Fault Location in Power Distribution Systems via Deep Graph Convolutional Networks

Kunjin Chen, Jun Hu, Yu Zhang, *Member, IEEE*, Zhanqing Yu, *Member, IEEE*, and Jinliang He, *Fellow, IEEE*

***Abstract*—This paper develops a novel graph convolutional network (GCN) framework for fault location in power distribution networks. The proposed approach integrates multiple measurements at different buses while takes system topology into account. The effectiveness of the GCN model is corroborated by the IEEE 123-bus benchmark system. Simulation results show that the GCN model significantly outperforms other widely-used machine learning schemes with very high fault location accuracy. In addition, the proposed approach is robust to measurement noise and errors, missing entries, as well as multiple connection possibilities. Finally, data visualization results of two competing neural networks are presented to explore the mechanism of GCN's superior performance.**

***Index Terms*—Fault location, distribution systems, deep learning, graph convolutional network.**

I. Introduction

Distribution systems are constantly under the threat of short-circuit faults that would cause power outages. In order to enhance the operation quality and reliability of distribution systems, system operators have to deal with outages in a timely manner. Thus, it is of paramount importance to accurately locate and quickly clear faults immediately after the occurrence, so that quick restoration can be achieved.

Existing fault location techniques in the literature can be divided into several categories, namely, impedance-based methods [1–3], voltage sag-based methods [4–6], automated outage mapping [6–8], traveling wave-based methods [9, 10], and machine learning-based methods [11–14]. Impedance-based fault location methods use voltage and current measurements to estimate fault impedance and fault location. Specifically, a generalized fault location method for overhead distribution system is proposed in [1]. Substation voltage and current quantities are expressed as functions of the fault location and fault resistance, thus the fault location can be determined by solving a set of nonlinear equations. To solve the multiple estimation problem, it is proposed to use estimated fault currents in all phases including the healthy phase to find the faulty feeder and the location of the fault [2]. It is pointed out in [15] that the accuracy of impedance-based methods can be affected by factors including fault type, unbalanced loads, heterogeneity of overhead lines, measurement errors, etc.

When a fault occurs in a distribution system, voltage drops can occur at all buses. The voltage drop characteristics for the whole system vary with different fault locations. Thus, the voltage measurements on certain buses can be used to identify the fault location. For instance, calculated fault currents can be applied to each bus in the system, and the values of voltage drop on a small number of buses can be obtained by calculating the power flows. The fault location can then be determined by comparing measured and calculated values of voltage drop [4, 5]. In [6], multiple estimations of fault current at a given bus are calculated using voltage drop measurements on a small number of buses, and a bus is identified as the faulty bus if the variance of the multiple fault current estimates takes the smallest value.

Automatic outage mapping refers to locating a fault or reducing the search space of a fault using information provided by devices that can directly or indirectly indicate the fault location. For example, when a fault occurs, if an automatic recloser is disconnected, smart meters downstream of the device would experience an outage. Smart meters downstream of the fault itself will also feature a loss of power. Thus, the search space of the fault can be greatly reduced if the geographic location of each smart meter is considered [6]. Authors in [7] proposed to use fault indicators to identify the fault location. Each fault indicator can tell whether the fault current flows through itself (it may also have the ability to tell the direction of the fault current). The location of the fault can then be narrowed down to a section between any two fault indicators. An integer programming-based method is proposed in [8] to locate a fault using information from circuit breakers, automatic reclosers, fuses, and smart meters. Multiple fault scenarios, malfunctioning of protective devices, and missing notifications from smart meters are also taken into consideration.

Traveling wave-based methods use observation of original and reflected waves generated by a fault. Specifically, different types of traveling wave methods include single-ended, double-ended, injection-based, reclosing transient-based, etc. The principle and implementation of single-ended and double-ended fault location with traveling waves are discussed in [9]. The traveling wave generated by circuit breaker reclosing is used to locate faults in [10]. In general, however, traveling wave-based methods require high sampling rates and communication overhead of measurement devices [5]. Systems such as the global positioning system (GPS) are required for time synchronization across multi-terminal signals.

Manuscript received in December 2018. This work was supported in part by National Key R&D Program of China under Grant 2018YFB0904603, Natural Science Foundation of China under Grant 51720105004, State Grid Corporation of China under Grant 5202011600UJ, and the Faculty Research Grant (FRG) of University of California, Santa Cruz.

K.-J. Chen, J. Hu, Z.-Q. Yu, and J.-L. He are with the State Key Lab of Power Systems, Dept. of Electrical Engineering, Tsinghua University, Beijing 100084, P. R. of China.

Y. Zhang is with the Dept. of Electrical and Computer Engineering, University of California, Santa Cruz, CA 95064, USA.

(Corresponding author email: hejl@tsinghua.edu.cn).

Machine learning models are leverage for fault location in distribution systems [16]. Using the spectral characteristics of post-fault measurements, data with feature extraction are fed into an artificial neural network (ANN) for fault location [13]. A learning algorithm for multivariable data analysis (LAMDA) is used in [12] to obtain fault location. Descriptors are extracted from voltage and current waveforms measured at the substation. Various LAMDA nets are trained for different types of faults. In [11], the authors first use support vector machines (SVM) to classify the fault type, and then use ANN to identify the fault location. Smart meter data serves as the input of a multi-label SVM to identify the faulty lines in a distribution system [14].

The deployment of distribution system measurement devices or systems such as advanced metering infrastructure [14], micro phasor measurement units [17], and wireless sensor networks [18] improve data-driven situational awareness for distribution systems [19, 20]. In addition, recent advances in the field of machine learning, especially deep learning, have gained extensive attentions. One of the major developments is the successful implementation of convolutional neural networks (CNN) in a variety of image recognition-related tasks [21]. While the measurements on different buses in a power distribution system are spatially distributed, it is hard to directly implement a CNN model that use such measurements as input. Nevertheless, when multiple buses in a distribution system become measurable, it is possible to treat the measurements as signals on a graph to which variants of traditional data analysis tools may be applicable [22, 23]. As an extension of CNNs for data on graphs, graph convolutional networks (GCN) have been designed and implemented [24–26]. In this paper, a GCN model is proposed for fault location in distribution systems. Data feature extraction is fulfilled jointly with the faulty bus classification task. Meawhile, spatial correlations of the buses can be automatically taken into account. In addition, the proposed model can be readily adapted or extended to various tasks concerning data processing for multiple measurements in modern power systems.

II. Fault Location Based on Graph Convolutional Networks

In this section, we first give a brief description of the fault location task. Next, we will revisit idea of spectral convolution on graphs, and show how a GCN can be constructed based on that idea. Finally, we will present the test case of the IEEE 123-bus distribution system.

A. Formulation of the Fault Location Task

In this paper, we assume that the voltage and current phasor measurements are available at phases that are connected to loads. That is, for a given measured bus in a distribution system, we have access to its three-phase voltage and current phasors $(V_1, \theta_1^V, V_2, \theta_2^V, V_3, \theta_3^V, I_1, \theta_1^I, I_2, \theta_2^I, I_3, \theta_3^I) \in \mathbb{R}^{12}$. Values corresponding to unmeasured phases are set to zero. A data sample of measurements from the distribution system can then be represented as $\mathbf{X} \in \mathbb{R}^{n_o \times 12}$, where n_o is the number of observed buses. We formulate the fault location task as a classification problem. More specifically, given a data sample matrix $\mathbf{X}_i$, the faulted bus $\tilde{y}_i$ is obtained by $\tilde{y}_i = f(\mathbf{X}_i)$, where f is a specific faulty bus classification model. A fault is correctly located if $\tilde{y}_i = y_i$, where y_i indicates the true faulty bus corresponding to $\mathbf{X}_i$.

B. Spectral Convolution on Graphs

To be self-contained, we first present a brief introduction to spectral graph theory [27]. Suppose we have an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V}$ is the set of vertices with $|\mathcal{V}| = n$, $\mathcal{E}$ is the set of edges, and $\mathbf{W} \in \mathbb{R}^{n\times n}$ is the weighted adjacency matrix. The unnormalized graph Laplacian of $\mathcal{G}$ is defined as $\mathbf{\Delta}_u = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the degree matrix of the graph with diagonal entries $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. Then, the normalized graph Laplacian is given as

$$\mathbf{\Delta} = \mathbf{D}^{-1/2}\mathbf{\Delta}_u\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \tag{1}$$

where $\mathbf{I}$ is the identity matrix. The eigendecomposition of the positive semi-definite symmetric matrix $\mathbf{\Delta}$ yields $\mathbf{\Delta} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^\top$, where $\mathbf{\Phi} = (\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_n)$ are orthonormal eigenvectors of $\mathbf{\Delta}$, and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ is the diagonal matrix with corresponding ordered non-negative eigenvalues $(0 = \lambda_1 \leq \lambda_2, \dots, \lambda_n)$.

By analogy with the Fourier transform in Euclidean spaces, graph Fourier transform (GFT) can be defined for weighted graphs using the orthonormal eigenvectors of $\mathbf{\Delta}$ [28]. For a signal $\mathbf{f} \in \mathbb{R}^n$ on the vertices of graph $\mathcal{G}$ (each vertice has one value in this case), GFT is performed as $\hat{\mathbf{f}} = \mathbf{\Phi}^\top\mathbf{f}$ while the inverse GFT is $\mathbf{f} = \mathbf{\Phi}\hat{\mathbf{f}}$. Further, we can conduct convolution on graphs in the spectral domain also by analogy with convolution on discrete Euclidean spaces facilitated by Fourier transform. That is, spectral convolution of two signals $\mathbf{g}$ and $\mathbf{f}$ is defined as

$$\mathbf{g} * \mathbf{f} = \mathbf{\Phi}\left((\mathbf{\Phi}^\top\mathbf{g}) \circ (\mathbf{\Phi}^\top\mathbf{f})\right) = \mathbf{\Phi}\,\mathrm{diag}(\hat{g}_1, \dots, \hat{g}_n)\mathbf{\Phi}^\top\mathbf{f} \tag{2}$$

where $\circ$ indicates element-wise product between two vectors. Filtering of signal $\mathbf{f}$ by spectral filter $\mathbf{B} = \mathrm{diag}(\boldsymbol{\beta})$ with $\boldsymbol{\beta} \in \mathbb{R}^n$ can then be expressed as $\mathbf{\Phi}\mathbf{B}\mathbf{\Phi}^\top\mathbf{f}$. One major drawback of this formulation, however, is that the filters are not guaranteed to be spatially localized, which is a crucial feature of CNNs for data in Euclidean spaces, since localized filters are able to extract features from small areas of interest instead of the whole input. Using filters $h_{\boldsymbol{\alpha}}(\mathbf{\Lambda})$ that are smooth in spectral domain can bypass such an issue [27, 29]. For example, consider using a polynomial approximation

$$h_{\boldsymbol{\alpha}}(\mathbf{\Lambda}) = \sum_{k=0}^{K} \alpha_k \mathbf{\Lambda}^k \tag{3}$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$ is the vector of coefficients to be learned for the filters. Further, as the computation of $\mathbf{\Phi}h_{\boldsymbol{\alpha}}(\mathbf{\Lambda})\mathbf{\Phi}^\top$ has complexity $\mathcal{O}(n^2)$, truncated Chebyshev polynomial expansion of $h_{\boldsymbol{\alpha}}(\mathbf{\Lambda})$ is introduced to reduce the complexity of the filtering operation [25, 28]. Specifically, expansion of $h_{\boldsymbol{\alpha}}(\mathbf{\Lambda})$ using Chebyshev polynomials $T_k(\tilde{\mathbf{\Lambda}})$ up to order K can be expressed as

$$h_{\boldsymbol{\alpha}}(\mathbf{\Lambda}) = \sum_{k=0}^{K} \alpha_k T_k(\tilde{\mathbf{\Lambda}}) \tag{4}$$

where $\tilde{\mathbf{\Lambda}} = 2\mathbf{\Lambda}/\lambda_n - \mathbf{I}$. The recursive formulation of the filtering process based on Chebshev polynomials is introduced in [25], which takes the form $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0 = 1$ and $T_1 = x$. Since $\mathbf{\Delta}^k = (\mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^\top)^k = \mathbf{\Phi}\mathbf{\Lambda}^k\mathbf{\Phi}^\top$, the filtering process $\mathbf{\Phi}h_{\boldsymbol{\alpha}}(\mathbf{\Lambda})\mathbf{\Phi}^\top\mathbf{f}$ can be expressed as

$$\mathbf{\Phi}h_{\boldsymbol{\alpha}}(\mathbf{\Lambda})\mathbf{\Phi}^\top\mathbf{f} = h_{\boldsymbol{\alpha}}(\mathbf{\Delta})\mathbf{f} = \sum_{k=0}^{K} \alpha_k T_k(\tilde{\mathbf{\Delta}})\mathbf{f} \tag{5}$$

where $\tilde{\mathbf{\Delta}} = 2\mathbf{\Delta}/\lambda_n - \mathbf{I}$. Consequently, with $\mathbf{d}_0 = \mathbf{f}$ and $\mathbf{d}_1 = \tilde{\mathbf{\Delta}}\mathbf{f}$, we can recursively calculate $\mathbf{d}_k = 2\tilde{\mathbf{\Delta}}\mathbf{d}_{k-1} - \mathbf{d}_{k-2}$, and the filtering operation $h_{\boldsymbol{\alpha}}(\mathbf{\Delta})\mathbf{f} = [\mathbf{d}_0, \cdots, \mathbf{d}_K]\boldsymbol{\alpha}$ has a computational complexity of $\mathcal{O}(K|\mathcal{E}|)$ considering the sparsity of $\mathbf{\Delta}$ [25]. In addition, because the Chebyshev polynomials are truncated to the Kth order, the filter is K-hop localized with respect to the connections embodied in $\mathbf{\Delta}$. To this end, GCN can be implemented with the aforementioned spectral convolution on graphs.

C. GCN Approach for Fault Location

The GCN model applied to the fault location task is illustrated in Fig. 1. The input $\mathbf{X}$ is passed through L_c graph convolution layers and L_f fully-connected layers followed by a softmax activation function. Specifically, the jth feature map of a graph convolution layer is calculated as

$$\mathbf{y}_j = \sum_{i=1}^{N_{in}} h_{\boldsymbol{\alpha}_{i,j}}(\mathbf{\Delta})\mathbf{x}_i \tag{6}$$

where $\mathbf{x}_i \in \mathbb{R}^n$ is the ith input feature map, $\boldsymbol{\alpha}_{i,j} \in \mathbb{R}^K$ is the trainable coefficients, and N_{in} is the number of filters of the previous layer. With N_{out} filters in the current layer, a total of $N_{in}N_{out}K$ parameters are trainable in this layer. In particular, $F_{in} = 12$ for the first layer of the model. The output of the last graph convolution layer is flattened into a vector and passed to the fully-connected layers. The index of the predicted faulty bus, $\tilde{y}$, can be obtained as $\tilde{y} = \operatorname{argmax}_i a_i$, where a_i is the ith activation of the last fully-connected layer.

The weighted adjacency matrix is constructed based on the physical distance between the nodes. First, the distance matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ is formed with $\mathbf{S}_{ij}$ being the length of the shortest path between bus i and bus j. We then sort and keep the smallest K_n values in each row of $\mathbf{S}$ to obtain $\tilde{\mathbf{S}} \in \mathbb{R}^{n\times K_n}$ and calculate $\sigma_S = \sum_i \tilde{\mathbf{S}}_{iK_n}/n$ (we have $\tilde{\mathbf{S}}_{ij} \leq \tilde{\mathbf{S}}_{ik}$ for $j < k$). Matrix $\tilde{\mathbf{W}} \in \mathbb{R}^{n\times K_n}$ is then constructed with $\tilde{\mathbf{W}}_{ij} = e^{-\tilde{\mathbf{S}}_{ij}^2/\sigma_S^2}$. By restoring the positional correspondence of $\tilde{\mathbf{W}}_{ij}$ to bus i and bus j, the weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ can be obtained. We can thus proceed to compute $\mathbf{D}$ and finally obtain $\mathbf{\Delta}$ according to (1).

D. The IEEE 123 Bus Distribution System Test Case

The IEEE 123 bus test case is used to carry out the task of fault location in distribution systems in this paper [30]. The overall topology of the distribution system is illustrated in Fig. 2. Note that the topology is only used to indicate the connections of the buses rather than their geometrical locations. Specifically, there are 128 buses in the system (cf. Fig. 2), 85 of which are connected to loads. Most of those loads are only connected to a single phase. Bus pairs (149, 150r), (18, 135), (13, 152), (60, 160(r)), (61, 61s), and (97, 197) are connected by normally closed switches. In addition, regulators are installed at buses 9, 25, and 160.

In order to generate the training and test datasets, faults are simulated for all buses in the system. Three types of faults are considered, namely, single phase to ground, two phase to ground, and two phase short-circuit. Two phase short-circuit faults have a constant 0.1 Ω fault resistance. The other two types of faults have the resistance ranging from 0.05 Ω to 20 Ω. The load level of the system varies between 0.316 and 1. Fig. 3 shows the discrete probability density function (PDF) with 50 equal-length load level intervals. The PDF is obtained from the annual load curve of the system. We randomly sample one value from the load level distribution and set all loads in the system to the same level. The simulations are implemented by the OpenDSS software [31]. The voltage and current phasors are measured during the fault. We obtain the training and test sets used for training and evaluating the fault location models.

We generate 20 data samples for each fault type at each bus. As a result, a total of 13520 data samples are generated for both the training and test datasets. We consider buses connected with normally closed switches or regulators as a single bus. Thus, there are a total of 119 faulty buses to be classified; i.e., 119 class labels for the classification task.

For the implementation of the GCN model[1], instead of using $n_o \times 12$ as the size of the input of the model, we expand $\mathbf{X}$ to include all 128 buses, i.e., each input data sample has a size of 128×12. As a result, each sample matrix has 1536 entries, 380 of which have measured values. For the non-measured buses, we set the corresponding values to be zero. The same measured quantity is run through the standardization process; i.e., subtracting the mean and dividing by the standard deviation.

III. Results and Discussion

In this section, we report the performance of GCN for fault location tested in the IEEE 123-bus benchmark system. Comparisons with baseline models are provided in detail. We also visualize the hidden features of samples in the test dataset to demonstrate that the proposed GCN model is able to learn more robust representations from data.

A. Implementation Details and Baseline Models

The hyper-parameters of the GCN model implemented in this paper are determined using 10% of the training dataset

[1] The implementation of GCN in this paper is modified from the implementation in [25], code available at: https://github.com/mdeff/cnn_graph

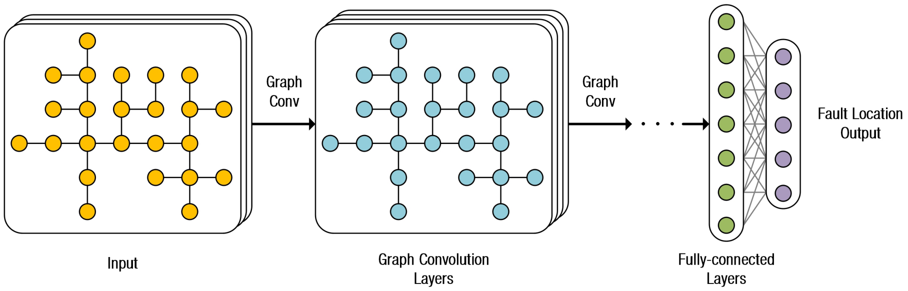


Fig. 1. The structure of the GCN model. Several graph convolution layers are followed by two fully-connected layers.

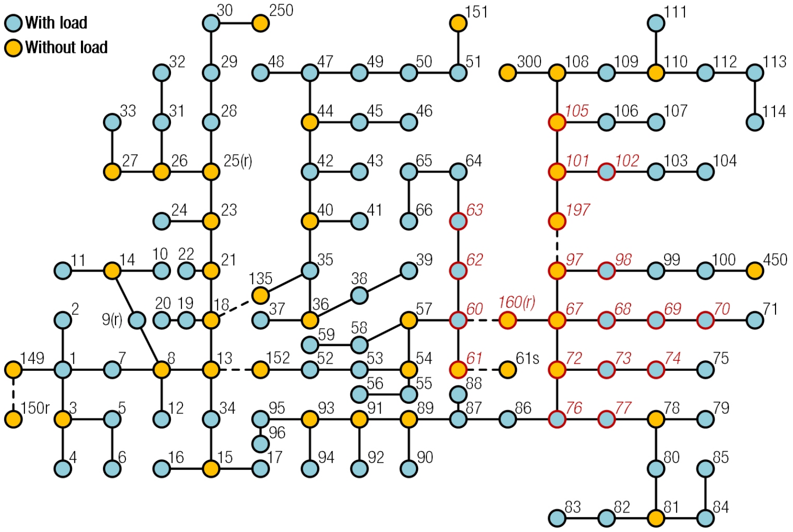


Fig. 2. An illustration of the IEEE 123 bus system. It is assumed that voltage and current phasors of PQ buses (connected to loads) are measured. As an example, the 20 buses that are closest (in distance) to bus 67 are highlighted with red color and italic numbers. Normally closed switches are represented by dashed lines.

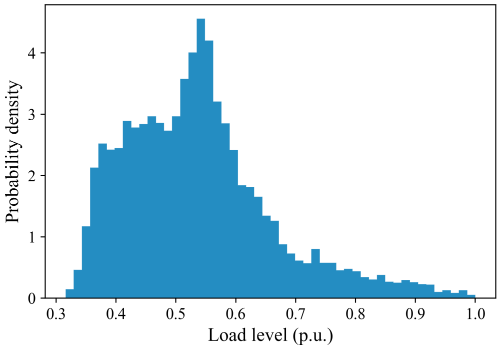


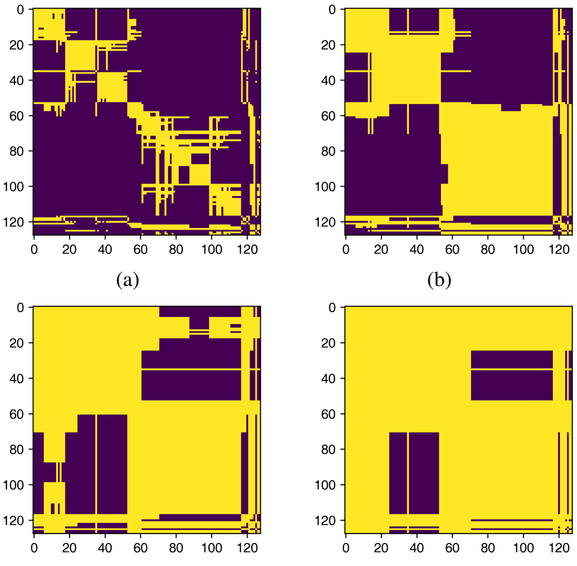
Fig. 3. The probability density of load level for the distribution system.

as the validation set. Specifically, the model has 3 graph convolution layers (all with 256 filters) followed by 2 fully-connected layers (with 512 and 256 hidden nodes). K_n is set to 20 and the values of K for the graph convolution layers are 3, 4, and 5, respectively. The two fully-connected layers have a dropout rate of 0.5. The Adam optimizer with an initial learning rate of 0.0002 is used to train the model for 400 epochs and a batch size of 32. We use Tensorflow in Python to implement the GCN model.

We first visualize $\mathbf{\Delta}^m$ with different values of m to illustrate the locality of the spectral filters, the results of which are shown in Fig. 4 and Fig. 5. In Fig. 4, we illustrate the support of a filter when m ranges from 1 to 4 (when $m = 5$, the support of filters becomes the whole graph). In Fig. 4, the absolute values of the entries in $\mathbf{\Delta}^m$ are visualized. Although the size of filters grows fast with the increase of m, we can observe in Fig. 5 that relatively large absolute values in $\mathbf{\Delta}^m$ are mainly limited to entries corresponding to bus pairs that are close to each other. Since the filters can be represented as polynomials of $\mathbf{\Delta}$, we conclude that the locality of filters are ensured the value of K_n is chosen properly. Note that higher-order terms in the polynomials facilitate the filters to explore more nodes in the graph.

Three baseline models are also implemented for comparison:

1) SVM: The dimensionality of the measurements is reduced to 200 by principal component analysis (PCA). The radial basis function (RBF) kernel is used for the SVM with $\gamma = 0.002$ and $C = 1.5 \times 10^6$. LibSVM [32] in Python is used for the implementation in this paper.
2) Random forest (RF): The dimensionality of the measurements is also reduced to 200 by PCA. The number of trees is set to 300, the minimal number of samples per leaf is 1, while the minimal number of samples required for a split is set to 3.
3) Fully-connected neural network (FCNN): A three-layer



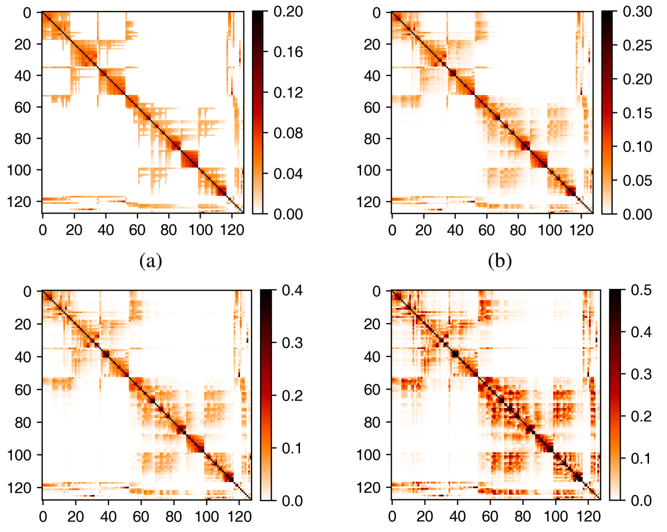
(a)

(b)

(c)

(d)

Fig. 4. Visualization of non-zero entries (yellow) in $\mathbf{\Delta}^m$ when $K_n = 20$ (20 nearest buses of bus 67 are red colored as shown in Fig. 2): (a) $m = 1$, (b) $m = 2$, (c) $m = 3$, and (d) $m = 4$.



(a)

(b)

(c)

(d)

Fig. 5. Visualization of $\mathbf{\Delta}^m$ when $K_n = 20$, (a) $m = 1$, (b) $m = 3$, (c) $m = 5$, and (d) $m = 10$. The indexes of the buses are sorted according to the order of bus numbers shown in Fig. 2. Absolute values of the entries in $\mathbf{\Delta}^m$ are visualized. Entries with values greater than certain thresholds (i.e., 0.2, 0.3, 0.4, and 0.5 for (a), (b), (c), and (d), respectively) have the same color.

FCNN is implemented as a vanilla baseline of neural networks. The numbers of hidden neurons for the three layers are 256, 128, and 64, respectively. Scaled exponential linear unit (SELU) is used as the activation function.

The hyper-parameters for SVM and RF are determined by 5-fold cross-validation. For the FCNN model, 10% of the training data is used to validate the hyper-parameters.

TABLE I
Fault Location Accuracies of Different Approaches

Model	Accuracy	One-hop Accuracy
PCA + SVM	94.60	98.31
PCA + RF	94.96	99.28
FCNN	84.64	96.38
GCN	**99.26**	**99.93**

In order to justify the effectiveness of our proposed approach in real-world conditions, we add noise and errors to the measurements and compare the performance of different models. More specifically, three types of modifications of measurements are added:

1) Gaussian noise: We add Gaussian noises to the data so that the SNR is 45 dB [33].
2) Data loss of buses: We randomly drop the data of N_{drop} buses (i.e., set the measured values to 0) per data sample in the test dataset.
3) Random data loss for measured data: Each measurement at all buses is replaced by 0 with a probability P_{loss}.

More specifically, we set $N_{drop} = 1$ and $P_{loss} = 0.01$ throughout the experiments. The detailed performance comparisons are given in the ensuing subsections.

B. *Fault Location Performance of the Models*

The fault location accuracies of various approaches are presented in Table I. In addition to the traditionally defined classification accuracy, we also use one-hop accuracy as a metric to measure the performance of the models. Specifically, a sample is considered correctly classified if the predicted faulty bus is directly connected to the actual faulty bus. For the GCN model, we repeat the trials three times and report the mean of the accuracy values.

In Table I, it is shown that the GCN model has the highest classification accuracy. PCA and RF (both with PCA) also have good performance, especially for one-hop accuracy. The accuracy obtained by FCNN is relatively low, but its one-hop accuracy is still satisfactory.

The performance of the models with measurement modifications on the test dataset are shown in Table II. Results corresponding to the individual and combined modifications are reported therein. A major observation is that the two data loss errors greatly lower the classification accuracy of the models. Nevertheless, the GCN model is quite robust to various modifications and significantly outperform other schemes. In addition, the FCNN model has higher accuracy than SVM and RF when data loss errors are involved, even though its classification accuracy is roughly 10% lower than those two models.

A more realistic setting is adding Gaussian noise to the data samples in the training dataset and observe the performance of the models. Table III gives the results of fault location accuracy corresponding to such a setup. The results for SVM and FCNN are in general consistent with the accuracy values in Table II. For RF, however, the accuracies for modifications including data loss errors all increase by more than 10%. Mild improvements are also observed for GCN. In summary, the GCN model has superior performance when measurement modifications are added to the data. Note that the modifications with data loss errors are not taken into account in the training

TABLE II
FAULT LOCATION ACCURACIES OF THE MODELS UNDER VARIOUS MEASUREMENT MODIFICATIONS

Model	Noise (I)	Bus (II)	Random (III)	I + II	I + III	II + III	I + II + III
PCA + SVM	89.13 / 97.30	58.73 / 79.97	61.43 / 81.78	57.76 / 79.26	60.33 / 81.09	45.44 / 69.64	44.87 / 69.20
PCA + RF	85.94 / 96.77	53.82 / 67.62	58.57 / 74.07	52.15 / 66.66	56.94 / 73.23	40.55 / 55.84	40.05 / 55.64
FCNN	85.72 / 95.95	62.61 / 82.93	69.40 / 88.09	61.24 / 82.47	69.51 / 87.96	53.54 / 76.42	54.12 / 76.83
GCN	**97.10** / **99.72**	**92.67** / **97.44**	**89.09** / **96.67**	**90.76** / **97.83**	**87.70** / **96.31**	**83.55** / **94.51**	**80.63** / **93.86**

TABLE III
FAULT LOCATION ACCURACIES OF THE MODELS UNDER VARIOUS MEASUREMENT MODIFICATIONS WHEN TRAINED WITH NOISY DATA

Model	Noise	Noise + Bus	Noise + Random	All Combined
PCA + SVM	85.70 / 96.21	55.98 / 77.74	58.00 / 80.17	44.12 / 68.51
PCA + RF	86.51 / 97.55	64.11 / 81.61	66.12 / 84.90	52.34 / 72.13
FCNN	86.95 / 97.19	61.95 / 82.58	70.32 / 88.52	53.98 / 76.55
GCN	**97.52** / **99.73**	**92.67** / **98.26**	**88.76** / **96.44**	**84.53** / **94.77**

phase. This indicates that the robustness of the GCN model may be generalizable to other types of errors in the data.

In the next subsection, we visualize the data upon the transformation by the FCNN model and the GCN model. Such visualizations facilitate our understanding of the performance differences induced by various schemes.

C. Visualization of Data After Transformations

Visualizing transformed data in two-dimensional spaces enables assessment of the ability of the models to extract useful information from the input data. In this paper, we use t-distributed stochastic neighbor embedding (t-SNE) with two components to visualize high-dimensional data [34]. Specifically, t-SNE is used to investigate the local structure of the input data (i.e., normalized raw measurements), the data transformed by FCNN, and the data transformed by GCN. In particular, we are interested in studying how closely the samples corresponding to the same faulty bus are distributed.

In Fig. 6, we visualize the data samples in the test dataset with t-SNE after the dimensionality of data is reduced to 200 by PCA, which is also used to speed up the calculation process of t-SNE). In order to highlight the distribution of data belonging to the same class (faulty bus), 6 groups of data samples of bus 1, 21, 66, 85, 111, and 250 are marked with colors. Data samples of other buses are plotted as the gray dots. It can be seen in the figure that the dots of different colors scatter around such that it is hard to separate the data samples from different classes.

We then visualize the data samples in the test dataset after they are transformed by the FCNN and the GCN models, as shown in Fig. 7. Both models are trained with added Gaussian noise while the test data is also added with Gaussian noise. We extract the data from the outputs of the fully-connected layer right before the final output layer. For the FCNN model, each data sample is 64-dimensional, while the dimensionality of data samples is 256 for the GCN model. In Fig. 7(a), the data samples of the same class hardly cluster together, except for the dark green dots in the upper-right corner. In Fig. 7(b), however, most samples of the same color appear closely together, except that only a small fraction of blue dots are separated from its main cluster. Note that the visualization in Fig. 7 corresponds to the "Noise" column of Table III. That is, the improved feature extraction capability of the GCN model gives a performance boost in classification accuracy of more than 10%.

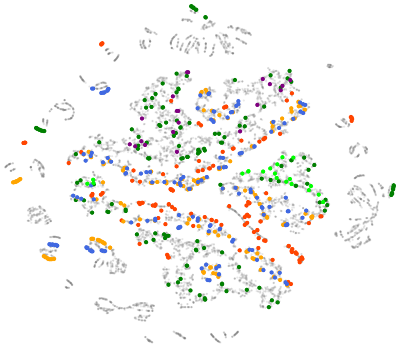
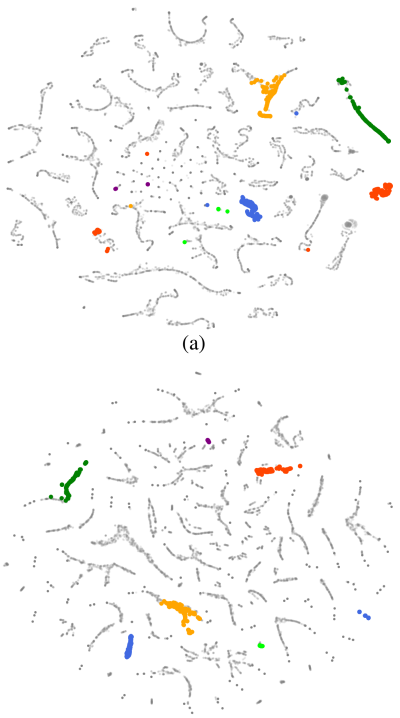


Fig. 6. Visualization of test data processed by PCA with 200 components and t-SNE with two components. Dots with the same color (except for small gray dots) correspond to the same faulty bus.

As shown in Table II and Table III, the two types of data loss errors have significant impact on the classification performance of all models. Thus, in Fig. 8 we proceed to visualize the data samples that are added with Gaussian noise and two types of data loss errors. A lot of small sample clusters of the six colored faulty buses can be seen at multiple locations in Fig. 8(a), which indicates that the FCNN model has difficulty in generalizing its feature extraction capability to the data modified with the two types of data loss errors. On the contrary, the GCN model still preserves the structures of the data to a large extend. The proportion of data samples that are separated from the main clusters is relatively small. Such a capability of preserving data structure gives rise to more than 30% performance gain for the proposed GCN, as shown in the last column of Table III.

D. Performance Under Multiple Connection Scenarios of Branches

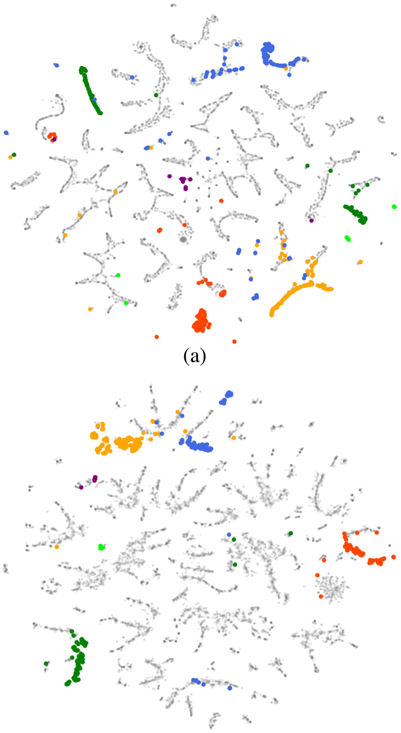
In this subsection, the performance of the models under multiple connection scenarios of several branches is examined.



(a)

(b)

Fig. 7. Visualization of hidden features of test data added with Gaussian noise using t-SNE with two components: (a) the FCNN model, and (b) the GCN model. The models are trained with added Gaussian noise.



(a)

(b)

Fig. 8. Visualization of hidden features of test data added with all three types of modifications using t-SNE with two components: (a) the FCNN model, and (b) the GCN model. The models are trained with added Gaussian noise.

Three branches are considered: i) branch connecting bus 36, 38, and 39; ii) branch connecting bus 67 to 71; and iii) branch connecting bus 108 to 114. Phase 1 and 2 of bus 36, and all three phases of bus 67 and 108 are connected to the distribution system. The buses on the branches use only one of the phases for connection.

TABLE IV
Fault Location Accuracies of the Models with Additional Data Generated With Changed Phases of Chosen Branches

Model	All Buses	Modified Buses
PCA + SVM	81.87 / 94.09	77.47 / 92.00
PCA + RF	81.57 / 95.11	79.64 / 93.93
FCNN	85.38 / 96.20	82.85 / 95.36
GCN	**97.65** / **99.77**	**93.66** / **99.38**

We implement a simple data generation process in order to add data with changed phases of aforementioned branches into the training and test datasets. Specifically, we change the phase of only one of the branches to another available phase and generate 5 data samples for each fault type at each bus. Thus, both the new training and test datasets contain 30420 data samples while Gaussian noise of 45 dB is added.

Fault location accuracies of the models with additional phase-changed data are presented in Table IV. The results for faults at all buses and at modified buses with changed phases are included. For all schemes, the accuracies for faults at modified buses are lower than the counterparts of faults at all buses. The GCN model has the highest accuracies for all scenarios while the one-hop accuracies are more than 99%. Comparing the "All Buses" column of Table IV with the "Noise" column of Table III, we can see that the additional data has almost no impact on GCN, while the accuracies for other models decrease by 1-5%. Thus, we conclude that the GCN model is robust to the change of connected phases of single branches if the training dataset covers samples of the additional connection scenarios.

IV. Conclusion and Future Work

In this paper, we develop a GCN model for the task of fault location in distribution systems. Results tested with the IEEE 123 bus test system show that the proposed GCN model is very effective in processing fault-related data. The proposed model is more robust to measurement errors compared with many other machine learning models such as SVM, RF, and FCNN. Visualization of the activations of the last fully-connected layer shows that the GCN model extracts features that are robust to missing entries in the measurements. In a nutshell, the present paper proposes a flexible and widely-applicable energy data analytics framework for situational awareness in power distribution systems.

The proposed framework and approach open up a few interesting research directions. First, the effectiveness of the GCN model in more realistic settings needs further investigation; e.g., use field data to fine-tune the model trained with synthetic data). Second, it is of great interest to ensure that the GCN model can be adapted to topology changes such as network reconfiguration or expansion. Another concern is the integration of distributed generation, which introduces high-level uncertainties into the distribution grids, and may alter the characteristics of the measurements during faults.

Acknowledgement

The authors are grateful for the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this project. We would also like to thank Ron Levie at TU Berlin, and Federico Monti at University of Lugano, who helped us with the implementation of the GCN model.

References

[1] Y. Liao, "Generalized fault-location methods for overhead electric distribution systems," *IEEE Transactions on Power Delivery*, vol. 26, no. 1, pp. 53–64, Jan. 2011.

[2] R. Krishnathevar and E. E. Ngu, "Generalized impedance-based fault location for distribution systems," *IEEE Transactions on Power Delivery*, vol. 27, no. 1, pp. 449–451, Jan. 2012.

[3] S. Das, N. Karnik, and S. Santoso, "Distribution fault-locating algorithms using current only," *IEEE Transactions on Power Delivery*, vol. 27, no. 3, pp. 1144–1153, July 2012.

[4] R. A. F. Pereira, L. G. W. da Silva, M. Kezunovic, and J. R. S. Mantovani, "Improved fault location on distribution feeders based on matching during-fault voltage sags," *IEEE Transactions on Power Delivery*, vol. 24, no. 2, pp. 852–862, Apr. 2009.

[5] S. Lotfifard, M. Kezunovic, and M. J. Mousavi, "Voltage sag data utilization for distribution fault location," *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 1239–1246, Apr. 2011.

[6] F. C. Trindade, W. Freitas, and J. C. Vieira, "Fault location in distribution systems based on smart feeder meters," *IEEE Transactions on Power Delivery*, vol. 29, no. 1, pp. 251–260, Feb. 2014.

[7] J.-H. Teng, W.-H. Huang, and S.-W. Luan, "Automatic and fast faulted line-section location method for distribution systems based on fault indicators," *IEEE Transactions on Power systems*, vol. 29, no. 4, pp. 1653–1662, July 2014.

[8] Y. Jiang, C.-C. Liu, M. Diedesch, E. Lee, and A. K. Srivastava, "Outage management of distribution systems incorporating information from smart meters," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 4144–4154, Sept. 2016.

[9] D. W. Thomas, R. J. Carvalho, and E. T. Pereira, "Fault location in distribution systems based on traveling waves," in *2003 IEEE Bologna Power Tech Conference Proceedings*, vol. 2, 2003, pp. 242–246.

[10] S. Shi, A. Lei, X. He, S. Mirsaeidi, and X. Dong, "Travelling waves-based fault location scheme for feeders in power distribution network," *The Journal of Engineering*, vol. 2018, no. 15, pp. 1326–1329, Oct. 2018.

[11] D. Thukaram, H. Khincha, and H. Vijaynarasimha, "Artificial neural network and support vector machine approach for locating faults in radial distribution systems," *IEEE Transactions on Power Delivery*, vol. 20, no. 2, pp. 710–721, Apr. 2005.

[12] J. Mora-Florez, V. Barrera-Núñez, and G. Carrillo-Caicedo, "Fault location in power distribution systems using a learning algorithm for multivariable data analysis," *IEEE Transactions on Power Delivery*, vol. 22, no. 3, pp. 1715–1721, July 2007.

[13] Y. Aslan and Y. E. Yağan, "Artificial neural-network-based fault location for power distribution lines using the frequency spectra of fault data," *Electrical Engineering*, vol. 99, no. 1, pp. 301–311, Mar. 2017.

[14] Z. S. Hosseini, M. Mahoor, and A. Khodaei, "AMI-Enabled distribution network line outage identification via multi-label SVM," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5470–5472, Sept. 2018.

[15] M. Majidi and M. Etezadi-Amoli, "A new fault location technique in smart distribution networks using synchronized/nonsynchronized measurements," *IEEE Transactions on Power Delivery*, vol. 33, no. 3, pp. 1358–1368, June 2018.

[16] K. Chen, C. Huang, and J. He, "Fault detection, classification and location for transmission lines and distribution systems: a review on the methods," *High Voltage*, vol. 1, no. 1, pp. 25–33, Apr. 2016.

[17] M. Farajollahi, A. Shahsavari, E. Stewart, and H. Mohsenian-Rad, "Locating the source of events in power distribution systems using micro-PMU data," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6343–6354, Nov. 2018.

[18] S. Hossan and B. Chowdhury, "Data-driven fault location scheme for advanced distribution management systems," *IEEE Transactions on Smart Grid*, 2018.

[19] Y. Zhou, R. Arghandeh, and C. J. Spanos, "Partial knowledge data-driven event detection for power distribution networks," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5152–5162, Sept. 2018.

[20] K. Chen, Z. He, S. X. Wang, J. Hu, L. Li, and J. He, "Learning-based data analytics: Moving towards transparent power grids," *CSEE Journal of Power and Energy Systems*, vol. 4, no. 1, pp. 67–82, Mar. 2018.

[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[22] 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, no. 3, pp. 83–98, May 2013.

[23] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[24] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *International Conference on Learning Representations*, 2014.

[25] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, 2016, pp. 3844–3852.

[26] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional neural networks architectures for signals supported on graphs," *arXiv preprint arXiv:1805.00165*, 2018.

[27] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," *arXiv preprint arXiv:1705.07664*, 2017.

[28] 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, Mar. 2011.

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

[30] W. H. Kersting, "Radial distribution test feeders," in *Proc. IEEE Power Engineering Society Winter Meeting*, vol. 2, 2001, pp. 908–912.

[31] R. C. Dugan, "Reference guide: The open distribution system simulator (OpenDSS)," *Electric Power Research Institute, Inc*, 2016.

[32] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[33] M. Brown, M. Biswal, S. Brahma, S. J. Ranade, and H. Cao, "Characterizing and quantifying noise in PMU data," in *Proc. IEEE Power and Energy Society General Meeting*, 2016, pp. 1–5.

[34] G. Van der Maaten, Laurensand Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, Nov. 2008.