

UC Santa Cruz

UC Santa Cruz Previously Published Works

Title

Fault Location in Power Distribution Systems via Deep Graph Convolutional Networks

Permalink

<https://escholarship.org/uc/item/4t20p9zv>

Journal

IEEE Journal on Selected Areas in Communications, 38(1)

ISSN

0733-8716

Authors

Chen, Kunjin
Hu, Jun
Zhang, Yu
[et al.](#)

Publication Date

2020

DOI

10.1109/jsac.2019.2951964

Copyright Information

This work is made available under the terms of a Creative Commons Attribution-NonCommercial-ShareAlike License, available at <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Peer reviewed

Fault Location in Power Distribution Systems via Deep Graph Convolutional Networks

Kunjin Chen¹, Jun Hu¹, *Member, IEEE*, Yu Zhang, *Member, IEEE*,
Zhanqing Yu², *Member, IEEE*, and Jinliang He², *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 taking 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 data loss errors. Data visualization results of two competing neural networks are presented to explore the mechanism of GCNs superior performance. A data augmentation procedure is proposed to increase the robustness of the model under various levels of noise and data loss errors. Further experiments show that the model can adapt to topology changes of distribution networks and perform well with a limited number of measured buses.

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

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

Manuscript received May 27, 2019; revised September 16, 2019; accepted October 5, 2019. Date of publication November 6, 2019; date of current version January 31, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0904603, in part by the Natural Science Foundation of China under Grant 51720105004, in part by the State Grid Corporation of China under Grant 5202011600UJ, in part by the Hellman Fellowship, and in part by the Faculty Research Grant (FRG) of the University of California, Santa Cruz. (*Corresponding author: Jinliang He.*)

K. Chen, J. Hu, Z. Yu, and J. He are with the State Key Laboratory of Power Systems, Department of Electrical Engineering, Tsinghua University, Beijing 100084, China (e-mail: hejl@tsinghua.edu.cn).

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

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSAC.2019.2951964

0733-8716 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See <https://www.ieee.org/publications/rights/index.html> for more information.

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.

Machine learning models are leveraged 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] improves data-driven situational awareness for distribution systems [19], [20]. There are two major challenges for fault location in distribution systems with the increased number of measurements available: first, traditional fault location methods are unable to incorporate the measurements from different buses in a flexible manner, especially when the losses of data are taken into consideration. Second, for traditional machine learning approaches, the topology of the distribution network is hard to model, let alone the possibility of topology changes.

Recent advances in the field of machine learning, especially deep learning, have gained extensive attentions from both academia and industry. 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, such that the advantages of CNNs can be exploited for data residing on graphs [24]–[26].

In this paper, a GCN model is proposed for fault location in distribution systems. Unlike existing machine learning models

used for fault location tasks, the architecture of the proposed model preserves the spatial correlations of the buses and learns to integrate information from multiple measurement units. Features are extracted and composited in a layer-by-layer manner to facilitate the faulty bus classification task. We also design a data augmentation procedure to ensure that the model is robust to varied levels of noise and errors. In addition, the proposed model can be readily adapted or extended to various tasks concerning data processing for multiple measurements in modern smart grids.

The organization of the rest of the paper is as follows: in Section II, the fault location task is formulated and the proposed GCN model is described in detail. We also introduce the IEEE 123 bus test case used in this paper. The effectiveness of the proposed GCN model is validated in Section III with extensive comparisons and visualizations. A data augmentation procedure for training robust models is introduced. The performance of the model under topology changes and on high impedance faults is evaluated. We also implement the GCN model on another distribution network test case and discuss several practical concerns. Finally, Section IV concludes the paper and points out some future works.

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 .

As the convolution operation of CNN is carried out in local regions within the input data, local features can be extracted, and complex structures within the data can be represented with the increase of convolution layers (for a detailed description of CNN, the readers may refer to [27]). However, traditional CNN models can not be applied to signals on a distribution network as the inputs for CNN are supposed to be in Euclidean domains, such as images represented by values on regular two-dimensional grids and sequential data that is one-dimensional [28]. Thus, we introduce how a convolutional network can be constructed with signals on graphs hereinafter.

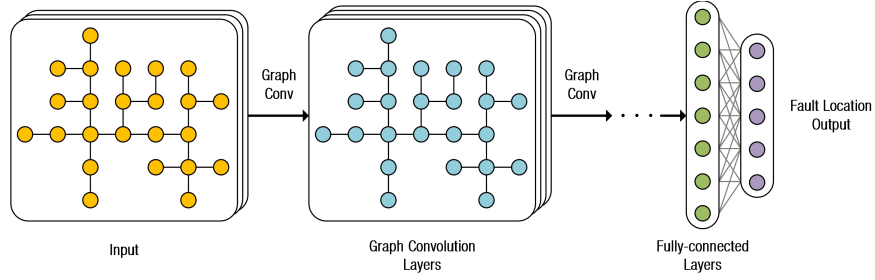


Fig. 1. The structure of the GCN model. Several graph convolution layers are followed by two fully-connected layers.

B. Spectral Convolution on Graphs

To be self-contained, we first present a brief introduction to spectral graph theory [29]. 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 $\Delta_u = \mathbf{D} - \mathbf{W}$, 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

$$\Delta = \mathbf{D}^{-1/2} \Delta_u \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}, \quad (1)$$

where \mathbf{I} is the identity matrix. The eigendecomposition of the positive semi-definite symmetric matrix Δ yields $\Delta = \Phi \Lambda \Phi^T$, where $\Phi = (\phi_1, \dots, \phi_n)$ are orthonormal eigenvectors of Δ , and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ is the diagonal matrix with corresponding ordered non-negative eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. Note that the smallest eigenvalue λ_1 equals zero with the eigenvector $\phi_1 = (\frac{1}{\sqrt{n}}, \dots, \frac{1}{\sqrt{n}})$. By analogy with the Fourier transform in Euclidean spaces, graph Fourier transform (GFT) can be defined for weighted graphs using the orthonormal eigenvectors of Δ [30]. For a signal $\mathbf{f} \in \mathbb{R}^n$ on the vertices of graph \mathcal{G} (each vertex has one value in this case), GFT is performed as $\hat{\mathbf{f}} = \Phi^T \mathbf{f}$ while the inverse GFT is $\mathbf{f} = \Phi \hat{\mathbf{f}}$. 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

$$\mathbf{g} * \mathbf{f} = \Phi ((\Phi^T \mathbf{g}) \circ (\Phi^T \mathbf{f})) = \Phi \text{diag}(\hat{g}_1, \dots, \hat{g}_n) \Phi^T \mathbf{f}, \quad (2)$$

where \circ indicates element-wise product between two vectors. Filtering of signal \mathbf{f} by spectral filter $\mathbf{B} = \text{diag}(\beta)$ with $\beta \in \mathbb{R}^n$ can then be expressed as $\Phi \mathbf{B} \Phi^T \mathbf{f}$. 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 $h_\alpha(\Lambda)$ that are smooth in spectral domain can bypass such an issue [29], [31]. For example, consider using a polynomial approximation

$$h_\alpha(\Lambda) = \sum_{k=0}^K \alpha_k \Lambda^k, \quad (3)$$

where $\alpha = (\alpha_1, \dots, \alpha_K)$ is the vector of coefficients to be learned for the filters and K is the degree of the polynomials. Further, in order to stabilize the training of the polynomial filters, the truncated Chebyshev polynomial expansion of $h_\alpha(\Lambda)$ is introduced [25], [30]. Specifically, expansion of $h_\alpha(\Lambda)$ using Chebyshev polynomials $T_k(\tilde{\Lambda})$ up to order K can be expressed as

$$h_\alpha(\Lambda) = \sum_{k=0}^K \alpha_k T_k(\tilde{\Lambda}), \quad (4)$$

where $\tilde{\Lambda} = 2\Lambda/\lambda_n - \mathbf{I}$. The recursive formulation of the filtering process based on Chebyshev 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 $\Delta^k = (\Phi \Lambda \Phi^T)^k = \Phi \Lambda^k \Phi^T$, the filtering process $\Phi h_\alpha(\Lambda) \Phi^T \mathbf{f}$ can be expressed as

$$\Phi h_\alpha(\Lambda) \Phi^T \mathbf{f} = h_\alpha(\Delta) \mathbf{f} = \sum_{k=0}^K \alpha_k T_k(\tilde{\Delta}) \mathbf{f}, \quad (5)$$

where $\tilde{\Delta} = 2\Delta/\lambda_n - \mathbf{I}$. Consequently, with $\mathbf{d}_0 = \mathbf{f}$ and $\mathbf{d}_1 = \tilde{\Delta} \mathbf{f}$, we can recursively calculate $\mathbf{d}_k = 2\tilde{\Delta} \mathbf{d}_{k-1} - \mathbf{d}_{k-2}$, and the filtering operation $h_\alpha(\Delta) \mathbf{f} = [\mathbf{d}_0, \dots, \mathbf{d}_K] \alpha$ has a computational complexity of $\mathcal{O}(K|\mathcal{E}|)$ considering the sparsity of Δ [25]. In addition, because the Chebyshev polynomials are truncated to the K th order, the filter is K -hop localized with respect to the connections embodied in Δ . 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 j th feature map of a graph convolution layer is calculated as

$$\mathbf{y}_j = \sum_{i=1}^{N_{in}} h_{\alpha_{i,j}}(\Delta) \mathbf{x}_i, \quad (6)$$

where $\mathbf{x}_i \in \mathbb{R}^n$ is the i th input feature map, $\alpha_{i,j} \in \mathbb{R}^K$ 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, $N_{in} = 12$ for the first layer of the model. The output of the last

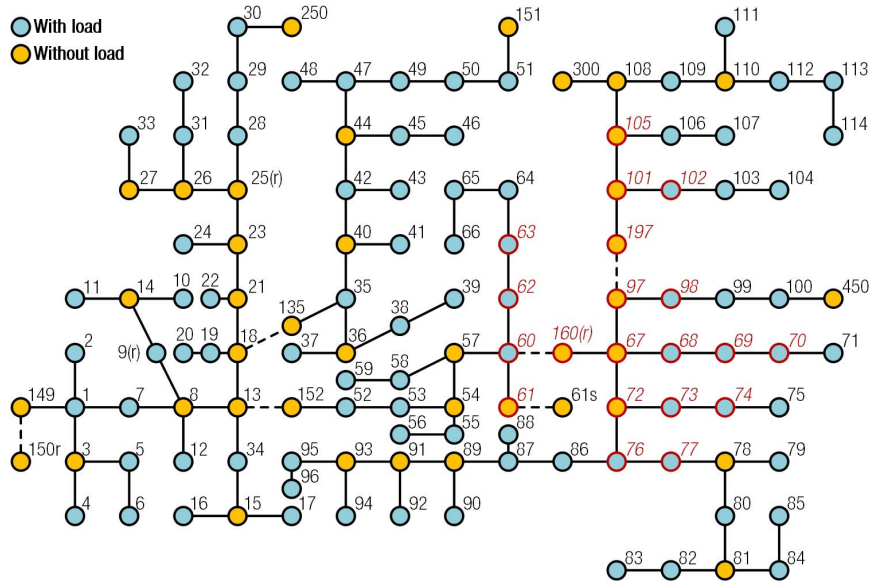


Fig. 2. An illustration of the IEEE 123 bus system. It is assumed that voltage and current phasors of PQ buses (connected to loads) are measured. As an example, the 20 buses that are closest (in distance) to bus 67 are highlighted with red color and italic numbers. Normally closed switches are represented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

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 i th 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 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 $\sigma_S = \sum_i \tilde{S}_{iK_n} / n$ (we have $\tilde{S}_{ij} \leq \tilde{S}_{ik}$ for $j < k$). Matrix $\tilde{\mathbf{W}} \in \mathbb{R}^{n \times K_n}$ is then constructed with $\tilde{W}_{ij} = e^{-\tilde{S}_{ij}^2 / \sigma_S^2}$. By restoring the positional correspondence of \tilde{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 Δ 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 [32]. 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. The faults have the resistance ranging from 0.05 Ω to 20 Ω . The load level of the system varies between 0.316 and 1. In order Fig. 3 shows

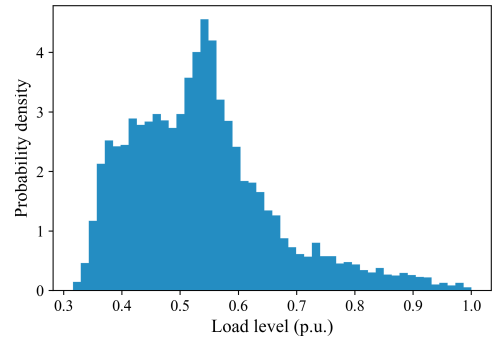


Fig. 3. The probability density of load level for the distribution system.

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 [33]. The voltage and current phasors are measured during the fault. We obtain the training and test datasets 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

¹The implementation of GCN in this paper is based on the implementation in [25]; see https://github.com/mdeff/cnn_graph

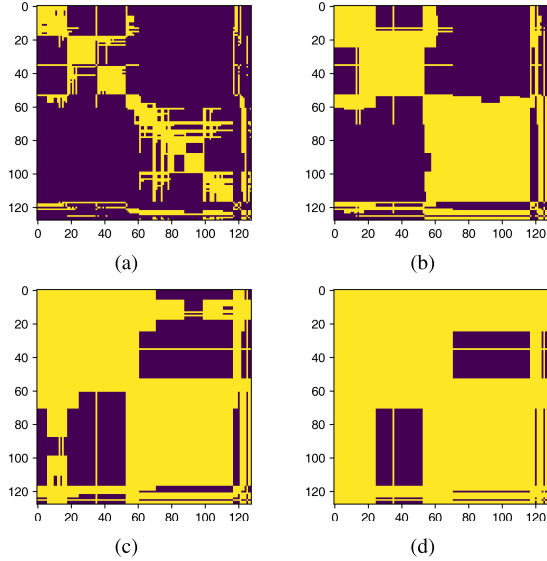


Fig. 4. Visualization of non-zero entries (yellow) in Δ^m when $K_n = 20$ (20 nearest buses of bus 67 are red colored as shown in Fig. 2): (a) $m = 1$, (b) $m = 2$, (c) $m = 3$, and (d) $m = 4$.

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 as the validation dataset. 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 (i.e., each data sample is used 400 times for training) and a mini-batch size of 32. We use Tensorflow in Python to implement the GCN model. When trained with a Titan Xp GPU, the GCN model takes less than 2 hours to train, and the time used to test each sample in the test dataset is less than 0.5 ms.

We first visualize Δ^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

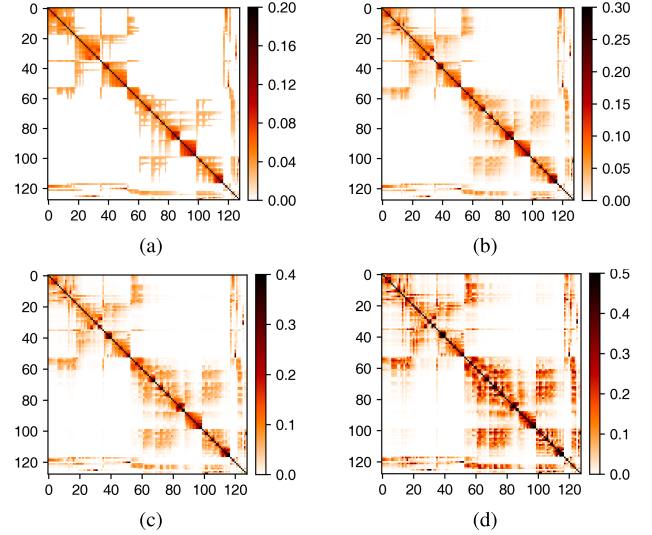


Fig. 5. Visualization of Δ^m when $K_n = 20$, (a) $m = 1$, (b) $m = 3$, (c) $m = 5$, and (d) $m = 10$. The indexes of the buses are sorted according to the order of bus numbers shown in Fig. 2. Absolute values of the entries in Δ^m are visualized. Entries with values greater than certain thresholds (i.e.,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

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 Δ^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 Δ^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 Δ , we conclude that the locality of filters are ensured when 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 [34] 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 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.

In order to justify the effectiveness of our proposed approach in real-world conditions including noise,

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

measurement errors and communication errors, 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 signal to noise rate (SNR) is 45 dB, as introduced in [35]. The noise has zero mean and the standard deviation, σ_{noise} , is calculated as $\sigma_{\text{noise}} = 10^{-\frac{\text{SNR}}{20}}$.
- 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 $\sigma_{\text{noise}} = 10^{-\frac{45}{20}}$, $N_{\text{drop}} = 1$ and $P_{\text{loss}} = 0.01$ throughout the experiments unless otherwise specified. 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. SVM 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

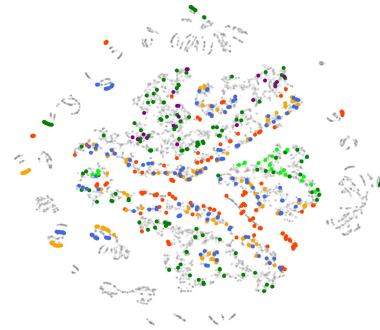


Fig. 6. Visualization of test data processed by PCA with 200 components and t-SNE with two components. Dots with the same color (except for small gray dots) correspond to the same faulty bus.

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 phase. This indicates that the robustness of the GCN model may be generalizable to other types of errors in the data. In all subsequent experiments in the paper, the samples in both training and test datasets are added with Gaussian noise of 45 dB unless otherwise stated.

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 [36]. 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

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

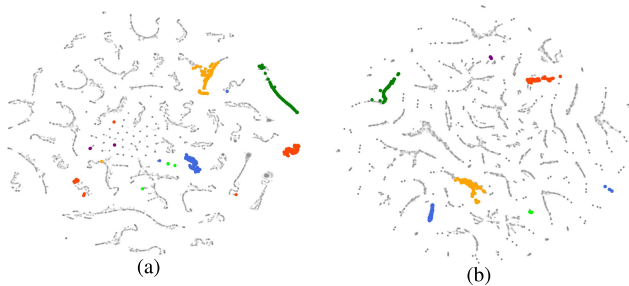


Fig. 7. Visualization of hidden features of test data added with Gaussian noise using t-SNE with two components: (a) the FCNN model, and (b) the GCN model. The models are trained with added Gaussian noise.

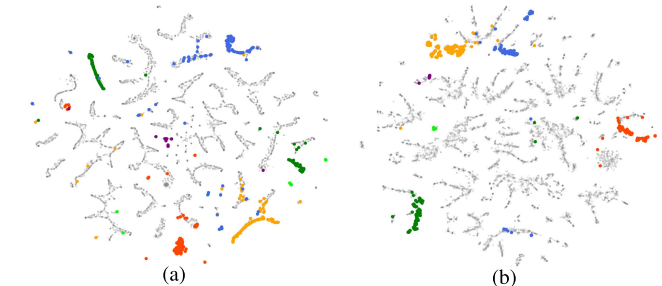


Fig. 8. Visualization of hidden features of test data added with all three types of modifications using t-SNE with two components: (a) the FCNN model, and (b) the GCN model. The models are trained with added Gaussian noise.

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%.

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 extent. 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. Increasing Model’s Robustness by Data Augmentation

We have shown that the GCN model is quite robust to mild noise and data loss errors (i.e., the SNR is 45 db, $N_{\text{drop}} = 1$, and $P_{\text{loss}} = 0.01$). The data collected from the field, however, may have lower SNR and higher data loss rates. Thus, it is desirable if the model is able to generalize to different levels of noise and data loss errors. In light of this, we implement data augmentation during training of the model by adding various levels of noise and data loss errors to the input data. Specifically, for the i th input sample in a mini-batch, we first add Gaussian noises with $\sigma_{\text{noise}} = \tilde{\sigma}_i$ to the measurements and randomly set measurements to 0 with $N_{\text{drop}} = \tilde{n}_i$ and $P_{\text{loss}} = \tilde{p}_i$. We randomly choose $\tilde{\sigma}_i$, \tilde{n}_i , and \tilde{p}_i from $[0, 10^{-\frac{45}{20}}, 10^{-\frac{40}{20}}, 10^{-\frac{35}{20}}, 10^{-\frac{30}{20}}, 10^{-\frac{25}{20}}]$, $[0, 1, 2, 3, 4, 5]$, and $[0, 0.01, 0.02, 0.03, 0.04, 0.05]$, respectively, with equal probability. Note that the data augmentation is applied to each mini-batch, thus a new data sample is generated for the j th data sample in each epoch unless $\tilde{\sigma}_j = \tilde{n}_j = \tilde{p}_j = 0$, in which case the data sample is unchanged.

We report the performance of the GCN model under various noise and data loss levels with and without data augmentation in Fig. 9. Specifically, case 1 has the lowest level of noise

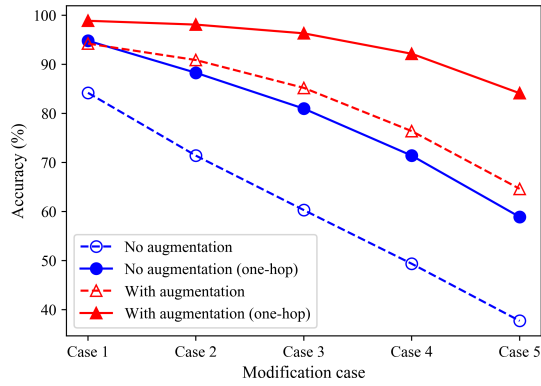


Fig. 9. Performance of the proposed GCN model with and without data augmentation. The values of σ_{noise} , N_{drop} , and P_{loss} are $(10^{-\frac{45}{20}}, 1, 0.01)$, $(10^{-\frac{40}{20}}, 2, 0.02)$, $(10^{-\frac{35}{20}}, 3, 0.03)$, $(10^{-\frac{30}{20}}, 4, 0.04)$, and $(10^{-\frac{25}{20}}, 5, 0.05)$ for the 5 cases.

and data loss errors while case 5 has the highest level. It is shown in the figure that the proposed data augmentation procedure greatly improves the fault location accuracies, and the one-hop accuracies for cases 1, 2, and 3 are higher than 95%. The one-hop accuracy when data augmentation is applied is higher than 84% even for case 5, for which the SNR is 25, measurements from 5 buses are lost, and each measurement may also be lost with a probability of 0.05.

Although the accuracy with data augmentation for case 1 is quite high, some samples are still assigned to wrong buses. In order to examine the characteristics of misclassified samples, we collect the samples with predicted faulty buses more than two hops away from the correct faulty buses. Note that random noise and data losses are added to the test samples, so the results are different for each trial. Specifically, the collected bus pairs are (25r, 250), (30, 25r), (33, 25r), (53, 56), (61s, 62), (61s, 68), (89, 79), (92, 95), (99, 105), (108, 102), (151, 49), (250, 28), and (250, 25r) (the first number is the correct bus). The majority of the two buses in the bus pairs are three hops away. In addition, 4 buses, namely, 29, 34, 76, and 108 are used to illustrate the characteristics of misclassified samples for case 5. The buses are located near the four corners of the network illustrated in Fig. 2. While bus 34 has no misclassified samples, the lists of wrong predictions more than two hops away for bus 29, 76, and 108 are [18 (5), 21 (4), 22 (5), 23 (3), 26 (3), 27 (4), 31 (4)], [60 (3), 61s (4), 66 (8), 74 (3), 75 (4), 79 (3), 81 (4), 83 (6), 97 (3), 99 (5)], and [61s (6), 97 (3), 98 (4), 100 (6), 111 (3), 450 (7)], respectively (the number of hops between the correct and predicted buses are included in the brackets). While hop numbers up to 8 exist in the results, most of the hop numbers are lower than 5. As the two-hop accuracy for case 5 is 92.49%, it can be concluded that the GCN model is able to locate a fault within the vicinity of its exact location under severe data loss errors in most cases.

In previous experiments, the values for the hyper-parameters K_n and K are 20 and [3, 4, 5] (the K for 3 graph convolution layers), respectively. It is expected that when data losses occur in some of the measurements close to a fault, information from other measurements may help the model locate the fault. In order to justify the choice of K_n and K , we report the

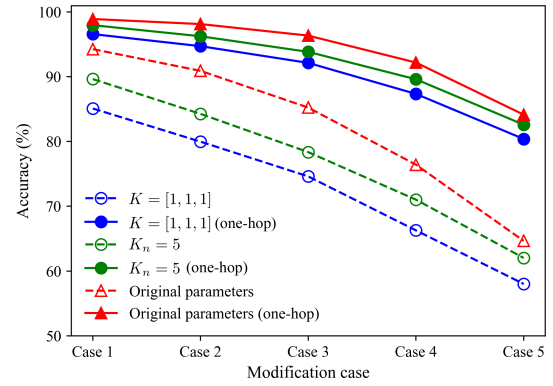


Fig. 10. Performance of the proposed GCN model with data augmentation and different hyper-parameter choices. The original hyper-parameters are $K = [3, 4, 5]$ and $K_n = 20$. Two experiments, namely, changing K to [1, 1, 1] and changing K_n to 5 are implemented for comparison.

performance of the GCN model with relatively small values of K_n and K in Fig. 10. Specifically, for one experiment we change K_n to 5, and for the other experiment we set K to [1, 1, 1]. It can be observed that the GCN model with the original hyper-parameters has higher accuracies. In addition, setting K to [1, 1, 1] hurts the performance of the model as it severely limits the range of measurements a node in the GCN can obtain information from.

E. Performance Under Distribution Network Reconfiguration

The configuration of a distribution network may change in order to reduce loss or balance the loads [37]. The reconfiguration of a network can be achieved by opening some of the normally closed switches and closing some of the normally open switches. The IEEE 123 bus system has a three-phase normally open switch between node 151 and node 300 (see Fig. 2), which can be used to guarantee electricity supply of the system if some of the normally closed switches are open.

In this work, we consider two cases of network reconfiguration:

- 1) Open the switch connecting node 18 and node 135, and close the switch connecting node 151 and node 300.
- 2) Open the switch connecting node 97 and node 197, and close the switch connecting node 151 and node 300.

In order to evaluate the performance of the proposed model under network reconfiguration, we generate 5 samples for each fault type at each bus for the two cases and directly use the GCN model trained with data augmentation to identify the faulty buses. As a result, the fault location accuracies and one-hop accuracies for the two cases are 88.37% (98.28%), and 91.89% (98.61%), respectively. As the reconfiguration scenarios are not considered during the training phase of the model, the results indicate that the GCN model has high stability against unseen network reconfiguration scenarios.

F. Performance Under Multiple Connection Scenarios of Branches

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

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

Specifically, the connected phase of a branch in a distribution network may change from time to time and it is expected that the model can deal with such changes. Three branches in the IEEE 123 bus system are considered:

- 1) The branch connecting bus 36, 38, and 39.
- 2) The branch connecting bus 67 to 71.
- 3) The branch connecting bus 108 to 114.

Phase 1 and 2 of bus 36, and all three phases of bus 67 and bus 108 are connected to the distribution system. The buses on the branches use only one of the phases for connection.

Unlike network reconfiguration achieved by opening and closing of switches, changing the connected phase of a single-phase branch requires additional data to train the models. 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.

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 column of ‘‘All Buses’’ 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.

G. Performance on High Impedance Faults

The detection of high impedance faults in distribution networks is a challenging task as the current magnitude is generally close to the level of load current [38]. In this subsection, we evaluate the performance of the GCN model on locating high impedance faults. Specifically, we add single-phase-to-ground faults with high fault resistance to the training dataset and report the fault location results on various ranges of fault resistance. For the construction of the training dataset, in addition to the data samples with small fault resistance values, we generate 40 samples for each phase at each bus and the fault resistance is uniformly sampled between 100 Ω

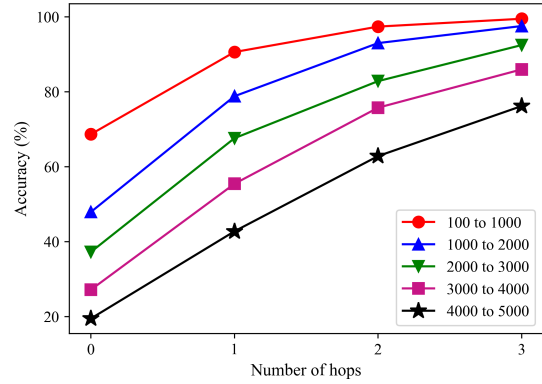


Fig. 11. Performance of the proposed GCN model for high impedance faults within various fault resistance ranges.

and 5000 Ω . Five fault resistance ranges, namely, 100 Ω to 1000 Ω , 1000 Ω to 2000 Ω , 2000 Ω to 3000 Ω , 3000 Ω to 4000 Ω , and 4000 Ω to 5000 Ω are used to generate test samples. For each range, 5 test samples are generated for each type of fault at each bus. In order to test the generalizability of the GCN model, we further split the fault resistance ranges into two sets of intervals (the length of each interval is 10 Ω), namely, $\{R_1 | 20k < R_1 < 20k + 10, k \in \mathbb{Z} : 5 \leq k \leq 249\}$, and $\{R_2 | 20k + 10 < R_2 < 20(k + 1), k \in \mathbb{Z} : 5 \leq k \leq 249\}$, where R_1 is used for samples in the training dataset and R_2 is used to generate samples in the test datasets.

The accuracies for high impedance faults with different ranges of fault resistance are shown in Fig. 11. As the zero-hop and one-hop accuracies are relatively low, we also report the two-hop and three-hop accuracies. It is seen in the figure that with the increase of fault resistance, the fault location accuracy drops. Although the increased fault resistance makes it hard to find the exact fault location, the accuracies increase rapidly with the increase of the number of hops, which indicates that the model is still able to capture a part of fault characteristics for high impedance faults.

H. Discussion on the Types of Measurements Used for the Model

As the proposed GCN model uses amplitudes and phase angles of both voltage and current measurements as inputs, it is necessary to examine the contributions of the different measurements to the performance of the model. Specifically, the first concern is that the measured currents are the injected currents at the loads, which provide less information about the faults compared with currents flowing in the branches connecting the buses. The second concern is the contribution of phase angle to the fault location accuracy as measuring the phase angle requires additional installation of phasor measurement devices.

In light of the concerns, we compare the performance of the GCN model under different measurement scenarios and present the results in Table V. The results for the last row are the same as the row of GCN in Table III. It is shown in the table that the results with voltage phasors are quite similar to the results with both voltage and current phasors. The results

TABLE V
FAULT LOCATION ACCURACIES OF THE GCN MODEL WITH DIFFERENT MEASUREMENT SCENARIOS

Measurement Scenario	Noise	Noise + Bus	Noise + Random	All Combined
Voltage amplitudes	94.66 / 99.36	67.51 / 82.91	75.55 / 87.38	58.02 / 75.34
Voltage phasors	97.43 / 99.88	90.38 / 97.49	89.82 / 97.33	83.54 / 94.63
Current phasors	91.04 / 98.43	82.40 / 93.64	83.04 / 94.44	76.32 / 90.44
Voltage and current phasors	97.52 / 99.73	92.67 / 98.26	88.76 / 96.44	84.53 / 94.77

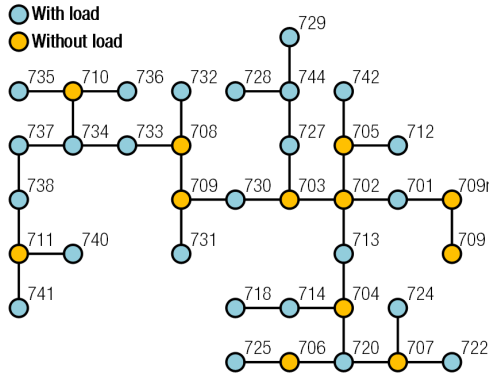


Fig. 12. An illustration of the IEEE 37 bus system.

with current phasors, however, is much lower than the results with voltage phasors. Further, when only voltage amplitudes are used, the accuracies with data loss errors are dramatically lower than other scenarios. Two conclusions can be drawn from the results:

- 1) For the design of the GCN model in this paper, the performance of the model mainly relies on the voltage phasors. Other types of current measurements such as currents flowing in the branches may be added in order to improve the fault location accuracy.
- 2) It is important to include phase angles in the inputs for the GCN model, especially when data loss errors are considered.

I. Implementation of the GCN Model on Another Distribution Network

As the previous experiments are all carried out in the IEEE 123 bus system, we implement the GCN model to the IEEE 37 bus system to verify that the model can perform well in a new distribution network. The topology of the IEEE 37 bus system is shown in Fig. 12. Similar to the implementation for the IEEE 123 bus system, we measure the voltage and current phasors at the phases connected to loads. The generation scheme for training and test datasets described in Section II. D is used. A series of hyper-parameters are used to see if the performance of the model is sensitive to the choice of hyper-parameters.

We first evaluate the performance of the GCN model with different values of K_n and K (the other hyper-parameters remain unchanged), and the accuracies are shown in Table VI. For each column in the table, the lowest two values are highlighted with underlines while the highest two values are marked in bold. It is observed that the performance of the

TABLE VI
FAULT LOCATION ACCURACIES OF THE GCN MODEL FOR THE IEEE 37 BUS SYSTEM WITH DIFFERENT VALUES OF K_n AND K

Hyper-parameters	Zero-hop	One-hop	Two-hop
$K_n = 10, K = [1, 2, 3]$	88.70	<u>95.31</u>	96.94
$K_n = 10, K = [2, 3, 4]$	88.94	96.36	97.03
$K_n = 10, K = [3, 4, 5]$	89.97	97.14	97.89
$K_n = 5, K = [2, 3, 4]$	88.90	<u>95.04</u>	96.81
$K_n = 15, K = [2, 3, 4]$	90.05	96.93	97.73
$K_n = 20, K = [2, 3, 4]$	89.51	96.20	97.25
$K_n = 5, K = [1, 1, 1]$	<u>88.12</u>	95.59	<u>96.48</u>
$K_n = 10, K = [1, 1, 1]$	<u>88.18</u>	<u>95.31</u>	<u>96.25</u>

TABLE VII
FAULT LOCATION ACCURACIES OF THE GCN MODEL FOR THE IEEE 37 BUS SYSTEM WITH DIFFERENT NUMBERS OF LAYERS

Hyper-parameters	Zero-hop	One-hop	Two-hop
$K = [2]$ (1 layer)	74.80	88.55	93.06
$K = [2, 3]$ (2 layers)	85.26	94.37	95.59
$K = [2, 3, 4]$ (3 layers)	88.94	96.36	97.03

GCN model is quite stable under different values of K_n and K . In Section III. D, we have shown that properly choosing the values of K_n and K can increase the model's robustness against data loss errors. When data loss errors are not considered, however, the negative effect of setting K_n and K to small values is insignificant. This indicates that laborious tuning of hyper-parameters is not needed when implementing the GCN model to a new distribution network.

Another important hyper-parameter is the number of graph convolution layers. Although the GCN model does not require a large number of layers, it is expected that the model may not have enough learning capacity when the number of layers is not enough. We report the performance of the GCN model with different numbers of graph convolution layers in Table VII. It is clearly observed in the table that increasing the number of layers improves the fault location accuracy, but the gain of adding another layer decreases when the third layer is added.

Finally, we use the IEEE 37 bus system to discuss some practicability issues. The first concern is the number of measured buses required for the GCN model. In the original scenario, the phases connected to loads at 25 buses are monitored. We gradually reduce the number of monitored buses and compare the performance of the model under these reduction cases. Specifically, the lists of removed buses are [714, 733, 737, 738, 744], [720, 727, 734], [724, 728, 735], [701, 712, 713, 730, 741], and [718, 722, 729, 731, 736].

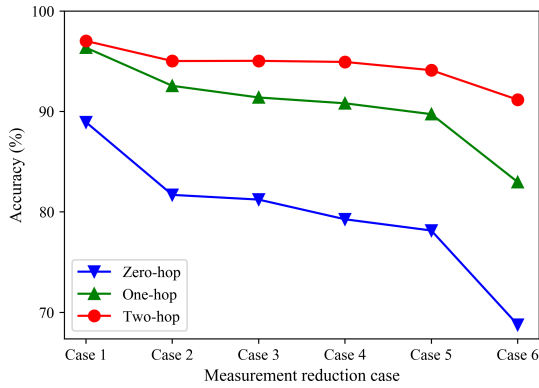


Fig. 13. Performance of the proposed GCN model with different cases of reduction on number of measured buses. Case 1 represents the original scenario with 25 measured buses.

TABLE VIII

FAULT LOCATION ACCURACIES OF THE GCN MODEL FOR THE IEEE 37 BUS SYSTEM WITH DIFFERENT SIZES OF TRAINING DATASETS

Dataset Size	Zero-hop	One-hop	Two-hop
100%	88.94	96.36	97.03
50%	66.99	87.04	93.73
25%	45.46	73.33	88.69
10%	21.42	46.79	67.49
5%	13.21	33.27	52.73

Although this is only one possibility of reduction, we try to remove the buses evenly across the network to avoid large areas of unmonitored buses. When the 4th list of buses are removed, only 9 buses at the end of branches are left. The final reduction case leaves the model with only 4 measured buses, namely, 725, 732, 740, and 742. The accuracies of the different reduction cases are illustrated in Fig. 13. Apparently, reducing the number of measured buses has a negative effect on the fault location accuracies. The two-hop accuracy, however, is not very sensitive to the reduction of measured buses until the last reduction. Four case 5, specifically, the two-hop accuracy is 94.12% with measurements from only 9 buses at branch ends. The results indicate that it is harder for the GCN model to find the exact fault locations when a large proportion of the measured buses are excluded, but the ability to find the vicinity of the faulty bus only requires a small proportion of buses to be measured.

The second concern is the number of training data needed to train the GCN model. The above mentioned dataset for the IEEE 37 bus system contains 20 samples for each fault type at each bus. With this as the size of 100%, we reduce the number of generated samples for each fault type at each bus to 10, 5, 2, and 1, and compare the accuracies of the scenarios in Table VIII. The size of the test dataset remain the same. It is seen in the table that the performance of the GCN model degrades as the size of the training dataset reduces. With only 1 sample for each fault type at each bus, the two-hop accuracy is a little above 50%. As it is hard to collect field data with varied fault types, fault resistances and load levels, a more practical solution is to combine field data with

synthetic data simulated according to the need of the model. With the help of transfer learning [39], the model can transfer the knowledge learned from simulated data to locate faults using actual measurements from the system. Such a problem formulation is beyond the scope of this paper, but the results in our work provides an upper bound for the performance of the GCN model as we use simulated data only.

IV. CONCLUSION AND FUTURE WORK

In this paper, we develop a GCN model for the task of fault location in distribution systems. Simulation results tested with the IEEE 123-bus and 37-bus systems show that the proposed GCN model is significantly effective in processing fault-related data. The proposed model is more robust to measurement errors compared with many other machine learning approaches including 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. Further experiments show that the model can adapt to topology changes and perform well with a limited number of measured buses. In a nutshell, the present paper proposes a flexible and widely-applicable energy data analytics framework for improving 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, or train the model with both field data and synthetic data by transfer learning). Second, it is valuable to develop new schemes for transferring a learned model to other distribution systems with different topologies. A new challenge comes from the integration of distributed generation, which introduces high-level uncertainties into the grids, and may alter the characteristics of the measurements during faults.

ACKNOWLEDGMENT

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 R. Levie at TU Berlin, and F. 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 Trans. Power Del.*, vol. 26, no. 1, pp. 53–64, Jan. 2011.
- [2] R. Krishnathar and E. E. Ngu, "Generalized impedance-based fault location for distribution systems," *IEEE Trans. Power Del.*, 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 Trans. Power Del.*, vol. 27, no. 3, pp. 1144–1153, Jul. 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 Trans. Power Del.*, 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 Trans. Power Del.*, vol. 26, no. 2, pp. 1239–1246, Apr. 2011.

- [6] F. C. L. Trindade, W. Freitas, and J. C. M. Vieira, "Fault location in distribution systems based on smart feeder meters," *IEEE Trans. Power Del.*, 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 Trans. Power Syst.*, vol. 29, no. 4, pp. 1653–1662, Jul. 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 Trans. Power Syst.*, vol. 31, no. 5, pp. 4144–4154, Sep. 2016.
- [9] D. W. P. Thomas, R. J. O. Carvalho, and E. T. Pereira, "Fault location in distribution systems based on traveling waves," in *Proc. IEEE Bologna Power Tech Conf. Proc.*, vol. 2, Jun. 2003, pp. 1–5.
- [10] S. Shi, A. Lei, X. He, S. Mirsaedi, and X. Dong, "Travelling waves-based fault location scheme for feeders in power distribution network," *J. Eng.*, vol. 2018, no. 15, pp. 1326–1329, Oct. 2018.
- [11] D. Thukaram, H. P. Khincha, and H. P. Vijaynarasimha, "Artificial neural network and support vector Machine approach for locating faults in radial distribution systems," *IEEE Trans. Power Del.*, vol. 20, no. 2, pp. 710–721, Apr. 2005.
- [12] J. Mora-Florez, V. Barrera-Nunez, and G. Carrillo-Caicedo, "Fault location in power distribution systems using a learning algorithm for multivariable data analysis," *IEEE Trans. Power Del.*, vol. 22, no. 3, pp. 1715–1721, Jul. 2007.
- [13] Y. Aslan and Y. E. Yağın, "Artificial neural-network-based fault location for power distribution lines using the frequency spectra of fault data," *Elect. Eng.*, 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 Trans. Smart Grid*, vol. 9, no. 5, pp. 5470–5472, Sep. 2018.
- [15] M. Majidi and M. Etezadi-Amoli, "A new fault location technique in smart distribution networks using synchronized/nonsynchronized measurements," *IEEE Trans. Power Del.*, vol. 33, no. 3, pp. 1358–1368, Jun. 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. M. Stewart, and H. Mohsenian-Rad, "Locating the source of events in power distribution systems using micro-PMU data," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6343–6354, Nov. 2018.
- [18] M. S. Hossain and B. Chowdhury, "Data-driven fault location scheme for advanced distribution management systems," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5386–5396, Sep. 2019.
- [19] Y. Zhou, R. Arghandeh, and C. J. Spanos, "Partial knowledge data-driven event detection for power distribution networks," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5152–5162, Sep. 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 J. Power Energy Syst.*, 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
- [23] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, 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 *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.
- [25] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
- [26] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional neural network architectures for signals supported on graphs," May 2018, *arXiv:1805.00165*. [Online]. Available: <https://arxiv.org/abs/1805.00165>
- [27] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.
- [28] K. Chen, J. Hu, and J. He, "Detection and classification of transmission line faults based on unsupervised feature learning and convolutional sparse autoencoder," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1748–1758, May 2018.
- [29] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "CayleyNets: Graph convolutional neural networks with complex rational spectral filters," May 2017, *arXiv:1705.07664*. [Online]. Available: <https://arxiv.org/abs/1705.07664>
- [30] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.
- [31] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," Jun. 2015, *arXiv:1506.05163*. [Online]. Available: <https://arxiv.org/abs/1506.05163>
- [32] W. H. Kersting, "Radial distribution test feeders," in *Proc. IEEE Power Eng. Soc. Winter Meeting Conf.*, vol. 2, Jan./Feb. 2001, pp. 908–912.
- [33] R. C. Dugan and T. E. McDermott, "An open source platform for collaborating on smart grid research," in *Proc. IEEE Power Energy Soc. General Meeting*, Jul. 2011, pp. 1–7.
- [34] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [35] M. Brown, M. Biswal, S. Brahma, S. J. Ranade, and H. Cao, "Characterizing and quantifying noise in PMU data," in *Proc. IEEE Power Energy Soc. General Meeting*, Jul. 2016, pp. 1–5.
- [36] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
- [37] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Trans. Power Del.*, vol. 4, no. 2, pp. 1401–1407, Apr. 1989.
- [38] A. Ghaderi, H. L. Ginn, and H. A. Mohammadpour, "High impedance fault detection: A review," *Electr. Power Syst. Res.*, vol. 143, pp. 376–388, Feb. 2017.
- [39] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.



Kunjin Chen received the B.Sc. degree in electrical engineering from Tsinghua University, Beijing, China, in 2015, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering.

His research interests include applications of machine learning and data science in power systems.



Jun Hu (M'10) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Department of Electrical Engineering, Tsinghua University, Beijing, China, in July 1998, July 2000, and July 2008, respectively.

He is currently an Associate Professor with the Department of Electrical Engineering, Tsinghua University. His research interests include overvoltage analysis in power systems, sensors and big data, dielectric materials, and surge arrester technology.



Yu Zhang (M'15) received the Ph.D. degree in electrical and computer engineering from the University of Minnesota, Minneapolis, MN, USA, in 2015.

He is currently an Assistant Professor with the ECE Department, UC Santa Cruz. Prior to joining UCSC, he held a post-doctoral position at UC Berkeley and the Lawrence Berkeley National Laboratory. His research interests include broad areas of cyber-physical systems, smart power grids, optimization theory, machine learning, and big data analytics.



Zhanqing Yu (M'07) received the B.Sc. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2003 and 2008, respectively.

He was a Lecturer with the Department of Electrical Engineering, Tsinghua University, from 2010 to 2012. He became a Post-Doctoral Researcher with the Department of Electrical Engineering, Tsinghua University, in 2008. Since 2013, he has been an Associate Professor. His research interests include electromagnetic compatibility and overvoltage analysis of HVDC and HVAC systems.



Jinliang He (M'02–SM'02–F'08) received the B.Sc. degree from the Wuhan University of Hydraulic and Electrical Engineering, Wuhan, China, in 1988, the M.Sc. degree from Chongqing University, Chongqing, China, in 1991, and the Ph.D. degree from Tsinghua University, Beijing, China, in 1994, all in electrical engineering.

He became a Lecturer and an Associate Professor with the Department of Electrical Engineering, Tsinghua University, in 1994 and 1996, respectively.

From 1997 to 1998, he was a Visiting Scientist with the Korea Electrotechnology Research Institute, Changwon, South Korea, where he was involved in the research on metal oxide varistors and high-voltage polymeric metal oxide surge arresters. From 2014 to 2015, he was a Visiting Professor with the Department of Electrical Engineering, Stanford University, Palo Alto, CA, USA. In 2001, he was promoted to a Professor with Tsinghua University, where he is currently the Chair with the High Voltage Research Institute. He has authored seven books and 400 technical articles. His research interests include overvoltages and EMC in power systems and electronic systems, lightning protection, grounding technology, power apparatus, smart sensors, and dielectric material.