Graph Convolutional Neural Networks for Power Line Outage Identification

Jia He
Illinois Institute of Technology
Chicago, USA
jhe58@hawk.iit.edu

Maggie Cheng
Illinois Institute of Technology
Chicago, USA
maggie.cheng@iit.edu

***Abstract*—In this paper, we consider the power line outage identification problem as a graph signal classification problem, where the signal at each vertex is given as a time series. We propose graph convolutional networks (GCNs) for the task of classifying signals supported on graphs. An important element of the GCN design is filter design. We consider filtering signals in either the vertex (spatial) domain, or the frequency (spectral) domain. Two basic architectures are proposed. In the spatial GCN architecture, the GCN uses a graph shift operator as the basic building block to incorporate the underlying graph structure into the convolution layer. The spatial filter directly utilizes the graph connectivity information. It defines the filter to be a polynomial in the graph shift operator to obtain the convolved features that aggregate neighborhood information of each node. In the spectral GCN architecture, a frequency filter is used instead. A graph Fourier transform operator first transforms the raw graph signal from the vertex domain to the frequency domain, and then a filter is defined using the graph's spectral parameters. The spectral GCN then uses the output from the graph Fourier transform to compute the convolved features. There are additional challenges to classify the time-evolving graph signal as the signal value at each vertex changes over time. The GCNs are designed to recognize different spatiotemporal patterns from high-dimensional data defined on a graph. The application of the proposed methods to power line outage identification shows that these GCN architectures can successfully classify abnormal signal patterns and identify the outage location.**

***Index Terms*—Graph Convolutional Neural Network, Spatial Domain, Spectral Domain, Graph Fourier Transform**

I. INTRODUCTION

In this paper we provide a data-driven approach to power line outage identification from measurement data. Measurement data are collected from buses in the power system and presented as times series. To discover the unique spatiotemporal pattern that characterizes a power line outage, we need a new technique for mining high-dimensional time series data defined on a graph.

Although convolutional neural networks (CNNs) are widely used in image processing, the convolution operation that is designed for image data cannot be readily applied to a graph structure. This is because in image processing, data are laid on a regular structure such as a two-dimensional array or a matrix, and a convolution operation is an element-wise matrix multiplication operation between the matrix representing the input image and a convolutional matrix, also called a kernel or a filter. In a typical convolution layer, a filter, typical of much smaller size than the input, is used to slide over the grid to learn new features. The convolution product is expected to extract the spatial feature hidden in the image. Such a convolution operation is based on the fact that adjacent signal supports are also adjacent in the matrix. However this foundation is shaken when we consider a graph signal. A graph signal is a discrete time signal defined on a graph, where each component is assigned to a vertex of the graph. Although we can represent it as an array, adjacent vertices on a graph may not be neighbors to each other in the array, and vice versa. This motivates the design of graph convolutional network models that are particularly designed for learning on graphs. It is a generalization of convolutional neural networks from regular grid data to irregular graph data.

The major building block of a GCN is the convolution layer. Graph convolution is realized through a graph filter, the counterpart of the filter used in CNN. The graph filter incorporates the graph structure by using a shift operator to effectively extracts high level features from signals defined on a graph. A shift operator for graph signals is a matrix associated with the graph. Common choices for the matrix include adjacency matrix, Laplacian matrix, or normalized versions of these matrices [1]. Different choices represent different trade-offs [2].

In designing the GCN, we use the convolution layer to roll out the signal on a graph structure. A given graph signal is filtered using one of the two methods: filtering in a graph's spatial domain or spectral domain. The construction of the convolution layer from spatial domain focuses on the graph connectivity information. The spatial filter is a polynomial of the shift operator. The degree H of the polynomial indicates that signal values from neighbors that are within H hops of a node are collected to contribute to the new feature at this node. On the other hand, convolution from the spectral domain focuses on the graph's frequency property. We first convert a graph signal into the frequency domain through graph Fourier transform to obtain its frequency representation. The spectral coefficients are then used as input for the convolution layer. The filters are multipliers of the graph frequencies, and the convolution layer is learned in the graph's Fourier space.

The proposed GCN models are applied to the spatiotemporal data from the power grid to identify line outages. Previous works for line outage identification mostly rely on load-flow analysis or state estimation ([3] , [4] , [5]). The proposed methods are data-driven and do not require knowledge of the models and parameters of the power system.

The rest of the paper is organized as follows. In sections II and III, we illustrate the construction of convolution layers from the spatial domain and the spectral domain, respectively, in particular, section III-A introduces graph Fourier transform (GFT) and related concepts such as interpretation of graph frequencies. Section IV illustrates the connection between spectral GCN and spatial GCN. In section V we show the application of the proposed GCN architectures to power line outage identification. Section VI summarizes the most related work, and section VII concludes the paper.

II. Graph Convolutional Network Using Spatial Domain Convolution

A. Spatial Convolution

Consider a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with N nodes. Let $\boldsymbol{x} \in \mathcal{R}^N$ be a snapshot of the graph signal, where each element of $\boldsymbol{x}$ corresponds to a nodal feature.

We consider to use $\tilde{A}$, the normalized adjacency matrix, as the shift operator to get convolution outputs.

$$\tilde{A} = D^{-1/2} A D^{-1/2},$$

where A is the adjacency matrix of the graph, $D = \mathrm{diag}(d)$ is an $N \times N$ diagonal matrix, and $d(i)$ is the degree of node i, i.e., $d(i) = \sum_{j=1}^{N} A_{ij}, i = 1, \cdots, N$.

If we only want the information from a node itself and its one-hop neighbors, we have

$$\boldsymbol{y} = w_0 \boldsymbol{x} + (w_1 \tilde{A}) \boldsymbol{x} \tag{1}$$

w_0 and w_1 are two scalars denoting the filter parameters. The output $\boldsymbol{y}$ is also a signal defined on the same graph. We write the convolution layer as a linear combination of product of $\tilde{A}^i$ and $\boldsymbol{x}$ to emphasize how the shift operator is used to gather information propagated from one-hop neighbors.

We now generalize this idea to H-hops, and understand the filter as a polynomial in $\tilde{A}$ by rewriting equation (1) as

$$\boldsymbol{y} = \sum_{i=0}^{H} (w_i \tilde{A}^i) \boldsymbol{x} \tag{2}$$

and define $w(\tilde{A}) = \sum_{i=0}^{H} w_i \tilde{A}^i$ as the vertex domain filter.

Equation (2) shows filtering in the vertex domain for a snapshot of a graph signal, $\boldsymbol{x} \in \mathcal{R}^N$. We can generalize this to cases where we have K input features, i.e., $X = [\boldsymbol{x_1}, \cdots, \boldsymbol{x_K}] \in \mathcal{R}^{N \times K}$. Now instead of having $H+1$ scaler filter parameters, we have $H+1$ column vectors $\boldsymbol{w_i} = [w_i^1, \cdots, w_i^K]^T \in \mathcal{R}^K, i = 0, \ldots, H$, and the output feature can be obtained as

$$\boldsymbol{y} = X\boldsymbol{w_0} + (\tilde{A}X)\boldsymbol{w_1} + \ldots + (\tilde{A}^H X)\boldsymbol{w_H}$$

We can interpret each element of $\boldsymbol{y}$ as a linear combination of the input features at that node and its neighbors within H-hops.

If we want to extract more than one output features from the input $X \in \mathcal{R}^{N \times K}$, we define matrices of filter parameters $W_i = [\boldsymbol{w_i^1}, \ldots, \boldsymbol{w_i^G}] \in \mathcal{R}^{K \times G}, i = 0, \ldots, H$. Here G is the number of output features, and $\boldsymbol{w_i^g} \in \mathcal{R}^K$ corresponds to the filter parameters to produce the gth output feature. The output Y is now a $N \times G$ matrix with

$$Y = [\boldsymbol{y^1}, \cdots, \boldsymbol{y^G}] = XW_0 + (\tilde{A}X)W_1 + \ldots + (\tilde{A}^H X)W_H \tag{3}$$

$\boldsymbol{y^g}$ is the gth convolved feature. The number of learnable parameters for the convolution layer is $\mathcal{O}(KG)$ for K input features and G output features. To complete the convolution layer, a point-wise nonlinear activation function $\sigma(\cdot)$ is applied to Y.

B. Spatial Convolution with Frequency Features

In Section II-A, we consider the K features at each node as a discrete time series of length K, and we have N such time series from N nodes. Such datasets are used in many applications. However in some applications, the magnitudes of the signal values reveal less about the status of the system than the changes of the values. Directly using the measurement can lead to large false positive and false negative rates. For the application we will discuss in this paper, i.e, power line outage identification, it is the change of phasor angles rather than the value itself that indicates the status of the power line.

One way to deal with the problem is to consider a discrete Fourier transform of the signal at each node. We preprocess X by first taking DFT of N time series sequence. Recall that for a discrete time signal $\boldsymbol{s} = [s_0 s_1 \cdots s_{N-1}]^\top \in \mathbb{C}^N$, where each element denotes a sample at time n, the DFT output $\widehat{\boldsymbol{s}} = [\widehat{s}_0 \widehat{s}_1 \cdots \widehat{s}_{N-1}]^\top \in \mathbb{C}^N$ is computed as

$$\widehat{s}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} s_n e^{-i\frac{2\pi}{N}kn}, k = 0, \cdots, N-1 \tag{4}$$

In (5), $\widehat{s}_k$ are Fourier coefficients correspond to frequency $f_k = \frac{2\pi k}{N}$, $\boldsymbol{x_k} = \frac{1}{\sqrt{N}}[1, e^{-i\frac{2\pi}{N}k}, \cdots, e^{-i\frac{2\pi}{N}k(N-1)}]$ are the spectral components corresponds to each k. The magnitude of Fourier coefficient, $|\widehat{s}_k|$ qualifies the energy distribution of the input signal at different frequencies.

Applying DFT to each row of X returns its frequency domain representation $\widehat{X}$, which is then used as input for the convolution layer (see Fig. 1). We denote this architecture as *Spatial+DFT*, and denote the previous one (in Section II-A) without DFT preprocessing simply as *Spatial*.

III. Graph Convolutional Network Using Spectral Domain Convolution

The convolution layer of the spatial GCNs in Section II operates on the vertex domain of the graph. A different concept is to operate on the spectral domain of the graph. We will study the eigenvalues and eigenvectors of the matrices associated with the graph. Each graph topology corresponds to a set of frequency components given by the eigenvectors of the matrix. These frequency components are used to convert the graph signal into its frequency domain. This can be done by

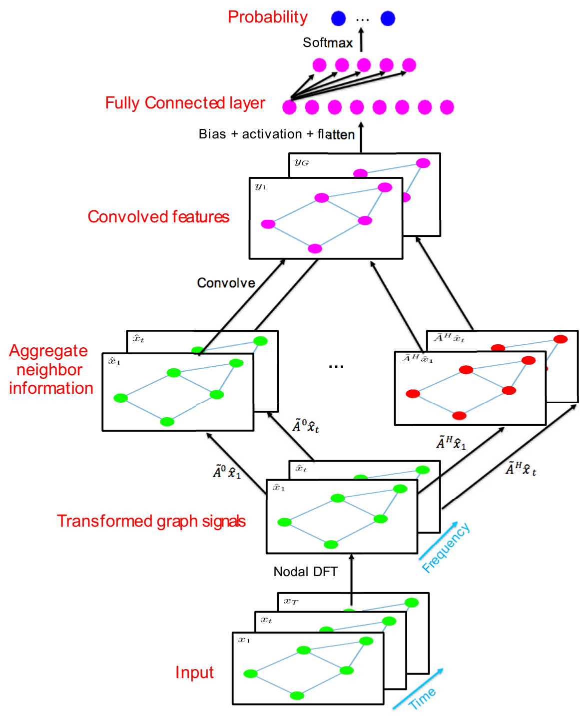


Fig. 1. The architecture for Spatial GCN with DFT preprocessing. Temporal DFT is first applied to the time series at each node. The Fourier coefficients at all nodes corresponding to the same frequencies define a new graph signal $\widehat{\boldsymbol{x}}$. Next, $\widehat{\boldsymbol{x}}$ is convolved by using the spatial filter defined as a linear combination of the shift operator. The convolved features $\boldsymbol{y}^g, g = 1, \cdots, G$, are then flattened to build a fully-connected layer. The last layer uses softmax for classification.

a Fourier transform on the graph signal. Similar to discrete Fourier transform of signals defined on a linear structure, such as a time series, we can analogously define Fourier transform of signals defined on a non-linear graph structure.

A. Graph Fourier Transform

Graph Fourier transform extends the idea of discrete Fourier transform to vectors defined on a graph. We first provide a graph interpretation of DFT and show how it can be extended to GFT.

The discrete Fourier transform of a vector $\boldsymbol{s}$ can be written as a matrix-vector multiplication:

$$\widehat{\boldsymbol{s}} = \mathbf{DFT}\boldsymbol{s} \tag{5}$$

where the element of $\widehat{\boldsymbol{s}}$ is obtained by:

$$\widehat{s}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} s_n e^{-i\frac{2\pi}{N}kn}, \; k = 0, \cdots, N-1$$

Let $\omega = e^{-i2\pi/N}$. The $\mathbf{DFT}$ matrix can then be defined as follows:

$$\mathbf{DFT} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^k & \cdots & \omega^{k(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

The row vectors of $\mathbf{DFT}$ are $\frac{1}{\sqrt{N}}[1, \omega^k, \cdots, \omega^{k(N-1)}]$, $k = 0, \cdots, N-1$, which are the spectral components.

To give DFT operation a graph interpretation, we can consider vector $\boldsymbol{s}$ as a signal defined on a directed cycle graph $0 \to 1 \to \ldots n-1 \to 0$ ([2], [6]). The adjacency matrix of the cycle graph, in which element $A_{i,j}$ represents the edge from j to i, is the following matrix,

$$\mathbf{A}_c = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

The eigen-decomposition of $\mathbf{A}_c$ gives

$$\begin{aligned} \mathbf{A}_c &= \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1} \\ &= \mathbf{DFT}^{-1} \begin{pmatrix} e^{-i\frac{2\pi}{N}\cdot 0} & & \\ & \ddots & \\ & & e^{-i\frac{2\pi\cdot(N-1)}{N}} \end{pmatrix} \mathbf{DFT} \end{aligned} \tag{6}$$

Columns of $\mathbf{V}$ are eigenvectors of $\mathbf{A}_c$, and $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues of $\mathbf{A}_c$. Since the frequencies $\frac{2\pi k}{N}$ of a DFT operation correspond to the eigenvalues of $\mathbf{A}_c$, we can view the eigenvalues as a representation of frequencies. Analogously, $\mathbf{V}^{-1}$ is considered playing the role of the $\mathbf{DFT}$ matrix in the context of Fourier transform of graph signals.

The connection between DFT and circulant adjacency matrix of time signal leads to the natural definition of Fourier transform operated on graph signals. We now extend from the directed cycle graph to general graphs, which only requires to replace $\mathbf{A}_c$ with the normalized adjacency matrix $\tilde{A}$:

$$\tilde{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1} \tag{7}$$

Here $\mathbf{\Lambda} = \mathrm{diag}\,[\lambda_0 \cdots \lambda_{N-1}]$ is a complete set of eigenvalues. $\mathbf{V} = [\boldsymbol{v}_0, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_{N-1}]$ is an $N \times N$ matrix with $\boldsymbol{v}_i$ denoting the eigenvector corresponding to λ_i. The eigenvectors are the graph spectral components.

We define matrix $\mathbf{GFT} = \mathbf{V}^{-1}$. Analogous to DFT in (5), the graph Fourier transformation of a vector $\boldsymbol{s}$ is also given as a matrix-vector multiplication:

$$\widehat{\boldsymbol{s}} = \mathbf{GFT}\boldsymbol{s} = \mathbf{V}^{-1}\boldsymbol{s} = [\widehat{s}_0, \widehat{s}_1, \cdots, \widehat{s}_{N-1}]^\top \tag{8}$$

where $\widehat{s}_k$ is the Fourier coefficient corresponds to λ_k.

The remaining question is how to understand the notion of frequency on graphs. In DFT this is straightforward, as frequency $f_k = \frac{2\pi k}{N}$, which is interpreted as k cycles per N samples. Large frequency corresponds to highly variated spectral components. Similarly, we can define high graph frequencies as those whose spectral components have a high variation, i.e., magnitudes of neighboring nodes are highly variated. It is proved in [7] that for adjacency matrix, smaller eigenvalues corresponds to higher variated eigenvectors, thus represents higher frequencies.

B. Spectral Convolution in GCN

For our application, the graph under consideration is undirected. Since the normalized adjacency matrix $\tilde{A}$ is symmetric, we have $\mathbf{GFT} = \mathbf{V}^{-1} = \mathbf{V}^T \in \mathcal{R}^{N\times N}$.

To derive output features from the convolution layer, we first define the frequency filter as a degree H polynomial in matrix $\mathbf{\Lambda}$ to consider information from nodes within H hops.

$$w(\mathbf{\Lambda}) = \sum_{h=0}^{H} w_h \mathbf{\Lambda}^h \tag{9}$$

The convolutional output is then given by

$$\widehat{\boldsymbol{y}} = w(\mathbf{\Lambda})\widehat{\boldsymbol{x}} = w_0\widehat{\boldsymbol{x}} + w_1\mathbf{\Lambda}\widehat{\boldsymbol{x}} + \ldots + w_H\mathbf{\Lambda}^H\widehat{\boldsymbol{x}}, \tag{10}$$

where $w_i, i = 0, 1, \ldots, H$, are trainable weights, and $\widehat{\boldsymbol{x}} = \mathbf{V}^T\boldsymbol{x}$. Elements of $\widehat{\boldsymbol{x}}$ are the spectral coefficients of $\boldsymbol{x}$. In spectral convolution, we take $\widehat{\boldsymbol{x}}$ as input features for the convolution layer.

The generalization of the above operation to a middle layer in the GCN, where there are G output features and K input features, is achieved by using a simple summation,

$$\widehat{\boldsymbol{y}}^g = \sum_{k=1}^{K} w^{k,g}(\mathbf{\Lambda})\widehat{\boldsymbol{x}}^k \tag{11}$$

where each parameter $w^{k,g}(\mathbf{\Lambda})$ is used to connect the kth input feature $\widehat{\boldsymbol{x}}^k$ to the gth output feature $\widehat{\boldsymbol{y}}^g$. Same as convolution in spatial domain, the number of learnable weights is $\mathcal{O}(KG)$.

The input to the GCN is a graph signal $\{\boldsymbol{x_t}, t = 1, \ldots, T\}$, with each $\boldsymbol{x_t} \in \mathcal{R}^N$. The GFT operation is first applied to each $\boldsymbol{x_t}$ to get $\widehat{\boldsymbol{x}}_t$, and then the convolved features are obtained using (11). To complete the convolution layer, a point-wise nonlinear activation function is applied to each component of $\widehat{\boldsymbol{y}}$. The corresponding architecture is shown in Figure 2. We denote this architecture as *Spectral*.

IV. Discussion on Convolution in Spatial and Spectral Domain

In this part, we exploit the relation between the spectral representation and the spatial representation of the signal and the implication of it to the GCN design. It is shown in [1] that the convolution in the spatial domain is equivalent to multiplication in spectral domain. We show that this equivalence relation can be achieved in the GCN design.

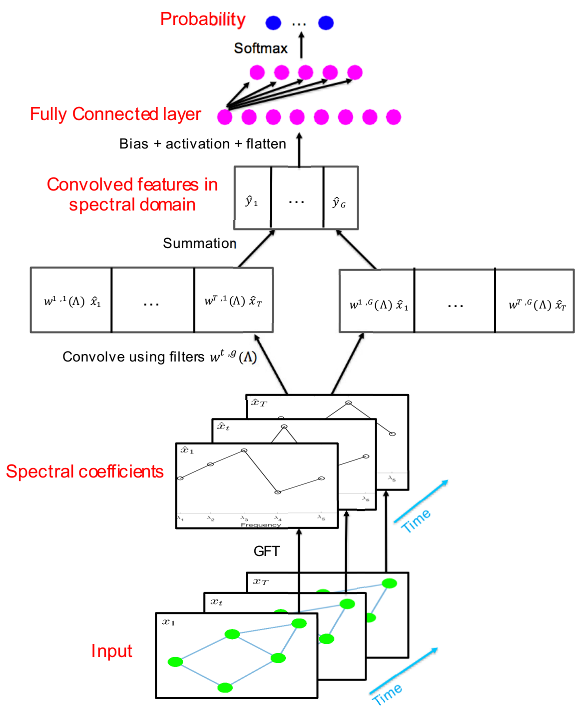


Fig. 2. Spectral GCN model architecture. GFT is first applied to $\boldsymbol{x_t}$ to obtain graph frequency representation $\widehat{\boldsymbol{x}}_t$ of the signals. Then convolution in frequency domain is carried out by defining filters as a linear combination of graph frequencies matrix $\mathbf{\Lambda}$. The operation after convolution layer is the same as in the GCN from spatial domain.

We first look at a GCN design with some modification from the proposed spectral method in Section III-B. We denote this new architecture as *Spectral-B*.

Spectral-B is similar to Fig. 2 in the sense that the convolution layer is developed in the graph frequency domain, however, different from Fig. 2, the operation in equation (10) is further followed by an inverse Fourier transform to convert the output back into the graph spatial domain. For that purpose, we first introduce the inverse graph Fourier transform, or inverse GFT, which is defined as follows

$$\mathbf{GFT}^{-1} = \mathbf{V}$$

Then, we can define the convolution layer as

$$\boldsymbol{y} = \mathbf{GFT}^{-1}\big(w(\mathbf{\Lambda})\widehat{\boldsymbol{x}}\big) = \mathbf{V}\big(w(\mathbf{\Lambda})\widehat{\boldsymbol{x}}\big) \tag{12}$$

It is worth noticing the following relationship of the outputs between equations (10) and (12)

$$\mathbf{V}^T\boldsymbol{y} = w(\mathbf{\Lambda})\widehat{\boldsymbol{x}} = \widehat{\boldsymbol{y}} \tag{13}$$

That is to say, after filtering the signal in the frequency domain, instead of using the frequency information of the filtered signal as in equation (10), we use its spatial information for model training. The operation concerning multiple input and output features is carried out similarly to equation (11), i.e., the gth convolved feature is

$$\boldsymbol{y}^g = \mathbf{V}\Big(\sum_{k=1}^{K} w^{k,g}(\mathbf{\Lambda})\widehat{\boldsymbol{x}}^k\Big) \tag{14}$$

followed by a nonlinear activation $\sigma(\boldsymbol{y}^g)$.

Recall that in GCN *Spatial* (Section II-A), the convolution is given by $\boldsymbol{y} = w(\tilde{A})\boldsymbol{x}$. This can be successively carried out as:

$$\begin{aligned}\boldsymbol{y} &= w(\tilde{A})\boldsymbol{x} \\ &= w(\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T)\boldsymbol{x} \\ &= \mathbf{V}w(\boldsymbol{\Lambda})\mathbf{V}^T\boldsymbol{x} \\ &= \mathbf{V}\Big(w(\boldsymbol{\Lambda})\widehat{\boldsymbol{x}}\Big)\end{aligned} \tag{15}$$

This is exactly the convolution done in GCN *Spectral-B*, as in equation (12). Furthermore, it provides another interpretation of filtering a graph signal in spatial domain. The convolution process is decomposed by first taking the GFT of $\boldsymbol{x}$ to get $\mathbf{V}^T\boldsymbol{x}$, then filtering the signal in the frequency domain via multiplying the spectrum of graph signal by frequency filter $w(\boldsymbol{\Lambda})$. The last step is to perform inverse GFT to transform the filtered results back to spatial domain. Experiment results in section V further verified the performance of the two neural nets have the same discriminant power.

V. Experiment

A. Power Line Outage Identification Problem

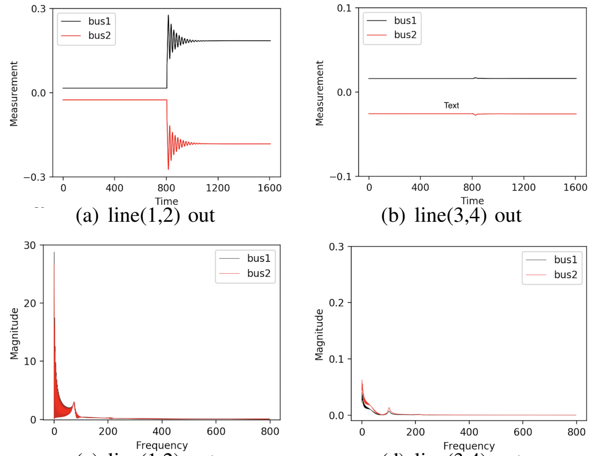
We study the potential of GCN models on power line outage identification. Given a set of spatial-temporal structured data collected from a power system, we build a model to identify the power lines where outages have occurred. It is possible that multiple outages occur simultaneously. Our GCN models are built to predict the probability of outage for each power line, therefore multiple line outages do not incur additional computational complexity or cost.

We test the GCN models on IEEE 39-bus and 118-bus systems. First we simulate line outage cases through Power System Analysis Toolbox (PSAT) [8] and collect phasor angle θ at each bus node provided by Phasor Measurement Units (PMUs), a device that provides real time power system measurement data. Simulations last for a total of 200s, and the line outage is fixed at 100s. We collect the data at the interval of 1/8 second (i.e., 8 data points per second), and obtain m-dimensional spatial-temporal data $\{\boldsymbol{\theta}_t\}$, where $\boldsymbol{\theta}_t \in \mathcal{R}^m$, is a snapshot of the graph signal, m is the number of nodes in a power system and $t = 1 \ldots n$ is the ticks in time series. For each system, we simulate 20 outage cases, with each having a different line outage location randomly chosen from all power lines. The dataset for the 39-bus system has 460 inputs, among which 180 inputs are labeled *1* denoting a line outage case; others are labeled *0* denoting a non-outage case. For the 118-bus system, there are 1120 inputs with 100 line outage cases.

The power line outage problem is formulated as a binary classification problem. A GCN model has multiple output nodes with each answering whether outage has occurred on a particular power line.

B. Feature Extraction

To extract features that are highly correlated with power line topology from the collected measurements, we consider the power transfer between two nodes. Let P_{ij} be the real power transfer from node i to node j, we have



(a) line(1,2) out (b) line(3,4) out (c) line(1,2) out (d) line(3,4) out

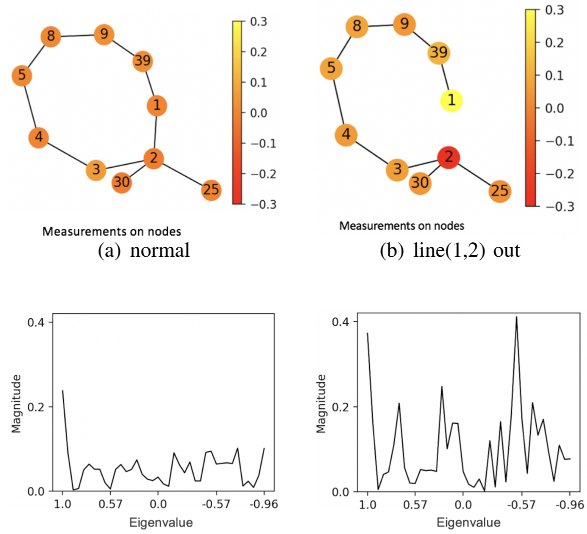
Fig. 3. Upper row: θ_1' and θ_2' against time regarding different line outages. Lower row: the magnitudes of DFT spectral coefficients corresponding to angles in the upper row. The left column is when line(1,2) is out, the right column is when line(3,4) is out. The DFT magnitude of the signal is symmetric, so only the first half is shown.

$$P_{ij} = |V_i|\,|V_j|\,(G_{ij}\cos(\theta_i - \theta_j) + B_{ij}\sin(\theta_i - \theta_j))$$

where $|V_i|$ is the voltage magnitude at bus i. G_{ij} and B_{ij} are line parameters between bus i and bus j. It has been shown in [9] that the the real part of admittance matrix $G_{ij} \to 0$, $|V_i|, i \in \mathcal{N}$ is close to 1 in per unit system, and we can use $(\theta_i - \theta_j)$ to approximate $\sin(\theta_i - \theta_j)$. Thus under normal operation the power flow between two nodes i and j is loosely proportional to the angle difference $(\theta_i - \theta_j)$. When line outage occurs at two connected nodes, the power flow between them will be disrupted, and we expect an abrupt change in $(\theta_i - \theta_j)$. Thus, we use the angle difference between connected node to capture the dynamic change caused by an outage. Moreover, since including $(\theta_i - \theta_j)$ of every line(i, j) as input variables leads to a high dimensional problem in large networks, for each node i we sum up the line information in its neighborhood for the purpose of dimension reduction, thus we define $\theta_i' = \sum_{(i,j)\in\mathcal{E}}(\theta_i - \theta_j)$, and use θ_i' as the input graph signal for classification.

Method Spatial uses θ_i' directly in the convolution operation, and method Spatial+DFT first performs a DFT on the time series at each node. Fig. 3 shows θ_1' and θ_2' against time and the resulting DFT magnitudes against frequency when a line outage occurs. The left column are when line(1,2) is out, θ_1' and θ_2' show a sharp change. The picture in Fig. 3(c) shows the corresponding high frequency caused by the oscillation. The right column serves as a comparison that when line$(3, 4)$ is out bus $1, 2$ are only slightly affected, thus the energy of the signal are mostly concentrated in low frequencies, as shown in Fig. 3(d).

Methods Spectral and Spectral-B both use a graph Fourier transform on input signal θ_i' before the convolution operation.



(a) normal (b) line(1,2) out

(c) normal (d) line(1,2) out

Fig. 4. Upper row: the phasor angle θ_i' from a subgraph. The color at each node reflects its value according to the scale on the right. Lower row: the magnitudes of graph spectral coefficients corresponding to the upper row.

Fig. 4 shows angles $\{\theta_i'\}$ from a subgraph of the power system. Fig. 4(a) is before line(1,2) is out, and (b) is right after line(1,2) is out. We observed that under normal operation, the angles vary slowly among local neighborhood, but the outage at line(1,2) causes high variation between nodes 1 and 2. The lower row shows the magnitudes of graph spectral coefficients resulting from the graph Fourier transform of the signal in the upper row. The eigenvalues of $\tilde{A}$ give a notion of frequency in the sense that the lowest eigenvalue corresponds to the highest frequency. Under normal operation, the energy of the signal is mostly distributed on low frequencies (see Fig. 4(c)). When line outage happens, we see that line outage causes high variation in nodes 1 and 2, which is reflected by the presence of energy in high frequencies in Fig. 4(d).

C. Model Parameters Setting

All the methods have a convolution layer, followed by a readout layer, a fully-connected layer and a softmax activation function for classification. Since we formulated the problem as a binary classification problem, a separate binary classifier is trained for each line outage identification. The softmax function outputs the predicted probability of line l being out. The response variable Y_l is a binary variable: $Y_l = 1$ if the input example has the lth line out and $Y_l = 0$ if not.

The parameters are set as follows. The dataset follows a 60%-20%-20% random split for training, validation, and testing, respectively. Adam optimizer is used for optimization, and exponential decay applied to the learning rate, with starting rate 0.001 and decay rate 0.9. Cross entropy is used for loss function. The number of epochs during training is 50 with 80 examples per batch.

D. Results

This section evaluates the performances of the proposed methods. For GCN from the spatial domain, we consider two versions: method *Spatial* that directly uses the graph signal as input, which is a $m \times n$ matrix for a graph with m nodes and time series of length n; method *Spatial+DFT* that first transforms the input graph signal by DFT before feeding into the convolution layer (see Fig. 1).

For GCN from the spectral domain, we also consider two versions: method *Spectral* as shown in Fig. 2, and *Spectral-B*, which is considered equivalent to method *Spatial*.

Table I shows the averaged results of 10 runs per line. Dataset are randomly split into training, validation, and testing sets in each run. To evaluate the model performance, we use recall and precision as in this problem a high recall (i.e., detection rate) is more important than the overall accuracy.

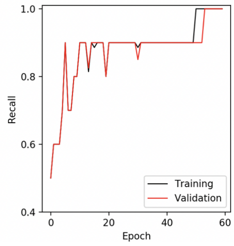
TABLE I
TEST RESULTS OF PROPOSED MODELS ON POWER SYSTEM LINE OUTAGE IDENTIFICATION

Method	System	$H=1$		$H=0$	
		Recall	Precision	Recall	Precision
Spatial+DFT	39	0.99	0.97	0.81	0.85
	118	0.98	0.97	0.91	0.74
Spatial	39	0.95	0.91	0.92	0.88
	118	0.94	0.83	0.91	0.66
Spectral	39	**0.99**	**0.98**	**0.98**	**0.94**
	118	**0.99**	**0.98**	**0.98**	**0.92**
Spectral-B	39	0.95	0.91	0.92	0.88
	118	0.93	0.85	0.91	0.66

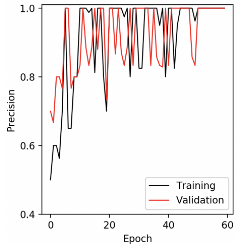
Remarks on Table I

- Results show that using signal values from one-hop neighborhood ($H = 1$) is an improvement over using the signal value from a node itself only ($H = 0$), and one-hop is sufficient to reach a satisfying performance as all methods can achieve at least 93% detection rate. This means convolution in both spatial domain and spectral domain can extract the important features that can be used to classify line status.
- GCN in the spectral domain (highlighted) significantly outperforms others on both small and large systems. This indicates filtering the signals in frequency domain and deriving high level features based on their spectral information is most effective in line outage identification.
- The performance of Spatial and Spectral-B are overall consistent. This further proved the equivalency of convolution in vertex domain and spectral domain. The small difference is caused by random initialization of weights, and numerical truncating error in the computation of Fourier transform and inverse Fourier transform.

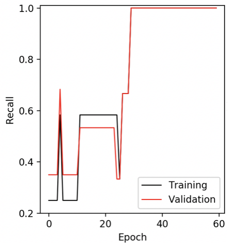
Figure 5 shows the recall and precision of each method during the training stage. Notice that the recall and precision of the training and validation sets during the training stage increases. This indicates that the models can efficiently learn



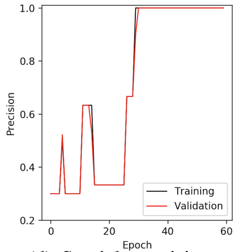
(a) Spatial+DFT, recall



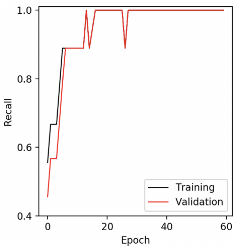
(b) Spatial+DFT, precision



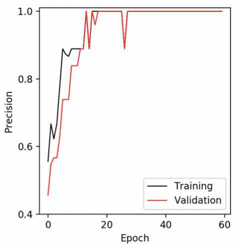
(c) Spatial, recall



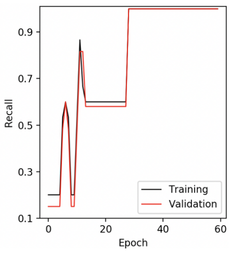
(d) Spatial, precision



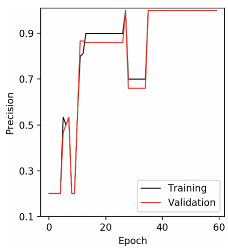
(e) Spectral, recall



(f) Spectral, precision



(g) Spectral-B, recall



(h) Spectral-B, precision

Fig. 5. Recall and precision during the training stage for line(1,2) outage identification in the 118-bus system. All the architectures have competitive performance, the recall and precision reach 100% within 60 epochs.

from the features. We also observed that although the models may start with a low recall or precision, they effectively increase these values within a short amount of epochs.

E. Robustness to Noisy Inputs

The input signal we have considered up to now is noise-free in the sense that the changes in the signal are only related to line outages, therefore it only suffers a significant oscillation when a line outage occurs, and it is relatively steady before the outage. In real life, the measurements can include small oscillations even under normal operation. This can be caused by dynamic loads at buses. The signal change caused by dynamic loads introduces noise in identifying the line outage, as shown in Fig. 6. To test the model robustness to such noisy input, we consider line outage cases when loads at buses vary. We use IEEE 39-bus and 118-bus systems as examples. We

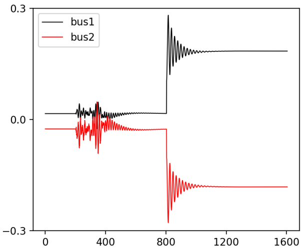


Fig. 6. Measurements θ_1' and θ_2' when line(1,2) is out in the 39-bus system. X-axis: time; Y-axis: measurements. The major oscillation is caused by line outage, and small oscillation before the outage is caused by dynamic loads.

set $H = 1$, and the other settings are the same as that used for Table I. The results are shown in Table II.

TABLE II
TEST RESULTS OF THE PROPOSED MODELS WITH NOISY INPUT. THE LOWER HALF SHOWS THE RESULTS FROM OTHER BASELINE METHODS.

Method	System	Recall	Precision
Spatial+DFT	39	0.94	0.93
	118	0.93	0.89
Spatial	39	0.95	0.9
	118	0.86	0.81
Spectral	39	**0.99**	**0.98**
	118	**0.98**	**0.98**
Spectral-B	39	0.95	0.9
	118	0.84	0.82
Conventional CNN	39	0.72	0.6
	118	0.55	0.51
LTSM	39	0.74	0.78
	118	0.73	0.73
OMP	39	0.9	0.85
	118	0.86	0.8

Table II shows that when considering input signals with noises caused by dynamic loads, *Spectral* is the most robust to perturbation and is still able to extract meaningful information. The performance of *Spatial+DFT* is degraded regardless of the size of the network. *Spatial* and *Spectral-B* remain stable for small systems, but are noticeably affected when the network becomes large. The winner of all is still method *Spectral*, which is robust to noise even for large networks. This again confirms its efficacy in extracting convolved features.

For comparison purpose, we added in Table II the results of three other methods: the conventional CNN, Long Short Term Memory network (LSTM), and Orthogonal Matching Pursuit (OMP) that is used in [3].

- OMP uses two sets of the measurements data, taken before and after the line outage, respectively. CNN and LSTM use time series of voltage phasor angles as input, same as the proposed GCNs.
- The architecture for conventional CNN is a convolutional layer with filter size 3×3, followed by a max-pooling layer with filter size 2×2, a fully connected layer and an output layer.

- The architecture for the LSTM model is an LSTM layer followed by a dropout layer, a fully connected layer and an output layer. For each node we use the measurements from its one hop neighbors, and classify which line is out from all of its connected lines.

While GCNs achieve comparable performance with OMP in a noise-free setting, they are more effective than OMP in a noisy environment. The GCNs are much more effective than CNN and LSTM since these methods are not suitable for capturing the network-wide spatial feature. The proposed GCNs, especially the spectral GCN, are most effective to capture the features in spatiotemporal data defined on graphs.

VI. Related Work

Recently machine learning algorithms have been studied widely for power line outage identification problem. [10] developed a binary classifier by building neural networks using data collected from power grids, in which a classifier is trained to return the Maximum a Posteriori probability that indicates the line status. [11] also employs neural networks for outage identifications. However, rather than extracting features directly from PMU data, it trains the neural network using an AC model. Other machine learning classifiers include support vector machine [12], random forest and logistic regression [9].

GCNs are built upon the success of graph signal processing ([1], [2], [6], [7], [13]) and spectral graph theory ([2], [14]–[17]). In [1], the authors gave a tutorial overview of methods that generalized fundamental signal processing operations such as filtering, translation, downsampling, etc. Our work is based on these fundamental operations. Among many others, [18] is a representative work that uses graph signal processing for machine learning, which developed two graph convolutional architectures based on convolution in the vertex domain. The first method, selection graph neural network, generalizes conventional CNN by replacing linear shift invariant filter with graph shift invariant filter. The second method, aggregate graph neural network, aims to create a sequence of signals through successive applications of the shift operator to the original signal, then the conventional CNN can be used on the sequence of signals as on regular grid data. [15] generalizes the convolution layer by first performing multi-scale clustering of the graph, then defines the convolution for every locally connected network. Our method, to our knowledge, is by far the first work that uses spectral information to develop a convolution layer.

VII. Conclusions

We develop two fundamental convolutional architectures to process signals supported on graphs. In the spatial GCN the convolution filter is defined as a degree H polynomial of the graph shift operator matrix, thus the convolved feature at a node incorporates the signal value of every node within its H-hop neighborhood. In the spectral GCN the convolution filter is defined as a polynomial of the graph frequencies. This allows us to filter the signal in the frequency domain.

The models have been tested on the problem of line outage identification in IEEE 39-bus and 118-bus systems. The results show that all the methods perform well for the purpose of line outage identification, among which the method *Spectral* outperforms the other methods and is also the most robust to noisy inputs. GCNs use only measurement data but produce results comparable to those that use deep knowledge of power system models and parameters and significantly outperform other neural network models such as CNN and LSTM.

References

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, pp. 83–98, May 2013.

[2] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[3] H. Zhu and G. B. Giannakis, "Sparse overcomplete representations for efficient identification of power line outages," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 2215–2224, 2012.

[4] J. E. Tate and T. J. Overbye, "Line outage detection using phasor angle measurements," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1644–1652, 2008.

[5] Y. Zhao, J. Chen, A. Goldsmith, and H. V. Poor, "Identification of outages in power systems with uncertain states and optimal sensor locations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1140–1153, 2014.

[6] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[7] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[8] F. Milano, L. Vanfretti, and J. C. Morataya, "An open source power system virtual laboratory: The psat case and experience," *IEEE Transactions on Education*, vol. 51, no. 1, pp. 17–23, 2008.

[9] J. He, M. X. Cheng, Y. Fang, and M. L. Crow, "A machine learning approach for line outage identification in power systems," in *International Conference on Machine Learning, Optimization, and Data Science*, pp. 482–493, Springer, 2018.

[10] Y. Zhao, J. Chen, and H. V. Poor, "Efficient neural network architecture for topology identification in smart grid," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 811–815, IEEE, 2016.

[11] C.-p. Lee and S. J. Wright, "Using neural networks to detect line outages from pmu data," *arXiv preprint arXiv:1710.05916*, 2017.

[12] A. Abdelaziz, S. Mekhamer, M. Ezzat, and E. El-Saadany, "Line outage detection using support vector machine (svm) based on the phasor measurement units (pmus) technology," in *2012 IEEE Power and Energy Society General Meeting*, pp. 1–8, IEEE, 2012.

[13] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs: Graph filters," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6163–6166, IEEE, 2013.

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

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

[16] A. Agaskar and Y. M. Lu, "A spectral graph uncertainty principle," *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4338–4356, 2013.

[17] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[18] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional neural network architectures for signals supported on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 1034–1049, 2018.