Deep Convolutional Neural Networks for Distribution System Fault Classification

Guanyu Tian
School of Electrical and
Computer Engineering
University of Central Florida
Orlando, Florida 32816
Email: tiang@knights.ucf.edu

Qun Zhou
School of Electrical and
Computer Engineering
University of Central Florida
Orlando, Florida 32816
Email: Qun.Zhou@ucf.edu

Liang Du
School of Electrical and
Computer Engineering
Temple University
Philadelphia, Pennsylvania 19122
Email: ldu@temple.edu

***Abstract*—Faults happen very frequently in distribution systems. Identifying fault types and phases are of critical importance for outage management, fault location, and service restoration. However, this task becomes very challenging due to measurement scarcity in distribution systems. This paper is among the first few that applies deep learning techniques in distribution system fault classification. Specifically, a sequential Convolutional Neural Network(CNN)-based classifier is developed to identify fault buses and phases. The input to the CNN is the steady-state voltage and current data measured at substations. The fault identification is modeled as a multi-label classification problem. Training data under various fault scenarios are obtained in OpenDSS and Gaussian noises are added to mimic measurement errors. A case study in IEEE 13-feeder test system is conducted with single and multiple bus faults scenarios. Numerical results demonstrate the high accuracy and fast computation of the proposed deep CNN-based fault classification.**

I. Introduction

Distribution system state estimation has become possible with advanced sensing and communication technology. The pre-requisite of state estimation is the topology identification because distribution system topology constantly changes over time due to faults or switching actions [1]. Topology identification is a sub-problem and also a fundamental problem of state estimation [2]. However, due to the lack of sufficient measurement or communication devices installed in distribution systems, fault identification becomes very challenging [3]. Identifying fault types and phases are of critical importance for outage management, fault location, and service restoration.

Many techniques have been introduced to classify and identify faults. A basic fault location identification is based on fault impedance, inferring the distance by calculating the fault impedance from fault voltage and fault current [4]. With the development of distribution system measurement devices, data based methods are introduced [5]. An outage escalation procedure and meter-polling procedure for searching outage region are introduced in [6] . A synergistic method to use automatic meter reading(AMR) to identify the outage map is introduced in [7] . Jiang et al. introduced a method to identify the outage section by calculating the credibility of each hypothesis and select the hypothesis of the highest credibility as the identified outage scenario [8]. The credibility is a ratio of supporting measurements to non-supporting measurements from smart meters and fault indicators.

Using substation-only measurement data, this paper is among the first few that applies deep learning techniques in distribution system fault classification. Specifically, a sequential Convolutional Neural Network(CNN)-based classifier is developed to identify fault buses and phases. The input to the CNN is the steady-state voltage and current data measured at substations. The fault identification is modeled as a multi-label classification problem. Training data with various fault scenarios are obtained in OpenDSS and Gaussian noises are added to mimic measurement errors. Numerical results demonstrate the high accuracy and fast computation of the proposed deep CNN-based fault classification.

The paper is organized as follows. Section II introduces the architecture and key components of the CNN fault classifier, including the feature extraction layers and the multi-label classification layer. Section III discusses the case study and results using an IEEE 13-bus test feeder system in an OpenDSS environment Section IV summarizes the paper and discusses future work.

II. Problem Formulation

The fault classification problem is tackled using deep convolutional neural networks. CNNs are inspired by humanvisual cognition mechanism, originally created for image processing. After decades of development, it has been widely used for feature extraction and classification in many areas, including natural language processing(NLP), video classification, etc. [9]–[12].

This paper applies CNNs to solve a challenging power system problem - fault classification. The principle is to process measurements in distribution power system as 2D or 3D images and capture the spatial-temporal correlations among measurement data. In addition, a multi-label classification layer is added in the output layer to predict the status of nodes. The architecture of the classifier is shown in Fig. 1. First, the measurement data is reshaped from a 3D-tensor to 2D-matrix. Then two convolutional layers are applied to extract multiple features from the data. Finally, a fully-connected (FC) multi-layer perceptrons are added for classification and an indicator vector is an output for fault identification.

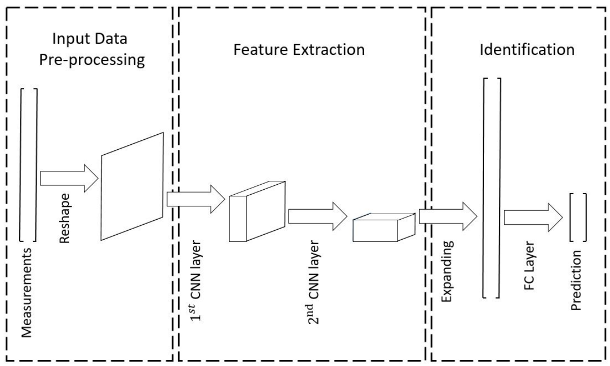

Fig. 1. CNN-based Fault Classifier

A. Measurement Data and Data Generation

We assume measurements are only available at the substation [13].The measurement data are steady-sate pre-fault and post-fault voltage and current phasors, as well as voltage and current phasors after the fault is cleared. Specifically, the data measured are three-phase voltage magnitudes, voltage angles, current magnitudes and current angles. Therefore, there are 12 measurement items in total. Each item is a vector containing time series values during the fault observing window. The fault observing window is a time period that is long enough to observe fault from happening to clearing. Each input of CNN model is a matrix [14]. The format of the matrix is shown in Fig. 2.

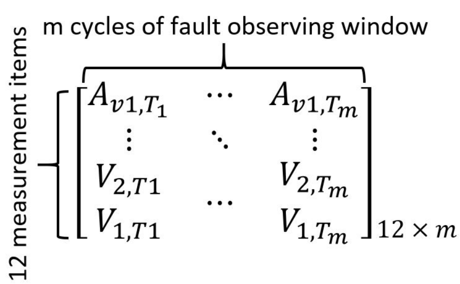

Fig. 2. Data Structure

The data set is generated from OpenDSS fault dynamic simulation interfaced with python [15]. Loading conditions are taken into consideration with an absolute volatility(α) and relative volatility(β). The absolute load volatility is the ratio to the base load value and relative volatility is the variance of each load. So the actual value of i^{th} load is formulated as

$$L_{i,actual} = \alpha \times L_{i,base} \times (1 + \beta_i) \tag{1}$$

where $\alpha \in (0, 1]$ and $\beta_i \sim N(0, 0.1)$.

The flowchart of data generation is shown in Fig. 3. We first start with the base load profile and run power flow in OpenDSS. Then generate fault scenarios that cover various fault buses, fault phases, and fault types. The loading condition will also be varied to reflect real-world load fluctuations. We then apply a specific fault, solve power flow, and record steady-state voltages and currents at the substation. We also add noises to the recorded data to mimic measurement errors. We continue this procedure until a specified number of training data is satisfied.

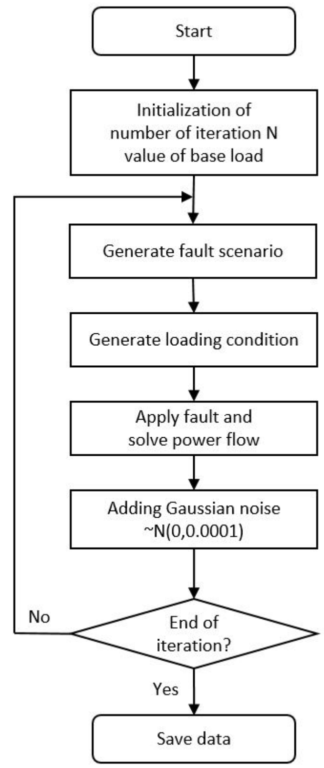

Fig. 3. Data generation flowchart

B. Sequential CNN Model

CNN is the extension of the neural network in 3-dimension [16], in the sense that the inputs of CNN are usually 3D tensors. This feature enables the model to explore the correlation among inputs in 3 directions(width, length, depth). Like human eyes observing a picture, the width and length construct the shape, and the depth represents the color. In deep learning applications, CNN models are used in a stacked way to enhance the ability of feature extraction. Therefore, the feature extraction of fault classification can be efficiently carried out using a sequential CNN model. A typical architecture of sequential is shown in Fig. 4. The data set is first separated into mini-batches, then a set of feature map is extracted and eventually classified into a class.

Fig. 5 shows how the filter moves along the length or width and the corresponding output position in the feature map. The size of convolution layer output depends on the setting of filter, padding and stride. Their relation is formulated as

$$N_{out} = (N_{in} - F + 2P)/S + 1 \tag{2}$$

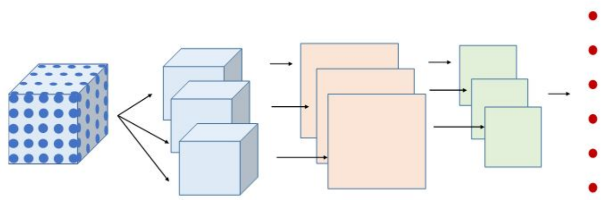

Fig. 4. Typical architecture of CNN

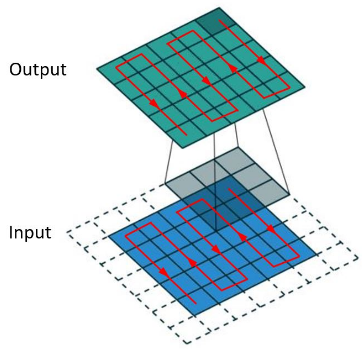

Fig. 5. Example of convolution [17]

in which, N is the size of either length or width, F, P and S are filter size, padding size and stride size respectively of the dimension. So, for the example shown in Fig. 5, input size is (5,5), filter size is (3,3), padding size is (1,1), stride is (1,1) and the output size is (5,5).

The parameters of the identifier model are specified in Table I. Filter size determines the size of weights and bias matrix. Padding is to preserve the spatial size of the tensor, compensating for the size lost during convolution [16]. Pooling layer is the downsampling layer in CNN and the pooling size determines the downsampling rate along the spatial dimension [16]. Adam optimizer is a stochastic optimization method that has fast convergence speed and great robustness [18]. Initialization of weight is very important in deep learning because weight determines whether the signal is decayed or amplified, and Xavier initialization method makes sure that the initial weight of each layer is in a reasonable range, enhancing the robustness of the model [19].

TABLE I
CNN MODEL SETTINGS

Item	Value
Filter size	(5,5)
Padding	(2,2)
Pooling size	(2,2)
Optimizer	Adam
Initialization	Xavier

C. Multi-label Classifier

Multi-layer perceptron(MLP) is a simple feedforward architecture of artificial neural networks. A typical architecture of MLP is shown in Fig. 6. The MLP consists of an input layer, a hidden layer, and an output layer [20]. Each neuron has an activation function used for simulating the non-linearity of the target function.

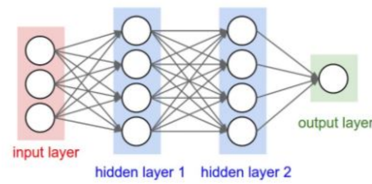

Fig. 6. Typical architecture of MLP [16]

The relation between the input and output of a layer is formulated as

$$out = a(w \times x_{in} + b) \tag{3}$$

where $a(\cdot)$ is the activation function, w is the weights matrix and b is the bias vector.

MLP itself is capable of simple classification problems, and it is also an indispensable part of deep learning models for complicated problems. The last layer of the outage identifier is a fully connected MLP that outputs classification labels.

Traditional fault section identification methods rely on generating hypothesis set and choose the scenario with highest credibility [8]. The size of hypothesis increases exponentially with the number of lines. To avoid this computation inefficiency, a multi-label classification criterion is used in the training. Unlike single labeling for each fault scenario, the multi-label classification model is suitable for identifying multiple faults in distribution systems. Each phase of a node is an element in the label vector, and each label has two classes [0,1], whereas 0 represents no fault and 1 represents fault. In this way, any fault scenarios can be easily represented by the label vector. The illustration of label vector is shown in Fig. 7.

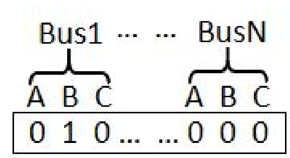

Fig. 7. Format of label vector

The loss function used for the training is the soft margin cross-entropy [21], formulated as

$$loss(p, y) = -\frac{1}{N}\sum_{i}^{N} y_i log(p_i) + (1 - y_i) log(1 - p_i) \tag{4}$$

p and y in the loss function are the prediction probability and the target label. p is a vector of probability corresponding to the element in y. By minimizing the cross-entropy loss, the expected value of misclassification is minimized. As the prediction is a probability vector, and the labels are binary incidence, so a threshold needs to be chosen for classification. The common value of the threshold is 0.5 that indicates the corresponding fault is more likely to happen.

III. Case Study

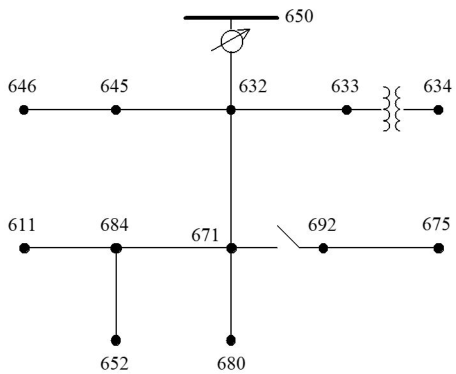

Fig. 8. Topology of IEEE-13bus feeder [22]

The test system is the IEEE-13bus feeder shown in Fig. 8. Experiments for single bus fault and multiple bus fault scenarios are conducted. We also examine the robustness of CNN-based classification assuming communication errors with missing data. The measurement device is installed at the node 650. The training data set has 5000 samples, the time consumption of training is 415.2 seconds, on a GPU. The specifications of training setting are shown in Table II.

TABLE II
Summary of training settings

Item	Value
batch size	10
epoch	20
learning rate	0.001

A. Training Data Generation

The generation of training data follows the flowchart in Fig. 3. An example illustrating the steps of generating the fault and loading condition of a data sample is shown in Table III. The sample fault is a single phase A to ground fault at bus 675 in IEEE-13bus test feeder. First, we create a label vector for applying the specified fault and then generate loading condition by multiplying based load with $\alpha(1+\beta)$. Having the fault and loading condition generated, measurements are calculated by OpenDSS power flow dynamic simulation. The plot of three-phase voltage and current magnitudes are shown in Fig. 9(a)(c). Fig. 9(b)(d) shows the curves of data added with Gaussian noise. The visualized comparison of comprehensive measurement data to the heat map matrix is shown in Fig. 10. Fig. 10(a) shows the comprehensive 3D plot of all measurement data, 3 axes are variable, time and measured value; Fig. 10(b) shows the image input of identifier, though there is only 2 axis(variable and time), the hidden axis(value) is represented by color. The use of heat map enables the model to explore the relation between both variables and time axis, therefore more features can be extracted, making the result more accurate.

TABLE III
Data generation example

Item	Value
Step 1: Generate label	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0]
Step 2: Read base load	[1155.0, 160.0, 120.0, 120.0, 170.0, 230.0, 170.0, 485.0, 68.0, 290.0, 170.0, 128.0, 17.0, 66.0, 117.0]
Step 3: Generate α	0.8
Step 4: Generate β	[0.079 0.013 0.109 0.121 -0.098 -0.045 0.106 0.113 0.155 0.131 0.171 -0.108 0.020 -0.119 -0.016]
Step 5: Calculate actual loads	[997.3 129.7 106.5 107.6 122.7 175.8 150.4 431.7 62.8 262.4 159.2 91.4 13.9 46.5 92.1]

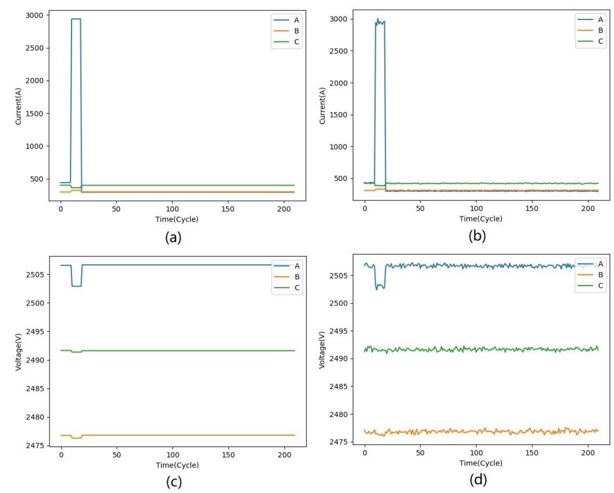

Fig. 9. Three phase curves of measured voltage and current

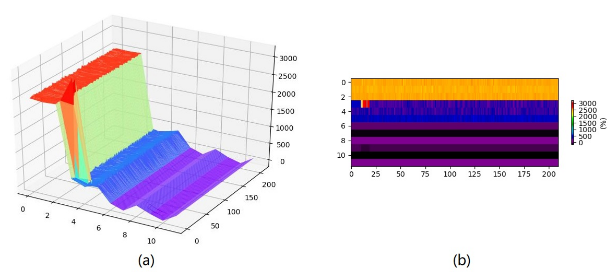

Fig. 10. 3D visualization and heat map of measurement data

B. Single Bus Fault

The test data set of single bus fault validation has 3000 samples in total, including 1000 single-phase fault, 1000 two-phase fault and 1000 three phase fault. The identification accuracy is 99% for the whole test data set and the accuracy for single-phase, two-phase and three-phase fault are 100%, 97%, 100% respectively. The time consumption of identifying 3000 samples is 13.873 seconds. In average, it takes less than 5 ms to complete one identification. 20 examples from the test data set are listed in Table IV as well the corresponding identification results.

TABLE IV
EXAMPLES OF SINGLE BUS FAULT TEST DATA

Index	Bus.Phase(s)	α	Identified fault
1	633.A	0.9	633.A
2	645.B	0.7	645.B
3	646.A	0.5	646.A
4	611.C	0.6	611.C
5	684.B	0.8	684.B
6	652.A	0.9	652.A
7	671.A	0.7	671.A
8	680.C	0.9	680.C
9	675.C	0.5	675.C
10	633.AB	0.9	633.AB
11	675.BC	0.7	675.BC
12	671.AC	0.6	671.AC
13	680.AB	0.7	680.AB
14	646.BC	0.5	646.BC
15	684.AC	0.8	611.C,652.A
16	645.BC	0.7	645.BC
17	675.ABC	0.9	675.ABC
18	671.ABC	0.8	671.ABC
19	680.ABC	0.6	680.ABC
20	633.ABC	0.5	633.ABC

C. Multiple Bus Fault

As mentioned earlier, one of the improvements of sequential CNN based identifier is that it can be easily generalized to multiple fault identification. The test data set for multiple bus fault validation has 1000 samples, including 500 two-fault samples and 500 three-fault samples. The overall identification accuracy is 93%, time consumption of predicting 1000 samples is 7.113 seconds. 15 examples from the test data set are listed in Table V as well the corresponding identification results.

TABLE V
EXAMPLES OF MULTIPLE BUS FAULT TEST DATA

Index	Fault	α	Identified fault
1	633.A,671.A	0.9	633.A,671.A
2	645.B,675.B	0.7	645.B,675.B
3	646.C,611.C	0.5	646.C,611.C
4	684.AC,633.AB	0.6	684.AC,633.AB
5	671.AB,675.C	0.8	671.AB,675.ABC
6	652.A,645.BC	0.9	652.A,645.BC
7	671.A,680.BC	0.7	680.ABC
8	680.C,671.AB	0.9	680.ABC
9	675.C,684.AC	0.5	675.C,684.AC
10	633.ABC,671.ABC	0.6	633.ABC,671.ABC
11	633.A,671.A,645.B	0.9	633.A,671.A,645.B
12	684.A,680.C,633.A	0.7	684.A,680.C,633.A
13	645.C,675.A,611.C	0.5	645.C,675.A,611.C
14	611.C,652.A,680.A	0.6	611.C,652.A,680.A,684.AC
15	646.B,680.B,675.C	0.8	646.B,680.B,675.C

IV. CONCLUSION

Distribution system fault classification is a challenging task due to measurement scarcity. This paper introduces a deep learning technique to tackle this problem for better outage management. The sequential CNN based identification is very accurate and fast. Also, the use of 2-class multi-label vector improves the computation complexity under the multi-fault scenario. The case study demonstrates that deep learning techniques can benefit power systems in traditionally challenging areas. The limitation of this method is relying on large training data set and having long training time. So, online training scheme will be implemented in the future to solve this problem.

REFERENCES

[1] A. Rendon, C. R. Fuerte, and J. G. Calderon, “Scada and pmu measurements for improving power system state estimation,” *IEEE Latin America Transactions*, vol. 13, no. 7, pp. 2245–2251, 2015.

[2] V. Miranda, J. Krstulovic, J. Hora, V. Palma, and J. C. Principe, “Breaker status uncovered by autoencoders under unsupervised maximum mutual information training,” 2013.

[3] Z. Zhang, W.-J. Lee, D. A. Wetz, B. Shrestha, J. Shi, A. Jackson, H. Honang, and J. Fielder, “Evaluation of the switching surges generated during the installation of legacy and smart electric metering equipment,” *IEEE Transactions on Industry Applications*, vol. 50, no. 3, pp. 2164–2172, 2014.

[4] T. Takagi, Y. Yamakoshi, M. Yamaura, R. Kondow, and T. Matsushima, “Development of a new type fault locator using the one-terminal voltage and current data,” *IEEE Transactions on Power apparatus and systems*, no. 8, pp. 2892–2898, 1982.

[5] P. D. Rodrigo, A. Pahwa, and J. E. Boyer, “Location of outages in distribution systems based on statistical hypotheses testing,” *IEEE Transactions on Power Delivery*, vol. 11, no. 1, pp. 546–551, 1996.

[6] Y. Liu and N. N. Schulz, “Knowledge-based system for distribution system outage locating using comprehensive information,” *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 451–456, 2002.

[7] S. T. Mak, “A synergistic approach to using amr and intelligent electronic devices to determine outages in a distribution network,” pp. 447–453, 2006.

[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, 2016.

[9] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” pp. 1725–1732, 2014.

[11] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, “Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition,” pp. 4277–4280, 2012.

[12] C. N. Dos Santos and M. Gatti, “Deep convolutional neural networks for sentiment analysis of short texts.” pp. 69–78, 2014.

[13] Q. Song, W. Sheng, L. Kou, D. Zhao, Z. Wu, and H. Fang, “Smart substation integration technology and its application in distribution power grid,” *CSEE Journal of Power and Energy Systems*, vol. 2, no. 4, pp. 31–36, 2016.

[14] M. Kawahara, T. Inoue, and Y. Nishio, “Image processing application using cnn with dynamic template,” pp. 1–6, 2010.

[15] R. C. Dugan and O. P. E. Model, “Opendss manual,” *Electric Power Research Institute*, 2013.

[16] A. Karpathy. Convolutional Neural Networks for Visual Recognition. [Online]. Available: http://cs231n.github.io/convolutional-networks/

[17] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” *arXiv preprint arXiv:1603.07285*, 2016.

[18] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[19] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” pp. 249–256, 2010.

[20] S. S. Haykin, S. S. Haykin, S. S. Haykin, and S. S. Haykin, *Neural networks and learning machines*. Pearson Upper Saddle River, NJ, USA:, 2009, vol. 3.

[21] L. Y. Deng, “The cross-entropy method: a unified approach to combinatorial optimization, monte-carlo simulation, and machine learning,” 2006.

[22] IEEE 13 Node Test Feeder. [Online]. Available: https://ewh.ieee.org/soc/pes/dsacom/testfeeders/