative to several benchmarks, in power grid state forecasting and cyber-attack detection and localization.

Index Terms—Complex-valued graph neural networks, power system state forecasting, false data localization.

I. INTRODUCTION

In machine learning (ML) applications in which signals have a sparse spectrum, the best representation for signals is through their complex envelopes. This explains the popularity of complex-valued neural networks (Cplx-NN), introduced in the seminal paper [1], in a number of different domains, such as physical layer communications, image processing, automatic control etc. (see [2], [3] for a review of the theory behind Cplx-NN and [4] for a survey of its main applications). Motivated primarily by the application of artificial intelligence (AI) in electric power systems, the overarching goal of this paper is to extend the benefits of Cplx-NN to the analysis of complex graph signals, first introducing Complex Graph Convolutional Neural Networks (Cplx-GCNs), and then investigating their potential benefits in processing electric power systems measurements. In fact, in power systems the AC voltage at each node (called bus) is concentrated around 60 or 50 Hz and the vector of complex envelopes of the voltage signals (called phasors) represents the state of the electric power network. The abundance of high-quality estimates of the voltage and current phasors acquired using phasor measurement units (PMUs) has already spurred interest in complex graph signal processing (GSP) as a framework to process them and interpret their properties [5]. Complex GSP is the most natural framework to analyze the power system state because it has a physical interpretation rooted in Ohm’s law [5].

GSP is a vibrant branch of signal processing research that seeks to extend the concepts of digital signal processing (DSP), and Fourier analysis in particular, for data supported on graphs [6]. In the field of GSP, a graph signal (GS) refers to a vector that is indexed by the nodes of a weighted graph. This vector represents both the data, i.e., the node attributes, and the underlying structure, i.e., the edge attributes. To apply GSP methods, it is essential to define Graph Shift Operator (GSO). The majority of GSP-based algorithms use real-valued GSOs and consider real-valued graph signals (GS). Several surveys, including [7], [8], discuss this in detail. Once the GSO is selected, graph-filters can be defined. The Chebyshev filter is the most widely used graph filter model [6], [9]. It is important to note that the concepts presented in the aforementioned papers [7], [8], [9] apply solely to real-valued signals and cannot be directly extended to complex-valued signal systems such as power grids, which require the use of complex-valued GSOs.

While complex-valued GSP has seen applications in areas such as wireless communication networks [10], sensor networks [11] and the power grid [5], it has not received as much attention as real-valued GSP. However, in power systems, complex GSP is the most natural framework [5], [12].

The name comes from the fact that originally the GSO was a generalization of the z variable, corresponding to a time shift in the z transform; however, the definition often selects the Laplacian of the network graph, which is a graph signal differential operator.
Using the complex system matrix as the GSO has a physical interpretation rooted in Ohm’s law, making it an ideal choice for power systems [5], [12]. The caveat is that the GSO, which is the admittance matrix, is only symmetric, not conjugate symmetric, which, as we later discuss in Section II-B, requires some special care.

GSP algorithms that rely only on linear models, such as the ones studied in [5], [12] have limited representation capability. Interestingly, the first instance of GCN architecture appeared well ahead of the development of GSP [13]. The early models of GCN can be interpreted as a special case of the more general design introduced in [14], where the authors extend the Convolutional NN (CNN) model using graph filters. To process time-series of graph signals, whose samples are not independent and identically distributed (i.i.d.), the most effective architectures are Spatio-Temporal versions of the architecture, such as Spatio-Temporal GCN (STGCN) (see, e.g., [15], [16] which are early works on the subject) and Graph Recursive NN (GRN) (first proposed in [17]). In a nutshell, their design includes feed-forward and feedback graph-temporal filters in each layer. A thorough analysis of the stability of these designs is in [18], which inspired the stability analysis in this paper. Real-valued GCNs have showcased strong generalization capability in high-dimensional state spaces, learning complicated tasks with lower prior knowledge [19].

To the best of our knowledge, thus far, GCNs (and their variants) have been studied and applied only in the real domain (see, e.g., [19] for a review). The construction of complex GCN we study in this paper follows exactly the same logic of cascading layers of complex graph-temporal filters with nonlinear activation functions for complex data. Complex-valued Spatio-temporal GCNs (Cplx-STGCNs) are applicable not only to power systems, but to any networked system where nodal signals and their interactions can be modeled effectively as a vector of envelopes for its spectral components. Prior to summarizing our contributions, next we provide a brief review of the literature on real-valued GCNs for power systems applications, including the ones that we consider in our experiments to test the Cplx-STGCN performance.

A. Related Works

1) GCN Applications in Power Systems: Several papers have already applied real-valued GCN to power systems’ data analysis and management [20]. Applications include, for example, PMU missing data recovery [21], power system state estimation [16], optimal power flow calculation [22], detection and localization of stealth false data injection (FDI) attacks [23], [24], synthetic feeder generation [25], and power system fault localization [26], [27] to name a few. The two applications we choose to test numerically Cplx-GCN architectures are that of detection and localization of FDI attacks and power systems state estimation and forecasting (PSSE and PSSF). We note that PSSF has so far been pursued via single-hidden-layers NNs [28], [29], and further investigated by the Recurrent neural networks in [30] and Graph Recurrent neural networks [31]. The state-of-art neural network algorithms for FDI attack detection have been pursued by the Chebyshev GCN [23], CNN [32] and RNN [33] (see, e.g., [34] for a review).

All works on real-GCN for power systems have in common the following limitations: 1) they ignore the correlation among real and imaginary parts of power systems signals and use real GSO; 2) they do not consider temporal correlation of voltage phasors samples.

2) GCN Transfer Learning: The GCN is capable of processing various topologies. However, the optimal parameters for the GCN can differ based on these varying topologies [35], [36], [37]. This difference necessitates the retraining of the GCN when the network alters, which contributes to an increased computational complexity. Ding et al. [35] discuss the complexities in efficiently retraining the GCN due to the dynamic nature of real-world graphs. Chen et al. [36] suggest model retraining to adapt to evolving traffic networks and varied user social circles. However, Wang et al. [37] emphasize that the learning process for updated representations, which involves understanding both new and existing patterns, necessitates continual retraining of the GCN whenever the network changes, which results in high computational complexity. Additionally, [38] underscores the diminished performance in power system voltage stability assessments without transfer learning.

B. Contributions

The aim of this paper is to establish the framework of complex-valued STGCN and elucidate how they can be applied to power grid signals inference problems. Our main contributions are as follows:

- We generalize the GCN in complex domain, with complex-valued GSO and complex-valued graph signals for electric power systems AI. Furthermore, we evolve the GCN to manage streaming data via the Cplx-STGCN architecture. The differentiation between complex-valued GCN and its real-valued counterpart is pronounced, especially considering the intricate interplay between the real and imaginary components of GCN weights, GSO, and GS.

- We provide analytical bounds for the impact of perturbations in the GSO, and derive bounds for how the error propagates through the multi-layer GCN structure. This furnishes a theoretical foundation for the transfer learning capability of the proposed Cplx-STGCN, enabling its application in new environments without the necessity for retraining.

- We show how to apply correctly this framework to power systems. This entails choosing as input the voltage phasors signals, the admittance matrix as our GSO, and the Graph Fourier basis. As a result, our Cplx-STGCN is intrinsically physics-aware, specifically designed for power systems.

- Demonstrative results reveal the superiority of our approach over existing methods, both in detecting and localizing FDI attacks, and in PSSE and PSSF performance metrics. These outcomes underscore the
potential and benefits of the Cplx-GCN framework in power system contexts.

The paper organization is as follows. Section II contains a brief review of the fundamental concepts of GSP, and the definition of the physics inspired complex GSO for grid graph signals. We introduce our proposed graph neural network architectures, and study their sensitivity to GSO variations and errors in Section III. Section IV presents the spatiotemporal graph convolutional neural networks. In Section V, we describe two applications of our proposed Cplx-STGCN frameworks, and the numerical test results are presented in Section VI. Finally, in Section VII, we draw some conclusions and errors in Section III. Section IV presents the spatiotemporal graph convolutional neural networks. In Section V, we describe two applications of our proposed Cplx-STGCN frameworks, and the numerical test results are presented in Section VI. Finally, in Section VII, we draw some conclusions based on the results obtained from our proposed Cplx-STGCN frameworks and discuss potential future research directions.

II. Complex-Valued Graph Convolution Neural Network

A. Power System Notations

Electric grid networks have an associated undirected weighted graph, denoted as \( G(V, E) \). In this graph, the nodes correspond to the buses, and the transmission lines are represented as edges. The relationships between the current and voltage phasors for the network are described by Kichhoff’s and Ohm’s laws, which can be written as follows:

\[
i = \mathbf{Y} \mathbf{v}, \quad v_n = |v_n| e^{j\theta_n}, \quad i_n = |i_n| e^{j\phi_n}, \quad \forall n \in V.
\] (1)

Here, \( \mathbf{v} \in \mathbb{C}^{|V| \times 1} \) and \( |\mathbf{v}| \in \mathbb{R}^{|V| \times 1}_+ \) represent the vectors of bus voltage phasors and magnitudes, respectively, and \( i \in \mathbb{C}^{|V| \times 1} \) and \( |i| \in \mathbb{R}^{|V| \times 1}_+ \) are the vectors of net bus current phasors and magnitudes, respectively. The imaginary unit is denoted by \( j = \sqrt{-1} \). The analogous of an integrator for GSP is the inverse of the GSO. This operator smoothens the graph signal over neighborhoods, and behaves as a low-pass graph filter (c.f. [5]). Hence, Ohm’s law suggests that we can interpret voltages as the output low-pass filter, such that \( \mathbf{v} = \mathbf{Y}^{-1} \mathbf{i} \).

B. Preliminaries: Grid-Graph Signal Processing

Our study considers the power grid’s voltage phasors at each bus as the graph signal \( \mathbf{x} \in \mathbb{R}^{|V|} \). The GSO \( \mathbf{S} \in \mathbb{R}^{|V| \times |V|} \) is a matrix that combines the values of a graph’s neighboring nodes linearly [5], [7]. Our focus is on complex symmetric GSOs, i.e., matrices that satisfy \( \mathbf{S} = \mathbf{S}^\top \). Such GSOs are suitable for power grid applications, where \( \mathbf{S} = \mathbf{Y} \). In [7], a graph filter is defined as a linear matrix operator \( \mathcal{H}(\mathbf{S}) \) that operates on graph signals as follows:

\[
\mathbf{w} = \mathcal{H}(\mathbf{S})\mathbf{x}, \quad \mathcal{H}(\mathbf{S}) = \sum_{k=0}^{K-1} h_k \mathbf{S}^k.
\] (2)

where the matrix polynomial \( \mathcal{H}(\mathbf{S}) \) can have an infinite graph filter order \( K \). To perform eigenvalue decomposition, we can represent the GSO matrix \( \mathbf{S} = \mathbf{U} \Lambda \mathbf{U}^\top \), where \( \mathbf{U} \) is the eigenvector matrix, and \( \Lambda \) is the diagonal matrix with eigenvalues \( \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{|V|} \). Since the GSO \( \mathbf{S} \) is symmetric, \( \mathbf{U} \) is unitary. The equivalent of the Fourier basis for a graph signal is the Graph Fourier Transform (GFT). The GFT basis is \( \mathbf{U} \), and the GFT of a graph signal \( \mathbf{x} \) is represented by \( \tilde{\mathbf{x}} = \mathbf{U}^\top \mathbf{x} \). The eigenvalues \( \lambda_\ell, \ell = 1, \ldots, |V| \) correspond to the graph frequencies [5], [7], [8]. As illustrated in Fig. 1, when dealing with time series of graph signals \( \{\mathbf{x}_t\}_{t \geq 0} \) that have non-i.i.d. samples, using graph temporal filters is more appropriate:

\[
\mathbf{w}_t = \sum_{t=0}^{t} \mathcal{H}_{t-r}(\mathbf{S})\mathbf{x}_t, \quad \mathcal{H}(\mathbf{S}) = \sum_{k=0}^{K-1} h_k \mathbf{S}^k.
\] (3)

We can further utilize DSP techniques introduced in [7] to analyze graph temporal filters, by defining a joint transform that combines the Graph Fourier Transform (GFT) and the z-transform.

\[
\mathbf{X}(z) = \sum_{t=0}^{K-1} \mathbf{x}_t z^{-t}, \quad \tilde{\mathbf{X}}(z) = \mathbf{U}^\top \mathbf{X}(z),
\] (4)

where \( \mathbf{k} \) is the length of the graph signal time series. In [5], the graph-temporal transfer function and input-output relationship in the joint GFT-z-domain are:

\[
\mathbb{H}(\mathbf{A}, z) = \sum_{t=0}^{K-1} \sum_{k=0}^{K-1} h_{k,t} \Lambda^{k-t}, \quad \tilde{\mathbf{W}}(z) = \mathbb{H}(\mathbf{A}, z) \tilde{\mathbf{X}}(z),
\] (5)

where the matrix \( \mathbb{H}(\mathbf{A}, z) \) is diagonal. In the context of GCNs, the coefficients \( h_{k,t} \) can be trained [9]. The distinctions between real and complex GSP have been analyzed in [39]. While complex-GSP is articulated as a linear operator, its limited capability to address nuanced challenges becomes evident. This drives our endeavor to incorporate a nonlinear activation function into GSP, laying the foundation for a perceptron. From this, we cultivate the advanced Cplx-GCN, optimized for inference and control tasks.
C. Complex-Valued Graph Convolution Neural Network

In Fig. 2, the graph neural network perceptron based on (2) is expressed as:

$$\tilde{w} = \sigma[w] = \sigma\left[ \sum_{k=0}^{K-1} h_k S^k x \right]$$

(6)

where $x \in \mathbb{C}^{|V|}$, $h_k \in \mathbb{C}$, $S^k \in \mathbb{C}^{|V| \times |V|}$ and $w \in \mathbb{C}^{|V|}$ are the complex values. Since, $\sigma(\cdot)$ takes as input complex values, there is significant flexibility in defining this operator in the complex plane. We use CReLU to refer to the complex rectified linear unit activation function, which applies ReLU activation independently to the real and imaginary components of a neuron, i.e.:

$$\text{CReLU}(w) = \text{ReLU}(\Re(w)) + j \text{ReLU}(\Im(w))$$

(7)

This is a popular choice because the CReLU satisfies the Cauchy-Riemann equations if both the real and imaginary parts are either strictly positive or strictly negative $[2]$. Empirically, we found this choice to be preferable to other options proposed in the literature. In the selection of GCN one of the key steps is recognizing what is the correct GSO that defines the graph filter, since its choice affects the feature extraction capabilities of the GCN. In prior work $[5]$, we have argued and demonstrated that, for shallow methods, the admittance matrix $Y$ is the best GSO $S$, based on Ohm’s law and the properties of voltage phasors. This motivated the investigation in this work.

Spatio-Temporal GCN can be viewed as a particular instance of multi-feature GCN. To elaborate, suppose we have $X = [x^1, \ldots, x^F]$ representing our input features and $W = [w^1, \ldots, w^G]$ as the multi-channel outputs. Here, $F$ indicates the total number of input features, while $G$ denotes the total number of output channels. The functioning of a multi-feature GCN is one way of mixing and separating the real and imaginary variables is

$$\tilde{S}(w) = \sum_{k=0}^{K-1} h_k S^k \tilde{x}$$

(8)

where the coefficient matrix, denoted as $H_k$, has dimensions $G \times F$ and its individual entries can be expressed as $[H_k]_{fg} = h_k\gamma(f,g)$. Additionally, the term $H \ast G$ represents a graph convolution operation, which draws inspiration from the principle of spectral graph convolution.

1) Discussion About Cplx-GCN vs Real-GCN: Note that using real-GCN in lieu of complex GCN reduces significantly the number of trainable parameters. Specifically, in terms of Cplx-GCN, one way of mixing and separating the real and imaginary variables is

$$\begin{bmatrix} \Re(w) \\ \Im(w) \end{bmatrix} = \sum_{k=0}^{K-1} h_k \begin{bmatrix} \Re(S) \\ -\Im(S) \end{bmatrix} \begin{bmatrix} 1 & \Im(S) \\ -\Re(S) & 1 \end{bmatrix}^{\dagger} \begin{bmatrix} \Re(x) \\ \Im(x) \end{bmatrix}$$

(9)

using an $h_k$ which is a real scalar in the decoupled model. This removes the imaginary part of $h_k$, reducing the neural network function approximation capability. This is why, when such GCN methods are applied to voltage phasor signals, the resulting trained models under-perform the complex ones in inference and control tasks.

III. ANALYSIS OF THE CPLX-GCN SENSITIVITY

Particularly for power systems, it is quite common to incur in sparse system changes, due to switching or changes of line impedance. It is, therefore, of interest to understand how sensitive is the response of the Cplx-GCN to changes in the parameters. For the case where the changes in the GSO are known it is to study how parameters trained on a different GSO will respond. We refer to this as the transfer learning error. Next we provide insights on the impact of perturbations in the GSO on the end-to-end Cplx-GCN mapping. We improve substantially the results of $[18]$ which are exclusively for real GCN, do not consider the end to end distortion, and also rely on restrictive assumptions about the structure of the perturbation that we could not justify in our practical setting.

In the following we denote by $\sigma_{\text{max}}(A)$ its largest singular value of matrix $A$. We can prove the following bound:

Theorem 1: Consider graph filter $h = [h_0, \ldots, h_K]$ along with shift operator $S$ having $|V|$ nodes. Let $E \in \mathbb{C}^{|V| \times |V|}$ denote the matrix perturbation with $\|E\| \leq \epsilon$, and $\tilde{S} = S + E$. Let us denote by $\tilde{h} = [\tilde{h}_0, \ldots, \tilde{h}_K]$ the graph filter parameters obtained training the network using $\tilde{S}$ as GSO. Let $\hat{\mathcal{H}}(\tilde{S}) = \sum_{k=0}^{K} \tilde{h}_k \tilde{S}^k$ and the Cplx-GCN layer with the original filter coefficients obtained training with GSO $S$, albeit using the perturbed GSO, be $\mathcal{H}(S) = \sum_{k=0}^{K} h_k S^k$. Let us also define:

$$\gamma_1 \triangleq \max \left( 1, (\sigma_{\text{max}}(S) + \epsilon)^K \right)$$

(10)

The following bound holds:

$$\sigma_{\text{max}} \left( \hat{\mathcal{H}}(\tilde{S}) - \mathcal{H}(S) \right) \leq \gamma_1 \| \tilde{h} - h \|_1.$$  

(11)

Proof: The proof is in Appendix A.

| Theorem 2: Let $E \in \mathbb{C}^{|V| \times |V|}$ be the matrix perturbation with $\|E\| \leq \epsilon$, and $\tilde{S} = S + E$ and

$$\gamma_2 \triangleq \max_{1 \leq k \leq K} |h_k|(1 + \sigma_{\text{max}}(S))^K.$$  

(12)

Assume that the Cplx-GCN layer used is $\mathcal{H}(\tilde{S}) = \sum_{k=0}^{K} \tilde{h}_k \tilde{S}^k$ where the coefficients are the same as those obtained by training in the original Cplx-GCN layer that is defined as $\mathcal{H}(S) = \sum_{k=0}^{K} h_k S^k$. Then, the following bound holds:

$$\sigma_{\text{max}} \left( \mathcal{H}(\tilde{S}) - \mathcal{H}(S) \right) \leq \gamma_2 \frac{\epsilon(1 - \epsilon^K)}{1 - \epsilon}.$$  

(13)

Proof: The proof is in Appendix B.
This theorem clearly shows that for a small perturbation in the GSO one should get a similar response to the parameters, which suggests that for small GSO perturbations it is reasonable to use the same parameters and transfer the learning done to the new case. Different from [18], our work assumes the graph perturbation as \( E \), where \( \| E \| \leq \epsilon \). The perturbed shift operator, in our context, is written as \( \hat{S} = S + E \). This assumption mirrors practical considerations, especially in the context of power systems. Besides, our analysis of the error bound is equally applicable to real-value GCN. This is because our bound hinges on the largest singular value of the graph filter, which remains a real number for both real-valued and complex-valued matrices.

Next we bound the difference between the two outputs of the retrained network with the perturbed GSO and the original network, in other words the bound on the norm of the output difference, when the GCN perturbation consists of the perturbations of both parameters \( h_k \) and GSO.

**Corollary 1 (The Bound of Cplx-GCN Perturbation):** Let the retrained Cplx-GCN layer be \( \hat{h}(\hat{S}) = \sum_{k=0}^{K} \hat{h}_k \hat{S}^k \) and the original one be \( \hat{h}(S) = \sum_{k=0}^{K} h_k S^k \) with the new GSO \( \hat{S} \). Then:

\[
\| (\hat{h}(\hat{S}) - h(S))x \| \leq \left( \gamma_1 \| \hat{h} - h \|_1 + \gamma_2 \frac{(1 - e^K)}{1 - e} \right) \| x \| \tag{14}
\]

where the parameters in the right-hand side were defined in Theorem 1 and 2.

The proof is obvious because of the triangle inequality.

**Remark 1:** If the GSO \( S \) is normalized by \( \sigma_{\text{max}}(S) \), we can further bound the result in (11) as follows:

\[
\gamma_1 \| \hat{h} - h \|_1 \leq (1 + \epsilon)^K \| \hat{h} - h \|_1 \tag{15}
\]

\( \gamma_1 \) amplifies the sensitivity exponentially with \( K \) while the effect of the parameters difference increases linearly with \( K \). When \( K \to \infty \), we have

\[
\lim_{K \to \infty} (1 + \epsilon)^K \| \hat{h} - h \|_1 = e^{K} \| \hat{h} - h \|_1 \tag{16}
\]

Moreover, if \( \epsilon \) is approaching 0, i.e., \( \epsilon \to 0 \), the error can be further bounded by

\[
\lim_{K \to \infty, \epsilon \to 0} (1 + \epsilon)^K \| \hat{h} - h \|_1 = e \| \hat{h} - h \|_1 \tag{17}
\]

**Remark 2:** The interpretation of Theorem 2 is more straightforward. Its dependence on \( \epsilon \) is clear, and it is also clear that it exponentially increases with \( K \) with a rate \( (1 + \sigma_{\text{max}}(S))^{-K} \). So, when using an incorrect GSO in the network one would want to make sure that \( \epsilon \ll (1 + \sigma_{\text{max}}(S))^{-K} \). Here, \( \sigma_{\text{max}}(S) \) and \( K \) are clearly negatively impacting the sensitivity.

### IV. Spatiotemporal Graph Convolutional Neural Network

The time-varying voltage phasors in power systems are a graph signal time series. To capture both spatial and temporal information, we can extend to the complex domain the Conv1D Graph Convolutional Neural Network architecture [16, 40] in a relatively straightforward manner. Although RNNs are commonly used in time-series analysis, and have been extended to process graph signals as well, they are unsuitable for power systems due to slow convergence, complex gating, and slow response [16]. CNNs, in contrast, have a simpler structure and faster training. The temporal convolutional layer in Fig. 3 employs a 1-D CNN with a kernel of width \( T \) and \( K_t \) output channels. The input \( X \in \mathbb{C}^{[V] \times T} \) is mapped to an output graph signal \( \tilde{X} \in \mathbb{C}^{[V] \times K_t} \) using the convolutional kernel \( \Gamma \in \mathbb{C}^{T \times K_t} \). Therefore, the temporal convolution is defined as,

\[
\tilde{X} = \Gamma \ast_T X, \tag{18}
\]

where each column of \( [\tilde{X}]_t \) is defined as \( \tilde{x}_t, \tau = 0, 1, \ldots, K_t - 1 \). After the temporal convolutional layer, we are ready to put \( \tilde{X} \) into the spatial layer. Based on (5), we can design the following transfer functions and neuron:

\[
\mathbb{H}(S, z) = \sum_{t=0}^{K_t-1} \sum_{k=0}^{K-1} h_{k,\tau} S_k^T z^{-T},
\]

\[
\tilde{w}_t = \sigma [w_t] = \sigma \left[ \sum_{k=0}^{K_t-1} \sum_{\tau=0}^{K_t-1} h_{k,\tau} S_k^T x_{t-\tau} \right]. \tag{19}
\]

Consequently, the graph signal \( \tilde{w}_t \) from the spatial feature extraction layer (see Fig. 3) is:

\[
\tilde{w}_t = \text{CReLU} \left[ \sum_{k=0}^{K_t-1} \sum_{\tau=0}^{K_t-1} h_{k,\tau} S_k^T \tilde{x}_{t-\tau} \right], \tag{20}
\]

By combining the temporal and spatial convolutions at each layer, the multiple output channels of the Cplx-STGCN layer (\( \ell = 1 \)) are expressed as

\[
\tilde{w}_{t,\ell=1} = \text{CReLU}(\mathbb{H} \ast g (\Gamma \ast_T X_t)), \tag{21}
\]
where $H$ and $\Gamma$ are the trainable parameters. The first layer, defined by (21), serves as the feature extraction layer.

The subsequent hidden layers $\ell \in 1, \ldots, L - 1$ consist of a complex-valued fully connected neural network.

$$\mathbf{W}_{l, l+1} = \text{ReLU}(\Theta^\text{cplx} \mathbf{W}_{l, l})$$

For the output layer $L$, we transform the complex tensor $\hat{\mathbf{W}}_{l, L}$ into a real tensor, and then map it to the labels (or regression targets):

regression: $y' = \tanh(\Theta^\text{cplx} \ast \hat{\mathbf{W}}_{l, L})$,

classification: $y^c = \text{sigmoid}(\Theta^\text{cplx} \hat{\mathbf{W}}_{l, L})$.

where $y'$ and $y^c$ denote the complex regression targets and the real classification labels, respectively. Besides, $\Theta^\text{cplx}_L$ and $\Theta^\text{cplx}_{l-1}$ denote the complex and real trainable matrix. Then, we define the multi-layer Cplx-STGCN learning function as:

$$y^c_\ell = \Phi^c(x_\ell, S, \theta^c),$$

where $\theta^c = \{(\Theta^\text{cplx}_l, H, \Gamma)\}$ for $\ell = 1, \ldots, L$ and $\theta^c = \{(\Theta^\text{cplx}_l, \Theta^\text{real}_l, H, \Gamma)\}$ for $\ell = 1, \ldots, L-1$ represent the trainable parameters and $X_\ell = [x_{t-\ell+1}, \ldots, x_t]$. We have omitted the bias term to simplify the notation, but it is included in the trainable model.

In the following, we further investigate how the multilayer neural networks propagate the error due to the changes of parameter and GSO $S$.

**Lemma 1:** Assume a neural network consists of one Cplx-GCN layer and one Cplx-FNN layer with trainable parameters $\Theta^\text{cplx}$, denoted by $y = \Phi(x, S, \theta)$. The retrained neural network has the new GSO $S$ and the new parameters $\hat{\theta}$, denoted by $\hat{y} = \Phi(x, \hat{S}, \hat{\theta})$. We define the perturbation of the Cplx-FNN layer is $\|\Theta^\text{cplx} - \hat{\Theta}^\text{cplx}\|_2 \leq \delta_w$. Then, the distance between $y$ and $\hat{y}$ is bounded by:

$$\|y - \hat{y}\| \leq (\delta_w \ast \Psi_1 + \sigma_{\max}(\Theta^\text{cplx}) \ast \Psi_2) \|x\|$$

where $\Psi_1$ and $\Psi_2$ are given by:

$$\Psi_1 = \gamma_1 \|\hat{h}\|_1$$

$$\Psi_2 = \left[\gamma_1 \|\hat{h} - h\|_1 + \gamma_2 \frac{1 - \epsilon^{K}}{1 - \epsilon}\right]$$

**Proof:** The proof is in Appendix C.

**Corollary 2:** We generalize the bound into the multilayer neural networks, including one Cplx-GCN feature extraction layer and $L$ Cplx-FNNs as

$$\|y - \hat{y}\| \leq \Delta_L, \quad \Delta_L = \sigma_{\max}(\Theta^\text{cplx}_L) \Delta_{L-1}$$

$$+ \delta_w \prod_{\ell=1}^{L-1} \sigma_{\max}(\Theta^\text{cplx}_\ell) \Psi_1.$$  

where $\Delta_1$ is defined in (25).

The proof is obvious according to the norm triangle inequality so that we omit the proof. From Corollary 2, we can observe that if the retrained neural networks have small changes to make $\delta_w \approx 0$, we could find the dominant part, i.e., $\Delta_L \approx \sigma_{\max}(\Theta^\text{cplx}_L) \Delta_{L-1}$. Therefore, the error of the GCN perturbation is propagated by the largest singular values of original Cplx-FNN weight matrices.

V. APPLICATIONS OF CPLX-STGCN

A. Power System State Estimation and Forecasting

Measurements in power systems are relatively sparse. In this subsection, we propose a power systems state estimation algorithm that can use limited measurements to estimate the current and future state at all buses. Consider the vector $x_{\mathcal{A}}$ (time index $t$ ignored for simplicity) composed of measurements from the subset of buses $\mathcal{A} \subset \mathcal{V}$ equipped with sensors. Let $U_{\mathcal{A}}$ denote the GFT basis corresponding to the $k$ dominant graph frequencies in the voltage phasor GFT spectrum. The set of measurement buses corresponding to the lowest graph frequencies can be identified through Ohm’s law [5] and correspond to the least significant eigenvectors of the system matrix. To obtain an optimal subset of measurement buses with minimal correlation among their corresponding rows in $U_{\mathcal{A}}$, one can maximize the smallest singular value of the matrix $F_{\mathcal{A}}U_{\mathcal{A}}$, where $F_{\mathcal{A}} = \mathcal{Q}_{\mathcal{A}} \mathcal{Q}_{\mathcal{A}}^*$ and $\mathcal{Q}_{\mathcal{A}}$ contains columns that are the coordinate vectors pointing to each vertex/node in $\mathcal{A}$. This method is described in [41].

Once the optimal subset of measurement buses $\mathcal{A}$ has been determined using the method mentioned above, voltage phasor measurements can be obtained at these locations using PMUs. Note that this PMU localization selection significantly impacts the initial estimation of the voltage phasor, rather than the final forecasting of Cplx-STGCN. Let $\mathcal{U}$ denote the set of unavailable measurements. Eq. (1) can be reformulated to construct the vector observation $z_t$ by using the set of available measurements $\mathcal{A}$ and the set of unavailable ones $\mathcal{U}$.

$$\begin{pmatrix} \hat{i}_{A} \\ \hat{v}_{A} \end{pmatrix} = \begin{pmatrix} Y_{\mathcal{A}} & Y_{\mathcal{A}}^H \end{pmatrix} \begin{pmatrix} \mathcal{A} \vert \mathcal{U} \end{pmatrix} + \epsilon_t,$$

where $\epsilon_t$ is a vector of measurement noise. The prediction of voltage phasors is a common problem in time-series forecasting. Note that we did not use $H$ for graph convolution. Instead, our sole matrix for graph convolution is the admittance matrix $Y$. Specifically, the task involves predicting the most probable voltage phasor measurements in the next $H$ time steps, given the previous $T$ measurements that have been sub-sampled $[x_t]_{A}$ as

$$x^*_t = \arg \max_{x_t} P(x_{t+H} \mid [x_{t-1}]_{A}, \ldots, [x_t]_{A})$$

where observation vectors $[x_{t-1}]_{A}, \ldots, [x_{t}]_{A}$ record the historical observations of $[A]$ measurements.

1) Methodology: The initial step of the algorithm involves recovering the voltage phasors $x_t$ from $z_t = \hat{z}_t = \hat{v}_t = \hat{v}_t$. This is achieved by solving a regularized least square problem:

$$\min_{x_t} \|z_t - Hx_t\|_2^2 + \mu_1(x_t^H S x_t)$$
where $\mu_1$ is positive. A closed-form solution for (30) is obtained by solving the following equation:

$$\hat{x}_t = \left( H^H H + \mu_1 S \right)^\dagger H^H z_t, \quad (31)$$

where the estimated voltage phasor at time $t$ is denoted as $\hat{x}_t$, and $(\cdot)^\dagger$ represents the pseudo-inverse operation. The algorithm step are as follows:

1. Collect $T$ historical measurements $z_{t-T+1}, \ldots, z_t$.
2. Use (31) to estimate the full observations as $\hat{X} = [\hat{x}_{t-T+1}, \ldots, \hat{x}_t]$.
3. Define the Cplx-STGCN loss function for voltage phasor prediction as follows:

$$L(\Phi, \theta) = \sum_t \left\{ \left\| y_t^c - x_{t+H} \right\|^2 + \mu_2 \left\| \hat{v}_{t+H,A} \hat{v}_{t+H,A}^T - [y_t^c (SY_t^c)^T]_A \right\|^2 \right\}, \quad (32)$$

where we aim to predict this target using the predicted target $y_t^c = \Phi^c(\hat{X}_t, S, \theta)$ as the approximation of the ground-truth regression target $v_{t+H}$. In (32), a regularization term is included to promote voltage phasor predictions that minimize the sum of the absolute value of the apparent power injections. It is worth noting that when the forecasting horizon $H$ is zero, the task corresponds to voltage phasor estimation, while $H \geq 1$ corresponds to voltage phasor forecasting.

After the training of the Cplx-STGCN model, the learned parameters $\theta^+$ can be used to forecast the outputs $x_{t+H}$ by feeding the inputs $[x_{t-T+1}, \ldots, x_t]_A$ that correspond to the observations at the respective times into the trained model, denoted as $\Phi^c(\hat{X}_t, S, \theta^+)$. The Cplx-STGCN, utilized for PSSE, PSSF, and FDI Localization, takes the estimated voltage phasors $v_t$ from all buses spanning the last $T$-windows as its input. This data from the last $T$-windows is represented by $\hat{X}_t = [\hat{x}_{t-T+1}, \ldots, \hat{x}_t]$, where $\hat{x}_t$ is the estimated voltage phasors $v_t$, as given by (31). The output of the Cplx-STGCN produces the regression target voltage phasors $v_{t+H}$, forecasting the next $H$ hours.

The proposed Cplx-STGCN not only estimates the current voltage magnitudes and phase angles, but it also predicts future values for these parameters in subsequent hours. Since future demands remain unknown, it is impossible to determine the voltage magnitude and phase angle through power flow calculations. Rather than attempting to forecast future demands and then compute the voltage phasors, a more effective strategy is to forecast the voltage phasors directly. Additionally, while traditional power system state estimation methods can be computationally intensive or suboptimal, our method predicts optimal voltage phasors with minimized fuel costs fast, underscoring the benefits of leveraging Cplx-STGCN.

**B. False Data Injection Detection and Localization**

The conventional task of FDI attack detection is a binary hypothesis testing problem where the null-hypothesis is no false data are present and the positive one is that some data are compromised. The localization of the false measurement (FDI localization problem), is the one of interest in this subsection. Such problem amounts to classifying each measurements into two categories (false data or not) and, thus, is a multi-label classification problem.

A type of attack known as a stealth attack [5] involves an attacker manipulating current and voltage phasor measurements on a subset of buses, denoted as $\mathcal{C}$, by introducing a perturbation:

$$\delta x_t^c = [\delta x^c_t \theta^c_{t+1}], \quad \text{such that} \quad Y_{PC} \delta x_C = 0, \mathcal{C} \subset \mathcal{A} \quad (33)$$

where $\mathcal{P}$ is the set of honest nodes. The details can be found in Appendix D. Consequently, the received data $z_t$ with FDI attack have a certain structure:

$$z_t = H(x_t + \delta x_t) + e_t. \quad (34)$$

1. **Methodology:** Then, the algorithm for FDI localization is summarized as:

   1. We obtain $T$ historical measurements $z_{t-T+1}, \ldots, z_t$ with FDI attacks based on (34).
   2. We construct the ground-truth label vector $y_t^c = \logit(\delta x_t)$, where $\logit(\cdot)$ is an indicator function that $[y_t^c]_i = 1$ if $[\delta x_t]_i \neq 0$, otherwise $[y_t^c]_i = 0$. Note that $\mathcal{C} \subset \mathcal{A}$ is the set of randomly sampled buses with the fixed number.
   3. We utilize (31) to obtain the estimated voltage phasors $\hat{x}_{t-T+1}, \ldots, \hat{x}_t$.
   4. The loss function of the Cplx-STGCN function for voltage phasor prediction is written as

   $$L(\Phi^c, \theta^c) = \sum_t \left\| \Phi^c(\hat{X}_t, S, \theta^c) - y_t^c \right\|^2. \quad (35)$$

With the trained $\Phi^c(\hat{X}_t, S, \theta^c)$, we could predict the multiple labels $y_t^c = \Phi^c(\hat{X}_t, S, \theta^c)$ when $z_{t-T+1}, \ldots, z_t$ are observed. Note that the outputs $y_t^c$ of Cplx-STGCN provide a label for each bus, each indicating the presence or absence of a false data injection attack.

**VI. Numerical Experiments**

The results presented in this section are obtained using the IEEE 118-bus test case with 118 nodes and 186 edges for PSSE and PSSF [42]. The 141-bus radial distribution system is considered for FDI localization [42]. Realistic demand profiles from the Texas grid are collected, and the voltage phasors are computed using Matpower [42]. In all simulations, we repeat the training and testing 8 times to report the average values.

**Cplx-STGCN Setting:** The architecture of the Cplx-STGCN for PSSF and FDI localization detection includes the Cplx-STGCN layer, the two-layer Cplx-NNs and one real-valued output layer. In the Cplx-STGCN layer, the Cplx-CNN (temporal convolution) output channel is 10 and the Cplx-GCN (spatial convolution) output channel is 10. Other Cplx-NNs have 512 neurons per layer. The order of GSO $K$ in the Cplx-GCN is 5.

**Baseline Setting:** For both application domains, our baseline algorithms encompass:

1. **FNN:** A fully-connected neural network (NN) that has four layers, each with 512 neurons. The inputs to this...
model are the real and imaginary parts of voltage phasors.

2) CplxFNN: A complex-valued fully-connected NN with four layers, each containing 512 neurons, using the complex voltage phasors as inputs.

3) GNN1st: A model that incorporates the first-layer GNN [43], followed by three layers of fully-connected NNs, utilizing voltage amplitudes and angles as input.

4) GCN_reim serves as a benchmark, employing a traditional real GCN, processing both real and imaginary parts of voltage phasor measurements as input.

5) Transformer [44] consists of three sub-encoder layers in both encoder and decoder sections. The anticipated number of features for encoder/decoder inputs is set at 512, which accommodates the real and imaginary parts of voltage phasor measurements as input.

6) GAT [45] and GTN [46]: They both utilize the GAT layer for feature extraction, having 10 input and output channels. Following this layer, three layers of fully-connected NNs, each with 512 neurons, process the voltage amplitudes and angles as inputs.

Specific algorithms tailored for PSSF include:

1) RNN [30]: This model integrates numerous measurements such as voltage magnitudes, active and reactive power injections, among others.2

2) GRNN [31]: A combination of a GNN1st layer and an LSTM layer, aiming to capture spatio-temporal correlations using voltage amplitudes and angles as input.

State-of-the-art algorithms for false data detection and localization encompass:

1) ChebyGCN [23]: This algorithm leverages the absolute values of the admittance matrix and takes active and reactive powers as inputs.

2) CNN [32]: Employs both line and bus measurements with a similar sensor count to our algorithm for fair comparison.

3) LSTM [33] and GSP [5]: Both consider the real and imaginary parts of voltage phasors or the full complex-valued voltage phasors, respectively, as inputs.

Sensor Placement: We take the eigendecomposition of \( Y \), and choose \(|k|\) = 40 graph frequency components and \(|A|\) = 60 sensors. The resulting of optimal sensor placement is shown in Table I as stated in Section VI-A. We selected the hyperparameters \( \mu_1 = 1e-6 \) and \( \mu_2 = 1e-4 \) for all benchmarks after conducting numerous simulations for hyperparameter tuning.

| Systems | Bus Name |
|---------|----------|
| ieee 118-bus | 14, 117, 72, 86, 43, 67, 99, 87, 16, 33, 112, 28, 98, 111, 53, 97, 42, 107, 48, 22, 46, 13, 24, 101, 44, 73, 109, 20, 91, 26, 84, 10, 1, 52, 76, 78, 115, 39, 74, 104, 93, 79, 35, 6, 18, 88, 60, 116, 55, 58, 68, 64, 7, 50, 103, 75, 78, 83, 69. |

2) Results: Figs. 4(a) and 4(c) compare Cplx-STGCN with various baselines on the IEEE 118-bus system experiment. Cplx-STGCN outperforms all baselines with a much smaller MSE of 0.00003517 for \( H = 0 \). Additionally, Cplx-STGCN predictions show much smaller MAPE, e.g., 0.5359% at \( H = 4 \), compared to other methods such as 2.0875% of GRNN and 4.3227% of FNN. Fig. 6 illustrates two examples showing the close resemblance between the predicted and ground-truth voltage phasors for \( H = 0, 1 \).

3) Transferability of Cplx-STGCN Regarding Topology Changes: In this section, we validate the transferability of the proposed Cplx-STGCN against the topology changes. Retraining a new model based on the new topology is time-consuming. To handle this problem, we keep the trained parameters unchanged and modify the GSO of Cplx-STGCN corresponding to the topology changes of power grids. In this simulation, we trip one line of power grids as the new topology. The results are shown in Fig. 4(b) and 4(d), which indicates that Cplx-STGCN performs well in the new topology. However, the fully-connected neural networks do not adapt to the new topology well. This is because the GSO captures the topology changes while the fully-connected neural networks do not have this property.

B. False Data Injection Localization

1) FDI Setting: The output of Cplx-STGCN is classified using a discrimination threshold of 0.5, which can be adjusted for sensitivity. The power grid network has 60 sensors (|A| = 60), and thus FDI localization involves 60 binary labels, resulting in a total of 2^{60} possible classes. |C| denotes the number of buses attacked, which ranges from 25 to 50. We also provide the accuracy, precision, recall, and F1 scores with biases for the Cplx-STGCN.

2) Results: The performance comparison for FDI localization is in Fig. 5(a). We can observe that Cplx-STGCN has much higher accuracy over other NNs and the GSP method that solves a LASSO problem to detect the false data entries. Other NNs tend to predict all ones or all zero vectors depending on whether |C| is large or small, respectively, while Cplx-STGCN captures the high-order spatial dependency and temporal correlation of voltage phasors, and thus achieve better performance as a result. Another observation is Cplx-STGCN has higher accuracy than LSTM [33], especially when |C| is large. Finally, Cplx-STGCN exhibits more stable performance with different values |C|.

3) FDI Localization for Hybrid Dataset: In the previous subsection, we have considered the attack hypotheses that for every \( z_i \), \forall t, FDI attacks are launched on some buses \( C \) of \( z_i \). In this subsection, we test the proposed algorithm
on the data set under both no-attack \( (H_0) \) and attack \( (H_1) \) hypotheses. Therefore, the received data \( z_t \) with FDI attack have the structure:

\[
z_t = \begin{cases} 
H_0 : H(x_t) + \epsilon_t, \\
H_1 : H(x_t + \delta x_t) + \epsilon_t.
\end{cases}
\] (36)

Fig. 5(b) shows the simulation results for the data set under both no-attack \( (H_0) \) and attack \( (H_1) \) hypotheses. It shows that the proposed algorithm has very high accuracy, e.g., 98.1043\%, compared with other methods, e.g., 88.5806\% of FNN and 91.7466\% of LSTM with \(|C| = 50\).

4) Other Metrics of FDI Localization: This section introduces additional evaluation metrics, namely precision, recall, and \( F_1 \) score. The terms \( TN, TP, FN, \) and \( FP \) are used to denote correctly classified unattacked buses, correctly predicted attacked buses, unattacked buses incorrectly predicted as attacked, and attacked buses incorrectly predicted as unattacked, respectively. Consequently, precision and recall are defined as follows:

\[
\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}. \quad (37)
\]

The \( F_1 \) score is defined as the geometric mean of precision and recall, and it balances the trade-off between the two by using the formula:

\[
F_1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (38)
\]

The results are shown in Tables II and III. It shows that Precision, Recall and \( F_1 \) Score are very high. It indicates that the false alarm rate of the Cplx-STGCN is small for both balanced and unbalanced data sets. We further present a precision comparison with other NNs in Fig. 7. Here, precision refers to the quantification of random errors, serving as a metric for statistical variability. The results demonstrate that the proposed cplxSTGCN significantly widens the precision.
C. Algorithm Validation

In this subsection, we evaluate the proposed STGCN based on computation time, increased GSO perturbations, sparse PMU measurements, and the efficacy of the spatial and temporal correlations within the Cplx-STGCN.

1) Computation Time: In Fig. 8, we provide an analysis contrasting the training durations of PSSE and PSSF using various neural network methodologies. The results underscore that RNN-based methods, which include RNN and GRNN, remain challenged by extensive iterative processes, complicated gate operations, and a delayed responsiveness to dynamic variations. In contrast, the proposed Cplx-STGCN not only offers faster training speeds but also features a more streamlined architecture, free from dependencies linked to earlier steps. Furthermore, our introduced GCN model adeptly addresses time convolutions via a CNN, tapping into the power of GPU acceleration during the training process. As for the complex-valued design, its parameter footprint is more compact compared to the real-valued neural network, suggesting a potential boost in computational efficiency.

2) More GSO Perturbation: We have included simulation outcomes for further perturbations to elucidate the transfer learning prowess of our proposed GCN. By selecting random 2, 3, and 4 line trippings in the IEEE 118-bus system as our GSO perturbations, we aim to demonstrate that increased line tripping corresponds to greater perturbation. The testing MSE for voltage phasors is displayed in Figs 9. These results emphasize that the Cplx-STGCN effectively adapts to new environments, maintaining commendable performance in the face of GSO perturbations.

3) Sparser PMU Measurements: In Fig. 10, we present results obtained using sparse PMU measurements. We perform an eigendecomposition of $Y$ and select $|K| = 20$ graph frequency components along with $|A| = 25$ sensors. The outcome of optimal sensor placement is detailed in Table IV.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
strategy, wherein the spatial correlations are encapsulated by the polynomial GSO in the Cplx-STGCN, enables the model to operate efficiently even with fewer PMU measurements (only 25 PMUs). This performance is only slightly inferior to that of the Cplx-STGCN utilizing 60 PMU measurements. In contrast, other methods exhibit significant degradation under sparse PMU measurement conditions.

4) Effectiveness of Spatial and Temporal Components:
In Fig. 11, we detail five supplementary simulations conducted for PSSE and PSSF. These are: (1) Real-valued FNN using shuffled samples, (2) Cplx-valued FNN employing shuffled samples, (3) Real-valued GCN with shuffled samples, (4) Cplx-valued GCN using shuffled samples, and (5) Cplx-valued STGCN with time-series samples. Our analysis reveals that the spatial aspect plays a pivotal role. For instance, transitioning from Cplx-FNN to Cplx-GCN resulted in substantial performance enhancements, whereas the jump from RealGCN to CplxGCN yielded relatively small improvements. While introducing temporal dynamics bolsters the GCN’s performance, it’s essential to recognize the significance of complex correlation. This is particularly true when enhancing the capability of the STGCN for managing complex-valued graph signals.

5) Texas 2000-Bus System:
To delve deeper into the application of our proposed Cplx-STGCN on this large-scale system, we studied the forecasting phenomenon for the Texas 2000-bus system. We evaluated the top four algorithms as presented in Fig. 4(a), namely Cplx-FNN, 1stGNN, GAN, and Cplx-STGCN, across different forecasting horizons represented by $H = 0, 1, 2, 3, 4, 5$. The outcomes, as depicted in Fig. 12, reveal that the Cplx-STGCN stands out as the most effective algorithm for voltage phasor forecasting in larger systems. Further illustration of the algorithm’s efficacy can be found in Fig. 13, which showcases two instances that highlight the tight alignment between our model’s predictions and the ground-truth voltage phasors for $H = 1$. While larger systems are often more sparse, the consistent smoothness and Cplx-STGCN, across different forecasting horizons represented by $H = 0, 1, 2, 3, 4, 5$. The outcomes, as depicted in Fig. 12, reveal that the Cplx-STGCN stands out as the most effective algorithm for voltage phasor forecasting in larger systems. Further illustration of the algorithm’s efficacy can be found in Fig. 13, which showcases two instances that highlight the tight alignment between our model’s predictions and the ground-truth voltage phasors for $H = 1$. While larger systems are often more sparse, the consistent smoothness
across both time and space ensures that the Cplx-STGCN performs commendably.

VII. CONCLUSION

In this paper, we have pioneered a complex-valued adaptation of the graph convolutional neural networks for AI in electric power systems, integrating components like complex-valued graph shift operators and graph signals. This not only equips the GCN to manage streaming data through the Cplx-STGCN architecture but also underscores its edge over real-valued GCNs. We further established theoretical bounds to guide the Cplx-STGCN’s transfer learning, obviating retraining needs. Tailored for power systems, our model chooses optimal inputs like voltage phasors and uses the Graph Fourier basis, ensuring its physics-aware design. Our results highlight a marked enhancement in FDI attack detection and PSSE/PSSF performance metrics, proving the efficacy of the Cplx-GCN structure within power system applications. The results of the experiments attest the potential of the nascent field of geometric deep learning in the complex domain, and can spur future research in Artificial Intelligence for energy system, wireless communication and biological networks whose signals are sparse in the Fourier domain.

APPENDIX A

PROOF OF THEOREM 1

Proof: We want to characterize how much a change in \( \hat{\mathbf{S}} \) changes the response of the feature extraction layer as

\[
\hat{\mathbf{H}}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}) = \sum_{k=0}^{K} (\hat{h}_k - h_k) \mathbf{S}^k = \sum_{k=0}^{K} (\hat{h}_k - h_k) (\mathbf{S} + \mathbf{E})^k
\]

(39)

We are interested in characterizing how different are the outputs of \( \hat{\mathbf{H}}(\mathbf{S}) \) and \( \mathbf{H}(\mathbf{S}) \). Let \( \sigma_{\max}(\mathbf{A}) \) be the largest singular value of matrix \( \mathbf{A} \); we know that \( \| \mathbf{Ax} \| \leq \sigma_{\max}(\mathbf{A}) \| \mathbf{x} \| \). Hence, for a given input, the norm of the difference of the output of the first layer is scaled at most by:

\[
\sigma_{\max}(\hat{\mathbf{H}}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S})) = \max_{i} [\sigma_i(\hat{\mathbf{H}}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}))] = \max_{i} \left| \sum_{k=0}^{K} (\hat{h}_k - h_k) \sigma_i^k (\mathbf{S} + \mathbf{E}) \right|.
\]

(40)

Furthermore, the following inequalities hold:

\[
\max_{\| \mathbf{E} \| \leq \epsilon} \sigma(\hat{\mathbf{H}}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S})) = \max_{z \in \Sigma(\mathbf{S})} \left| \sum_{k=0}^{K} (\hat{h}_k - h_k) z^k \right|
\]

\[
\leq \max_{z \in \Sigma(\mathbf{S})} \sum_{k=0}^{K} |\hat{h}_k - h_k| |z|^k = \max_{0 \leq s \leq \sigma_{\max}(\mathbf{S})} \sum_{k=0}^{K} |\hat{h}_k - h_k| s^k.
\]

(41)

where \( \Sigma(\hat{\mathbf{S}}) \) denotes the set of singular values of \( \hat{\mathbf{S}} \). Note that we can write \( \hat{\mathbf{S}} = \mathbf{S} + \mathbf{E} \), and then Let \( \nu_{\max} \) be the largest right singular vector of \( \hat{\mathbf{S}} \); we can write the inequalities:

\[
\sigma_{\max}(\hat{\mathbf{S}}) = \max_{\mathbf{x}} \frac{\| \hat{\mathbf{S}} \mathbf{x} \|}{\| \mathbf{x} \|} = \| (\mathbf{S} + \mathbf{E}) \nu_{\max} \| \\
\leq \| \mathbf{S} \nu_{\max} \| + \| \mathbf{E} \nu_{\max} \| \leq \sigma_{\max}(\mathbf{S}) + \epsilon,
\]

(42)

Therefore, (41) can be further bounded as:

\[
\max_{0 \leq s \leq \sigma_{\max}(\mathbf{S}) + \epsilon} \sum_{k=0}^{K} |\hat{h}_k - h_k| s^k \\
\leq \max_{0 \leq s \leq \sigma_{\max}(\mathbf{S}) + \epsilon} \sum_{k=0}^{K} |\hat{h}_k - h_k| s^k + \epsilon \max_{s \leq \sigma_{\max}(\mathbf{S})} \sum_{k=0}^{K} |\hat{h}_k - h_k| s^k.
\]

(43)

This inequality holds due to the fact that the feasible set is relaxed. Moreover, the polynomial \( \sum_{k=0}^{K} |\hat{h}_k - h_k| s^k \) has all positive coefficients, and thus is bounded by

\[
\max_{0 \leq s \leq \sigma_{\max}(\mathbf{S}) + \epsilon} \sum_{k=0}^{K} |\hat{h}_k - h_k| s^k \leq \max_{0 \leq s \leq \sigma_{\max}(\mathbf{S}) + \epsilon} \sum_{k=0}^{K} |\hat{h}_k - h_k| s^k + \epsilon \max_{s \leq \sigma_{\max}(\mathbf{S})} \sum_{k=0}^{K} |\hat{h}_k - h_k| s^k.
\]

(44)

Therefore, we can conclude that

\[
\sigma_{\max}(\hat{\mathbf{H}}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S})) \leq \gamma_1 \| \hat{\mathbf{h}} - \mathbf{h} \|_1
\]

(45)

This completes the proof. 



APPENDIX B

PROOF OF THEOREM 2

Proof: To bound of \( \rho(\hat{\mathbf{H}}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S})) \), let \( \mathbf{E} \triangleq \epsilon \bar{\mathbf{E}} \) and study the following expansion:

\[
\hat{\mathbf{H}}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}) = \sum_{k=0}^{K} h_k [\mathbf{S} + \mathbf{E}]^k - \mathbf{S}^k
\]

\[
= \sum_{k=1}^{K} h_k \sum_{\ell=1}^{k} \binom{k}{\ell} \mathbf{E}^\ell \mathbf{S}^{k-\ell}
\]

\[
= \sum_{k=1}^{K} h_k \sum_{\ell=1}^{k} \binom{k}{\ell} \mathbf{E}^\ell \mathbf{S}^{k-\ell},
\]

(46)

where it is easily to verify \( \| \bar{\mathbf{E}} \| \leq 1 \) due to \( \| \mathbf{E} \| \leq \epsilon \). We can rewrite (46) as the following matrix polynomial function:

\[
P(\epsilon) = A_1 \epsilon + \ldots + A_K \epsilon^K
\]

(47)

where the coefficients \( A_\ell, \forall \ell = 1, \ldots, K \) are expressed as follows:

\[
A_\ell = \sum_{k=\ell}^{K} h_k \binom{k}{\ell} \mathbf{E}^\ell \mathbf{S}^{k-\ell},
\]

(48)
and in particular, we have \( A_K = h_k \hat{E}^K \). Therefore, we have the bound for the norm of \( A_k \):

\[
\| A_k \|_2 \leq \sum_{k=1}^{K} |h_k| \left( \frac{1}{\ell} \right) \| S_k \|_2 \leq \sum_{k=1}^{K} |h_k| \left( \frac{1}{\ell} \right) \| S \|_2
\]

\[
= \sum_{k=1}^{K} |h_k| \left( \frac{1}{\ell} \right) \sigma_{\text{max}}(S) \leq \| A \|_1
\]

\[
\leq \max_{1 \leq k \leq K} \| h_k \| (1 + \sigma_{\text{max}}(S))^K
\]

(49)

Consider the definition (12) \( y_2 \leq \max_{1 \leq k \leq K} |h_k| (1 + \sigma_{\text{max}}(S))^K \). We have that:

\[
\|P(\epsilon)\|_2 \leq \|A_1\| + \cdots + \|A_K\| \leq \frac{\epsilon (1 - \epsilon^K)}{1 - \epsilon}
\]

(50)

This completes the proof.

Lemma 2: Considering the nonlinear activation function CReLU(\( \cdot \)), the distance between CReLU(\( \hat{H}(S)x \)) and CReLU(\( \hat{H}(S)x \)) is also bounded by:

\[
\| \text{CReLU} \left( \hat{H}(\hat{S})x \right) - \text{CReLU} \left( \hat{H}(\hat{S})x \right) \| \leq \left( \gamma_1 \| h - \hat{h} \|_1 + \gamma_2 \| x \| \right)
\]

(51)

Proof: Consider two complex numbers \( z_1 = x_1 + iy_1 \) and \( z_2 = x_2 + iy_2 \), the following relationship holds:

\[
|\text{CReLU}(z_1) - \text{CReLU}(z_2)| = |\text{CReLU}(x_1 + iy_1) - \text{CReLU}(x_2 + iy_2)|
\]

\[
= |\text{ReLU}(x_1 - x_2) + i \text{ReLU}(y_1 - y_2)|
\]

\[
= \sqrt{\text{ReLU}(x_1 - x_2)^2 + \text{ReLU}(y_1 - y_2)^2}
\]

\[
\leq \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = |z_1 - z_2|
\]

(52)

The inequality follows from \( |\text{ReLU}(x_1 - x_2)| \leq |x_1 - x_2| \). Then, we consider the distance between CReLU(\( \hat{H}(S)x \)) and CReLU(\( \hat{H}(S)x \)):

\[
\| \text{CReLU} \left( \hat{H}(\hat{S})x \right) - \text{CReLU} \left( \hat{H}(\hat{S})x \right) \| \leq \| \hat{H}(\hat{S})x \| \leq \| \hat{H}(\hat{S}) \| \| x \| \leq \| \hat{H}(\hat{S}) \| \| x \|
\]

\[
= \gamma_1 \| h - \hat{h} \|_1 + \gamma_2 \| x \| \left( \frac{1 - \epsilon^K}{1 - \epsilon} \right) \| x \|
\]

(53)

This completes the proof.

Likewise, we can easily verify that:

\[
|\text{tanh}(z_1) - \text{tanh}(z_2)| = |\text{tanh}(x_1 + iy_1) - \text{tanh}(x_2 + iy_2)|
\]

\[
= |\text{tanh}(x_1 - x_2) + iy \text{tanh}(y_1 - y_2)|
\]

\[
= \sqrt{(\text{tanh}(x_1 - x_2))^2 + (\text{tanh}(y_1 - y_2))^2}
\]

\[
\leq \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = |z_1 - z_2|
\]

(54)

The last equation holds due to the 1-Lipschitz property of \( \text{tanh}(\cdot) \).

**APPENDIX C**

**PROOF OF LEMMA 1**

**Proof:** We could express the multilayer neural networks, i.e., \( y = \Phi(x, S, \theta) \), as a function composition:

\[
g(x) = \hat{H}(S)x = \sum_{k=0}^{K} h_k S_k x, \quad x^1 = \text{CReLU}(g(x)),
\]

\[
f(x^1) = \Theta^{\text{cplx}} x^1, \quad y = \text{tanh}(f(x^1))
\]

(55)

Therefore, we could have the following inequality:

\[
\| y - \hat{y} \| = \| \text{tanh}(f(x^1)) - \text{tanh}(\hat{f}(x^1))\|
\]

\[
\leq \| \Theta^{\text{cplx}} x^1 - \Theta^{\text{cplx}} \hat{x}^1 \|
\]

\[
\leq \| \Theta^{\text{cplx}} x^1 - \Theta^{\text{cplx}} \hat{x}^1 + \Theta^{\text{cplx}} \hat{x}^1 - \hat{\Theta}^{\text{cplx}} \hat{x}^1 \|
\]

\[
\leq \| \Theta^{\text{cplx}} x^1 - \Theta^{\text{cplx}} \hat{x}^1 \| + \| \Theta^{\text{cplx}} \hat{x}^1 - \hat{\Theta}^{\text{cplx}} \hat{x}^1 \|
\]

For the first part, we have:

\[
\| \Theta^{\text{cplx}} x^1 - \Theta^{\text{cplx}} \hat{x}^1 \| \leq \| \Theta^{\text{cplx}} x^1 - \Theta^{\text{cplx}} \hat{x}^1 \| \| \Theta^{\text{cplx}} \hat{x}^1 \| \| x \|
\]

\[
\leq \delta \sum_{k=0}^{K} \| h_k S_k \| \leq \| \hat{h}(S + E)^k \| \| x \| \|
\]

(57)

The last inequality is due to Lemma 2. For the second part, we have:

\[
\| \Theta^{\text{cplx}} \hat{x}^1 - \Theta^{\text{cplx}} \hat{x}^1 \| \leq \| \Theta^{\text{cplx}} \hat{x}^1 - \Theta^{\text{cplx}} \hat{x}^1 \| \| \Theta^{\text{cplx}} \hat{x}^1 \| \| x \|
\]

\[
\leq \delta \sum_{k=0}^{K} \| h_k (S + E)^k \| \leq \| \hat{h}(S + E)^k \| \| x \|
\]

(58)

Similar to Theorem 1, we could bound \( \sum_{k=0}^{K} \| h_k (S + E)^k \| \) as follows:

\[
\| y - \hat{y} \| = \delta \gamma \| h \|_1
\]

(59)

By adding (56) and (59), we have the bound for \( \| y - \hat{y} \|:

\[
\| y - \hat{y} \| \leq \delta \gamma \| h \|_1 + \| x \| + \sigma_{\text{max}}(\Theta^{\text{cplx}})
\]

\[
\leq \gamma \| h \|_1 + \gamma_2 \| x \| \left( \frac{1 - \epsilon^K}{1 - \epsilon} \right) \| x \|
\]

(60)

This completes the proof.
APPENDIX D

STEALTH ATTACK

A stealth attack, as detailed in [5], involves a malefactor modifying current and voltage phasor measurements at certain buses, represented by $C$. The attacker introduces a perturbation given by:

$$\delta x_i^T = \left[ \delta x_C^T \ 0_{P+U}^T \right],$$

such that $Y_P C \delta x_C = 0, C \subset A$, where $P$ represents the set of honest nodes. Following this, the data received, $z_t$, with a FDI attack adheres to a specific structure:

$$z_t = H(x_t + \delta x_t) + \epsilon_t$$

To further elucidate:

$$z_a = z + a$$

A stealth attack, represented by $z_a$, has the identical residual as $z$, with the residual defined as $r = z - Hx = z_a - Hx_a$. When $H\delta x$ satisfies a particular condition, the residuals remain equivalent, ensuring the undetectability of the injected false data. The mathematical proof is shown as follows.

$$z_a = \left[ \begin{array}{c} z_C \\ z_P \end{array} \right] = z + \left[ \begin{array}{c} H_C \delta x_C \\ H_C \delta x_P + U \end{array} \right] x + \epsilon + \left[ \begin{array}{c} H_C \\ 0 \end{array} \right] (x + \delta x) + \epsilon = H(x + \delta x) + \epsilon$$

(61)

Consider a stealth attack represented by $z_a = z + a$. If this attack results in the same residual as $z$, with the residual defined as $r = z - Hx = z_a - Hx_a$, then it suggests a stealthy attack that remains undetected in the power system state estimation. Due to the fact that $H\delta x = H \left[ \begin{array}{c} \delta x_C \\ 0_{P+U} \end{array} \right] = \left[ \begin{array}{c} Y_P C \\ Y_{P+U} C \end{array} \right] \left[ \begin{array}{c} \delta x_C \\ \delta x_C \end{array} \right] = \left[ \begin{array}{c} 0 \\ a \end{array} \right]$, we have

$$x_a = (H^H H)^{-1} H^H z_a, x = (H^H H)^{-1} H^H z$$

Therefore, the residual is the same as:

$$\epsilon = z_a - Hx_a = \left( I - (H^H H)^{-1} H^H \right) z_a$$

$$\Rightarrow \left( I - (H^H H)^{-1} H^H \right) z = z - Hx = \epsilon$$

(63)

Essentially, injected false data is adeptly masked, making it elusive to traditional bad data processing methods that rely on residuals. This inspired us to devise the Cplx-STGCN for detecting and pinpointing false data injection, as detailed in Section V-B.

APPENDIX E

ILLUSTRATION OF CPLX-GCN

Multiple real-valued GCNs cannot replicate the structure of a complex-valued GCN due to the inherent correlations between real and imaginary components. To illustrate this, consider the case where $k = 1$ and $h_1 = a + jb$:

$$(a + jb)(\Re(S) + j\Im(S))(\Re(x) + j\Im(x))$$

$$= (a + jb)(\Re(S)\Re(x) - \Im(S)\Im(x))$$

$$+ j(a\Re(S)\Im(x) + b\Im(S)\Re(x))$$

(64)

Given the preceding formulation, we can elegantly express the result as a two-dimensional vector:

$$\left[ a(\Re(S)\Re(x) - \Im(S)\Im(x)) \right]$$

$$\left[ b(\Re(S)\Im(x) + a\Im(S)\Re(x)) \right]$$

(65)

From the above calculations, it becomes evident that two real-GCNs cannot be equivalent to a complex-valued GCN in all situations due to the couplings between the real and imaginary parts of weights, GSO and GS.

ACKNOWLEDGMENT

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.

REFERENCES

[1] H. Leung and S. Haykin, “The complex backpropagation algorithm,” IEEE Trans. Signal Process., vol. 39, no. 9, pp. 2101–2104, Sep. 1991.
[2] C. Trabelsi et al., “Deep complex networks,” in Proc. ICRL, 2018, pp. 1–19.
[3] A. Hirose and S. Yoshida, “Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence,” IEEE Trans. Neural Netw. Learn. Syst., vol. 23, no. 4, pp. 541–551, Apr. 2012.
[4] J. Bassey, L. Qian, and X. Li, “A survey of complex-valued neural networks,” 2021, arXiv:2101.12249.
[5] R. Ramakrishna and A. Scaglione, “Grid-graph signal processing (Grid-GSP): A graph signal processing framework for the power grid,” IEEE Trans. Signal Process., vol. 69, pp. 2725–2739, Apr. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9415125
[6] A. Sandryhaila and J. M. Moura, “Discrete signal processing on graphs,” IEEE Trans. Signal Process., vol. 61, no. 7, pp. 1644–1656, Apr. 2013.
[7] A. Ortega, F. Frossard, J. Kovačević, J. M. Moura, and P. Vanderghynst, “Graph signal processing: Overview, challenges, and applications,” Proc. IEEE, vol. 106, no. 5, pp. 808–828, May 2018.
[8] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, “Graph signal processing for machine learning: A review and new perspectives,” IEEE Signal Process. Mag., vol. 37, no. 6, pp. 117–127, Nov. 2020.
[9] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs,” in Proc. NeurIPS, vol. 29, 2016, pp. 3837–3845.
[10] T. Adali, P. J. Scheuer, and L. L. Scharf, “Complex-valued signal processing: The proper way to deal with impropriety,” IEEE Trans. Signal Process., vol. 59, no. 11, pp. 5101–5125, Nov. 2011.
[11] I. Jabloński, “Graph signal processing in applications to sensor networks, smart grids, and smart cities,” IEEE Sensors J., vol. 17, no. 23, pp. 7659–7666, Dec. 2017.
[12] S. S. Saha, A. Scaglione, R. Ramakrishna, and N. G. Johnson, “Distribution systems AC state estimation via sparse AMI data using graph signal processing,” IEEE Trans. Smart Grid, vol. 13, no. 5, pp. 3636–3649, Sep. 2022.
[13] A. Sperduti and A. Starita, “Supervised neural networks for the classification of structures,” IEEE Trans. Neural Netw., vol. 8, no. 3, pp. 714–735, May 1997.
[14] B. Brun, W. Zarabina, A. Szlam, and Y. LeCun, “Spectral networks and deep locally connected networks on graphs,” in Proc. Int. Conf. Learn. Represent., 2014, pp. 1–14.
[15] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” in Proc. 32nd AAAI, 2018, pp. 7444–7452.

[16] T. Wu, I. L. Carreño, A. Scaglione, and D. Arnold, “Spatio-temporal graph convolutional neural networks for physics-aware grid learning algorithms,” IEEE Trans. Smart Grid, vol. 14, no. 5, pp. 4086–4099, Sep. 2023.

[17] Y. Seo, M. Defferrard, P. Vanderheynst, and X. Bresson, “Structured sequence modeling with graph convolutional recurrent networks,” in Proc. ICONIP, 2018, pp. 362–373.

[18] F. Gama, J. Bruna, and A. Ribeiro, “Stability properties of graph neural networks,” IEEE Trans. Signal Process., vol. 68, pp. 5680–5695, Sep. 2020, doi: 10.1109/TSP.2020.3026080.

[19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, “A comprehensive survey on graph neural networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 1, pp. 4–24, Jan. 2021.

[20] W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang, and Y. Wang, “A review of graph neural networks and their applications in power systems,” J. Mod. Power Syst. Clean Energy, vol. 10, no. 2, pp. 345–360, Mar. 2022.

[21] T. Wu, Y.-J. A. Zhang, Y. Liu, W. C. Lau, and H. Xu, “Missing data recovery in large power systems using network embedding,” IEEE Trans. Smart Grid, vol. 12, no. 1, pp. 680–691, Jan. 2021.

[22] M. Gao, J. Yu, Z. Yang, and J. Zhao, “A physics-guided graph convolution neural network for optimal power flow,” IEEE Trans. Power Syst., early access, Jan. 20, 2023, doi: 10.1109/TPWRS.2023.3238377.

[23] O. Boyaci, M. R. Narimani, K. R. Davis, M. Ismail, T. J. Overbye, and E. Serpedin, “Joint detection and localization of stealth false data injection attacks in smart grids using graph neural networks,” IEEE Trans. Smart Grid, vol. 13, no. 1, pp. 807–819, Jan. 2022.

[24] G. Zhang, J. Li, O. Bamisile, D. Cai, W. Hu, and Q. Huang, “Spatio-temporal correlation-based false data injection attack detection using deep convolutional neural network,” IEEE Trans. Smart Grid, vol. 13, no. 1, pp. 750–761, Jan. 2022.

[25] M. Liang, Y. Meng, J. Wang, D. L. Labkeman, and N. Lu, “FeederGAN: Synthetic feeder generation via deep graph adversarial nets,” IEEE Trans. Smart Grid, vol. 12, no. 2, pp. 1163–1173, Mar. 2021.

[26] K. Chen, J. Hu, Y. Zhang, Z. Yu, and J. He, “Fault location in power distribution systems via deep graph convolutional networks,” IEEE J. Sel. Areas Commun., vol. 38, no. 1, pp. 119–131, Jan. 2020.

[27] B. L. Nguyen, T. V. Vu, T.-T. Nguyen, M. Panwar, and R. Hovsapian, “Spatio-temporal recurrent graph neural networks for fault diagnostics in power distribution systems,” IEEE Access, vol. 11, pp. 46039–46050, 2023.

[28] M. B. Do Coutto Filho and J. C. S. de Souza, “Forecasting-aided state estimation—Part I: Panorama,” IEEE Trans. Power Syst., vol. 24, no. 4, pp. 1667–1677, Nov. 2009.

[29] M. B. Do Coutto Filho, J. C. S. de Souza, and R. S. Freund, “Forecasting-aided state estimation—Part II: Implementation,” IEEE Trans. Power Syst., vol. 24, no. 4, pp. 1678–1685, Nov. 2009.

[30] L. Zhang, G. Wang, and G. B. Giannakis, “Power system state forecasting via deep recurrent neural networks,” in Proc. IEEE ICASSP, 2019, pp. 8092–8096.

[31] M. J. Hossain and M. Rahnamay-Naeini, “State estimation in smart grids using temporal graph convolution networks,” in Proc. NAPS, 2021, pp. 1–5.

[32] 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 Things J., vol. 7, no. 9, pp. 8218–8227, Sep. 2020.

[33] Y. Wang, Z. Zhang, J. Ma, and Q. Jin, “KFRNN: An effective false data injection attack detection in smart grid based on Kalman filter and recurrent neural network,” IEEE Internet Things J., vol. 9, no. 9, pp. 6893–6904, May 2022.

[34] A. S. Musleh, G. Chen, and Z. Y. Dong, “A survey on the detection algorithms for false data injection attacks in smart grids,” IEEE Trans. Smart Grid, vol. 11, no. 3, pp. 2218–2234, May 2020.

[35] S. Ding, F. Feng, X. He, Y. Liao, J. Shi, and Y. Zhang, “Causal incremental graph convolution for recommender system retraining,” IEEE Trans. Neural Netw. Learn. Syst., early access, Mar. 16, 2022, doi: 10.1109/TNNLS.2022.3156066.

[36] X. Chen, J. Wang, and K. Xie, “Trafficstream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning,” 2021. arXiv:2106.06273.

[37] J. Wang, G. Song, Y. Wu, and L. Wang, “Streaming graph neural networks via continual learning,” in Proc. 29th ACM Int. Conf. Inf. Knowl. Manag., 2020, pp. 1515–1524.

[38] T. Wu, Y.-J. A. Zhang, and H. Wen, “Voltage stability monitoring based on disagreement-based deep learning in a time-varying environment,” IEEE Trans. Power Syst., vol. 36, no. 1, pp. 28–38, Jan. 2021.

[39] R. Ramakrishna and A. Scaglione, “On modeling voltage phasor measurements as graph signals,” in Proc. IEEE Data Sci. Workshop (DSW), 2019, pp. 275–279.

[40] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” in Proc. 27th IJCAI, 2018, pp. 3634–3640.

[41] A. Anis, A. Gadde, and A. Ortega, “Efficient sampling set selection for bandlimited graph signals using graph spectral proxies,” IEEE Trans. Signal Process., vol. 64, no. 14, pp. 3775–3789, Jul. 2016.

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

[43] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. ICLR, 2017, pp. 1–14.

[44] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 5998–6008.

[45] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” Stat, vol. 10, no. 1, 2023, Art. no. 10–48550.

[46] L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini, “Recipe for a general, powerful, scalable graph transformer,” in Proc. Adv. Neural Inf. Process. Syst., vol. 35, 2022, pp. 1–15.