Line Faults Classification Using Machine Learning on Three Phase Voltages Extracted from Large Dataset of PMU Measurements

Abstract

An end-to-end supervised learning method is developed to classify transmission line faults in a two-year field-recorded dataset that includes synchronized measurements of three-phase voltages recorded by 38 Phasor Measurement Units (PMU) sparsely located in in the US Western Grid interconnection. Statistical analysis is performed to extract features from this large dataset to train Support Vector Machine (SVM), Random Forest (RF), and eXtreme Gradient Boosting (XGBoost) classifiers initially. The training further leverages a simulated dataset from a synthetic grid with 12 PMUs to increase the number of faults of types infrequently seen in the field-recorded dataset. Training the classification models with the combined dataset resulted in a classification accuracy of 97.7%. This is a significant improvement over 89.7% to 92.5% accuracy obtained by relying on the field-recorded dataset alone.

1. Introduction

In recent years, Synchrophasor technology has been complementing legacy supervisory control and data acquisition (SCADA) system [1]. The benefit of using PMUs is in their ability to take synchronized phasor measurements at 30 frames per second (fps) or higher. This allows utility operators to monitor the power system with much higher data resolution when compared to the unsynchronized measurement taken by SCADA scan every few seconds [1] [2]. The goal of real-time automated detection of short-duration events, such as the faults, has become more attainable. However, high data reporting rate coupled with a steady increase in the number of PMUs deployed in the power grid makes fault classification processing of historical or real-time streaming PMU data increasingly challenging. This greatly increases interest in applying Machine Learning (ML) technologies to automatically process and analyze the captured PMU datasets for efficient and accurate faults classification.

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000913.

To manage a large volume of data, several methods of feature extraction and ML techniques have been studied.

Principal Component Analysis (PCA) was applied to reduce the dimensionality of the dataset collected from PMUs to make event detection possible at early stages [3]. The same method is used to detect complex cascading events [4]. Minimum Volume Enclosing Ellipsoid (MVEE) was used as the method of feature extraction [5] followed by the Agglomerative Hierarchical Clustering method for event classification. Other event detection methods include the Detrended Fluctuation Analysis [6], Fast variant of Discrete S-Transform [7], and Signal Energy Transform [8], among others [9,10]. Normalized value of the wavelet coefficient energy is used as feature engineering method to detect events [11]. More recently, Dynamic Programming based Swinging Door Trending is used to precisely pinpoint the start time of events [12]. Some studies also investigated the application of event detection in distribution networks using Micro-PMU data, such as [13] where performance of SVM, KNN, and Decision Tree for event detection was compared. A Wavelet Transform-based feature engineering method was used to generate inputs for Convolutional Neural Network classification model [14].

We propose an innovative feature engineering and ML approach to classify different types of faults from historical PMU datasets automatically. The main contribution is in the ability to improve the accuracy of the classification by supplementing sparsely field-recorded PMU data with simulated data in the cases when the number of events of certain types are insufficiently observed in the field recordings.

The background of the difficulties in using sparse PMU measurements for fault analysis is discussed in Section 2. The insights into the field-recorded and simulated data used to extract features are given in Section 3. Section 4 elaborates on the way how the datasets have been utilized. Section 5 introduces the ML classifiers and explains the feature extraction and labeling. The experiments and the results for the ML classifiers are documented in Section 6. Conclusion are drawn in Section 7 followed by References.

2. Background

We study eleven types of faults that may occur on a transmission line (three phases of the power system are marked as A, B, C and the G stands for ground): A-G, B-G, C-G, AB, AC, BC, AB-G, AC-G, BC-G, ABC, and ABC-G. To classify the line faults, we extract three-phases voltages from PMU measurements. By doing that, we encounter two problems:

Problem 1*: It is hard to separate PP, PP-G, and 3P faults using field-recorded data.* The voltages may be measured at a distance from the fault location due to the sparsity of PMU locations (less than 5% of buses are covered by PMUs), which makes the voltage signal less distinguishable among different fault types. In addition, noise and system unbalance are adding challenges to differentiating these types of faults.

Problem 2*: There are less examples of PP, PP-G, and 3P faults as compared to P-G faults in the field-recorded dataset.* Due to the statistical rarity of occurrence, the PMU recordings have uneven representation of fault types (phase-to-ground faults are much more frequent than any other types).

We further illustrate Problem 1 in Figures 1 to 3 where the three-phase voltage measurements for phase-to-phase (AB), phase-to-phase-to-ground (AB-G) and three phase (ABC) faults, respectively are compared to the corresponding PMU measurements obtained from simulated faults. The measurements are visualized using OSIsoft's PI Vision software [14]. We make three observations based on Figures 1-3:

Observation 1 based on Figure 1: It is difficult to obtain accurate field-recorded measurements due to noise and system unbalance. If we observe Figure 1

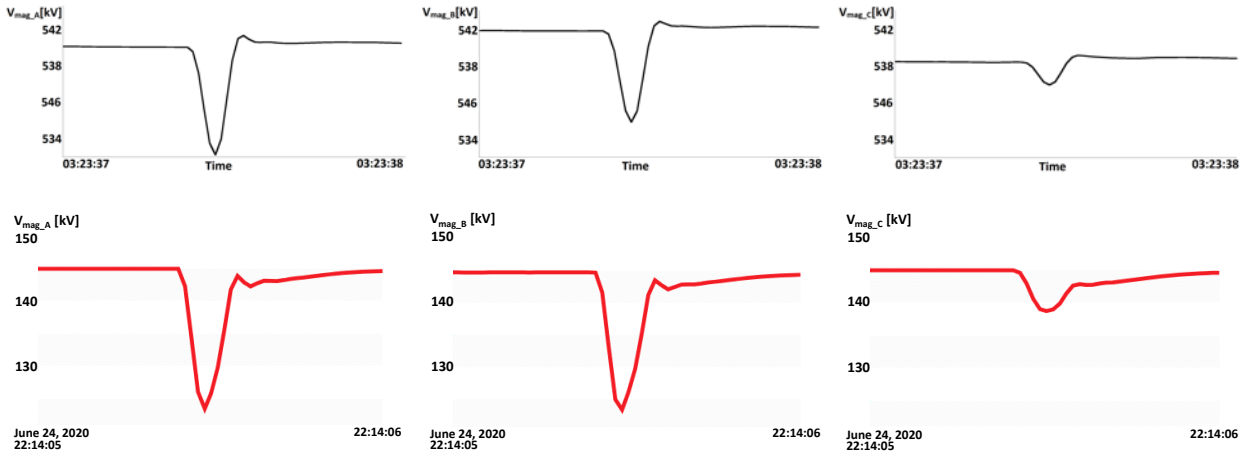

Figure 1. AB fault - Field-recorded (top) and simulated (bottom) PMU data (left to right: phases A, B and C)

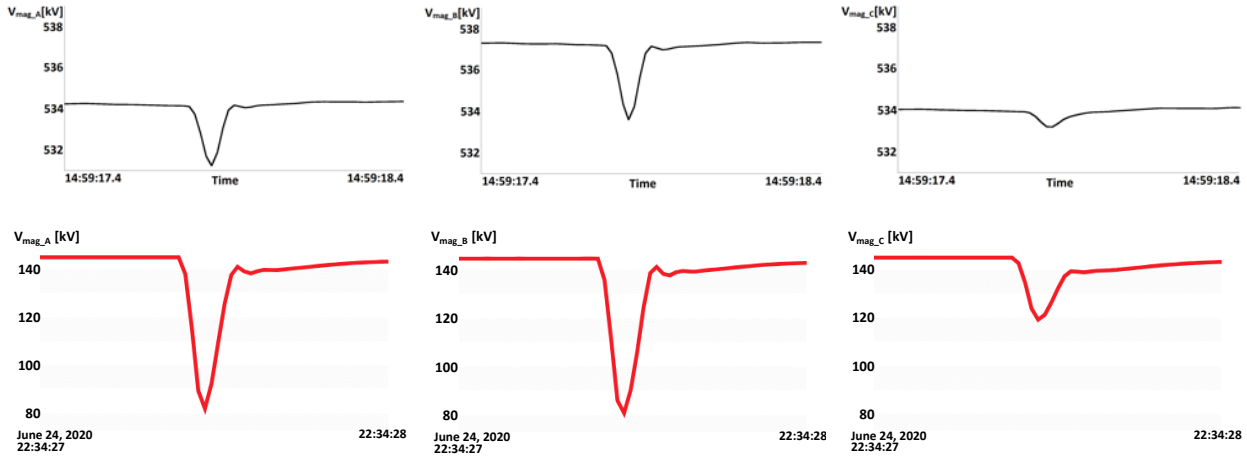

Figure 2. AB-G Fault Events - Field-recorded (top) and simulated (bottom) PMU data (left to right: phases A, B and C)

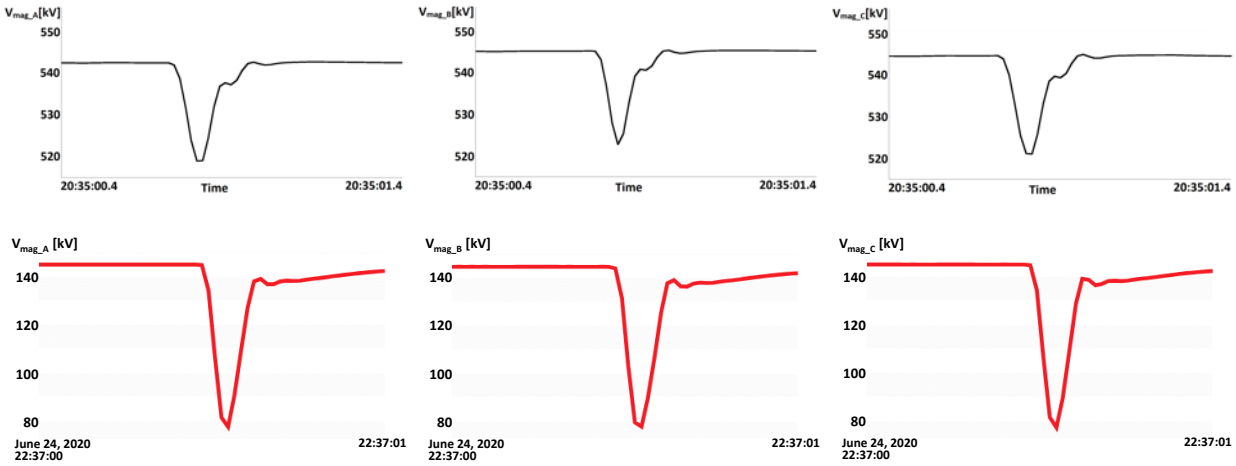

Figure 3. ABC fault event - Field-recorded (top) and simulated (bottom) PMU data (left to right: phases A, B and C)

with AB fault, we can see that the simulated example exhibits the same pre-fault voltage level on all three phases, equal voltage drops on phases A and B, and much smaller voltage drop on phase C compared to phases A and B, which is expected for a perfectly symmetrical system during the AB fault. However, field-recorded example in Figure 1 demonstrates that the there is a difference between both voltage levels and voltage drops between phases A and B during the same kind of AB event.

Observation 2 based on Figures 1 and 2: It is difficult to separate fault AB from AB-G in the field-recorded dataset. The two cases that exhibit the most similarities in field-recorded examples are phase to phase and phase- to-phase-to-ground waveforms as shown in Figures 1 and 2. The bottom three waveforms in each figure are obtained from simulation. Simulated waveform examples in Figures 1 and 2 demonstrate that it is easy to differentiate fault AB from AB-G due to the AB-G fault waveform always having a larger voltage drop on all phases than AB fault. The difference is not clearly detectable in field-recorded signals as seen in the corresponding top three waveforms in Figures 1 and 2 where it is much more difficult to distinguish between fault AB and AB-G recordings. These two cases (AB and AB-G) need selection of very precise thresholds to differentiate them automatically, which becomes challenging considering that in many scenarios these thresholds might be exceeded due to noise or different proximity of PMUs to the faults.

Observation 3 based on Figure 3: It is difficult to detect symmetrical ABC faults in the field-recorded dataset. When working with simulated data it is easy to detect three-phase faults by simply checking if the voltage drop is equal on all three phases, as can be observed from the simulated example in Figure 3. On the other hand, the field-recorded three-phase faults in Figure 3 do not have exactly the same voltage drop on all three phases, thus making it more challenging to separate this kind of event from some of the phase-to-phase-to-ground events.

The Problem 2 is illustrated in Figure 4, which shows an uneven statistics of different fault types in the field-recorded data collected in the Western Interconnection of the USA for the period of two years. As an example, the fault type P-G has far more recorded cases than fault types PP and PP-G combined.

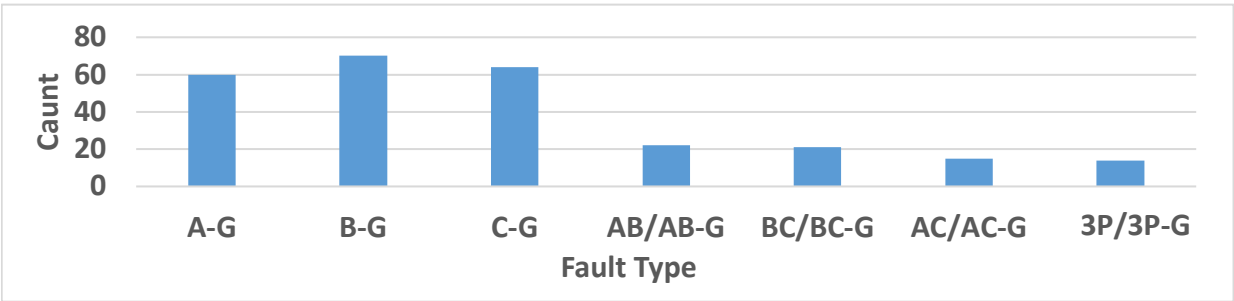

Figure 4. Distribution of faults in field-recorded data

Based on the above observations, our hypothesis is that both problems might be addressed by enhancing training data through carefully selecting PMU recordings of simulated measurements of incorrectly or insufficiently represented fault types in the field-recorded data. The combined field-recorded and simulated PMU measurement data are used to train ML algorithms to test a hypothesis that the resulting models are more accurate than models trained on field-recorded data alone.

In next section, we discussed the two sources of data by providing further details.

3. PMU Measurement Data

Field-recorded PMU measurement data provided by the electric utilities and simulated data coming from a synthetic grid have been used for this study. After visual inspection of field-recorded current magnitude measurements it was observed that the current magnitude change was not as prominent as the change observed in the voltage magnitude due to the large distance of PMUs from the fault location for most events in the dataset, so the currents were not used. Both datasets are preprocessed to extract the three-phase measurements of voltage magnitude. Next, the steps are taken to extract the features based on a statistical analysis of 2-second data windows. The duration of time window was decided empirically after multiple experiments on simulated data. This window selection gives high accuracy in terms of line fault-type classification. The simulated data was integrated into the field-recorded data to provide more balanced training dataset for the ML classification algorithms.

3.1. Field-Recorded PMU Data

3.1.1. Data Description. Datasets used include:

- Synchrophasor measurements from 38 PMUs located in the Western Interconnection of the USA collected for the period of two years. The dataset was anonymized by the provider by removing the information about geographical locations and any physical and technological characteristics of the PMUs or the electric grid to which they are connected. The PMUs under study have reported measurements of voltage and current magnitude and angle for each phase, positive sequence voltage and current magnitude and angle, frequency, and rate of change of frequency. For the reasons explained earlier, in this experiment, only the three phase voltage magnitudes were used to develop an automatic labelling system according to the phases affected by the fault.
- Historical event logs for the period of two years are stored as a CSV file. The event logs assigned by the data provider have inaccurate timestamps for the event start/stop times since the associated logs came from SCADA and were entered manually, which did not provide sufficient and accurate time-related details. We performed visual inspection to find the precise start time of each event and create a 2-second time window that contains it. Some of the events have a descriptor field that specifies the phases affected by the event. In some cases, this descriptor was provided in a form of P-G (phase-to-ground), P-P (phase-to-phase) without naming the phase. In these cases, visual inspection of the three phase voltage magnitudes was applied to determine the affected phase (A, B, or, C).

3.1.2. Data Preparation. While a dataset spanned field-recorded PMU measurements from 43 PMUs over the period of two years (2016-2017),only 38 PMUs were selected to proceed with feature extraction for the reason of a large fraction of bad data in the remaining five [16]. The data quality issues included missing and duplicate data, an excessive number of outliers, flat 60 Hz frequency recordings and erroneous time tags. Since the data was stored in the form of Apache Parquet files, Apache Spark [17] was used to retrieve the data. Python [18] was then used to implement the required analysis to prepare the data for the experiment. That stage focuses first on running the raw measurements through the calculations to extract six features, as described in section 4.1. These six features were used to decide which type of fault would be a correct label for an event, which then is assigned to the datapoint.

3.2. Simulated PMU Data

3.2.1. Data Description. For this research, Quanta Technology's Protection and Control test facility in Raleigh North Carolina was used to create a simulated dataset. The core element of the simulation system was a RTDS NovarCor real time simulation system with two cores. 12 PMU devices (four actual PMUs and 8 software emulated PMUs) were placed on a synthetic IEEE 14-Bus Power System (Figure 5) to monitor simulated transmission line faults. The simulated dataset includes measurements from 1,350 simulated faults at different locations with multiple combinations of fault resistances, locations and types. PMU data are streamed to a PDC (i.e. phasor data concentrator) (Figure 6) in IEEE C37.118 protocol and the data is archived and assembled to create the simulated dataset.

The faults were simulated on all lines also capturing the line fault clearing and reclosing sequences that include Direct Fault Clearing, Automatic Line Reclosing for Single Phase to Ground Faults, and Operator Line Reclosing for All Multi-Phase Faults. The simulated dataset has a separate event log file in which information for each simulated fault (i.e. time, location, resistance, and type) is provided.

3.2.2. Data Preparation. The data in the simulated dataset for each PMU includes three-phase voltage and current synchrophasors in the form of voltage/current magnitude, voltage/current angle, and frequency and ROCOF. For the fault classification training, magnitude from each phase's voltage phasor was extracted for all three phases, namely A, B, and C. Data windowing is performed by splitting the time-series data into consecutive windows of 2 seconds long each.

4.Methodology

In this section, we discuss our proposed method in three subsections: feature extraction, labeling, and integration of field-recorded and simulated data.

4.1. Feature Extraction

After data preparation, six features were extracted based on three-phase voltage measurements: AB_{diff}, BC_{diff}, CA_{diff}, XY_{diff}, YZ_{diff}, and ZX_{diff}. The features were extracted through the following four steps:

- Range of voltage for each PMU
- Aggregate range of voltage for all PMUs
- Difference between each two phases
- Ratio of differences between each two phases

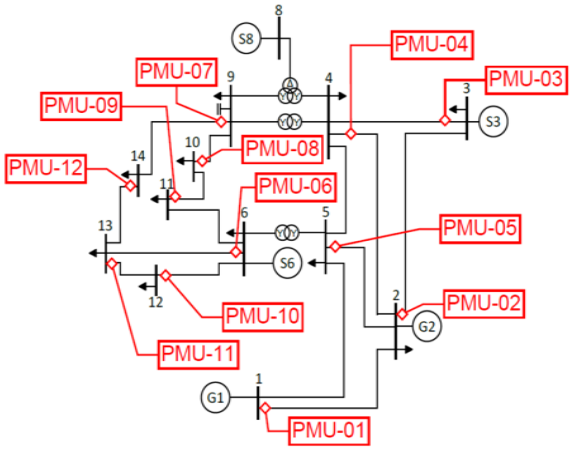

Figure 5. IEEE 14-bus power system with 12 PMU placement

Below, each step is discussed in detail.

Step 1- The range of voltage is calculated for each of the 38 PMUs (or 12 PMUs in simulations) over a 2-second window, where the minimum voltage measurement is subtracted from the maximum voltage measurement to compute the voltage range for each phase as

$$V_{range}(\phi_i) = \max(V_{mag}(\phi_i)) - \min(V_{mag}(\phi_i)) \quad (1)$$

where $V_{mag}(\phi_i)$ stands for the voltage magnitude of phase ϕ (A, B or C) for the i_{th} 2-second window.

Step 2- These ranges are then summed up for all PMUs as

$$SUM(V_\phi) = \sum_{i=1}^{\text{number of PMUs}} V_{range}(\phi_i) \quad (2)$$

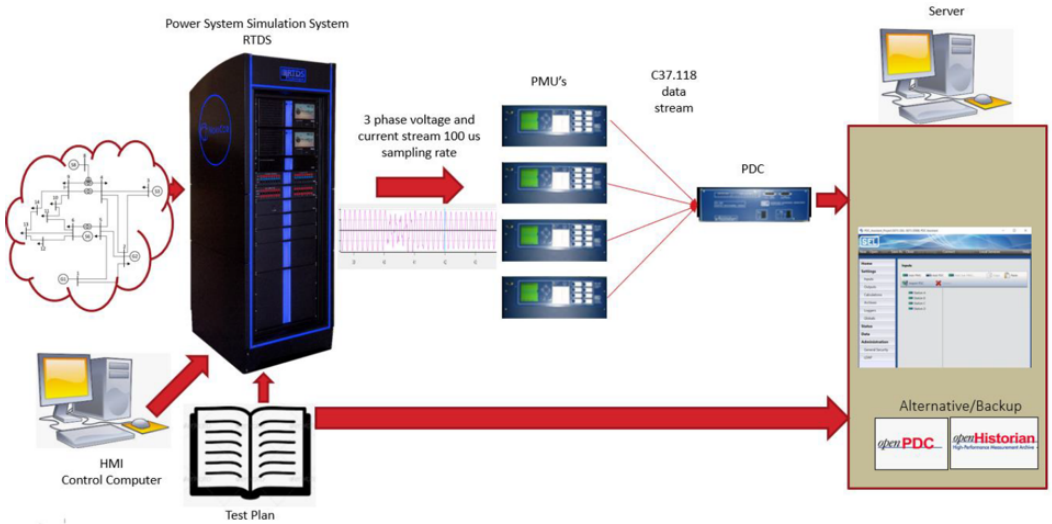

Figure 6. RTDS faults simulation framework

Step 3-The difference between each two phases is then calculated and signs of these three quantities are considered as the first three features

$$A_{to}B = SUM\,(V_A) - SUM\,(V_B) \quad (3)$$

$$B_{to}C = SUM\,(V_B) - SUM\,(V_C) \quad (4)$$

$$C_{to}A = SUM\,(V_C) - SUM\,(V_A) \quad (5)$$

$$AB_{diff} = sign(A_{to}B) \quad (6)$$

$$BC_{diff} = sign(B_{to}C) \quad (7)$$

$$CA_{diff} = sign(C_{to}A) \quad (8)$$

Last, the ratio between differences in voltage range mentioned in the third step is found such that the larger value is always divided by the smaller value. For example, if abs($A_{to}B$) > abs($B_{to}C$) then $R(AB_{to}BC)$ = abs($A_{to}B$) / abs($B_{to}C$). These ratios are then used to form remaining three features computed as described in equations (12-14):

$$X = R(AB_{to}BC) \quad (9)$$

$$Y = R(BC_{to}CA) \quad (10)$$

$$Z = R(CA_{to}AB) \quad (11)$$

$$XY_{diff} = sign(X - Y) \quad (12)$$

$$YZ_{diff} = sign(Y - Z) \quad (13)$$

$$ZX_{diff} = sign(Z - X) \quad (14)$$

4.2. Labeling

Intersecting sub-signal time windows were labeled with event log provided with the dataset. This step produces a binary label, which indicates whether an event occurred at this sub-signal or there was normal operation. The event log also contains a field "Descriptor" that has information of the type of fault that occurred (for example A-G, AB, 3P…).

Depending on values of AB_{diff}, BC_{diff}, CA_{diff}, XY_{diff}, YZ_{diff}, and ZX_{diff} that are extracted as described in section 4.1, the type of fault can be automatically determined as described below in Table 1. Automatic labeling is then performed for multiclass line faults as shown in Table 1. Each label represents occurrence of one line fault type. For example, a phase-to-ground fault (i.e. A-G, B-G, or C-G) is a combination of four labels. The labels are assigned as follows: for a phase-to-ground fault, if AB_{diff} = 1 and CA_{diff} = -1 and YZ_{diff} = 1 and ZX_{diff} = -1, then the label will be "A-G". Other line fault labels are described in Table 1. The automatic labeling in Table 1 only works well for separation of A-G, B-G, and C-G faults. When it comes to the PP, PP-G, and 3P faults majority of the events are mislabeled. Therefore, this automatic labeling system was combined with the machine learning model described in Sec. 5.

4.3. Integration of field-recorded and simulated data

Due to the limited number of examples per type of line fault in field-recorded PMU data (Figure 4) as discuss in the background section, simulated data with much more prominent fault types are combined with field-recorded data to generate an integrated training set (Figure 7) aimed to boost the line fault classification. As it was observed from Figure 4, some types of faults such as PP, PP-G, and 3P are less frequent in the field-recorded data. In the integrated version presented in Figure 7, simulated examples are added to make sure that each fault type has the same frequency of occurrence in the dataset. Data integration process of simulated data and field-recorded data is illustrated in Figure 8.

Table 2 shows the increase in the percentage of each fault type after simulated data is added to field-recorded data for each line fault type, resulting in much more balanced classes vs relying on field-recorded training data alone.

Table 1. Automatic labeling for seven types of line fault

Extracted Features	Type of faults
AB_{diff} = 1 and CA_{diff} = -1 and YZ_{diff} = 1 and ZX_{diff} = -1	"A-G"
AB_{diff} = -1 and BC_{diff} = 1 and XY_{diff} = -1 and ZX_{diff} = 1	"B-G"
BC_{diff} = -1 and CA_{diff} = 1 and XY_{diff} = 1 and YZ_{diff} = -1	"C-G"
BC_{diff} = 1 and CA_{diff} = -1 and XY_{diff} = 1 and YZ_{diff} = -1	"A-B/AB-G"
AB_{diff} = 1 and BC_{diff} = -1 and XY_{diff} = -1 and ZX_{diff} = 1	"A-C/AC-G"
AB_{diff} = -1 and CA_{diff} = 1 and YZ_{diff} = 1 and ZX_{diff} = -1)	"B-C/BC-G"
"3P/3P-G" will be assigned if the line fault is not one from all previous combinations	"3P/3P-G"

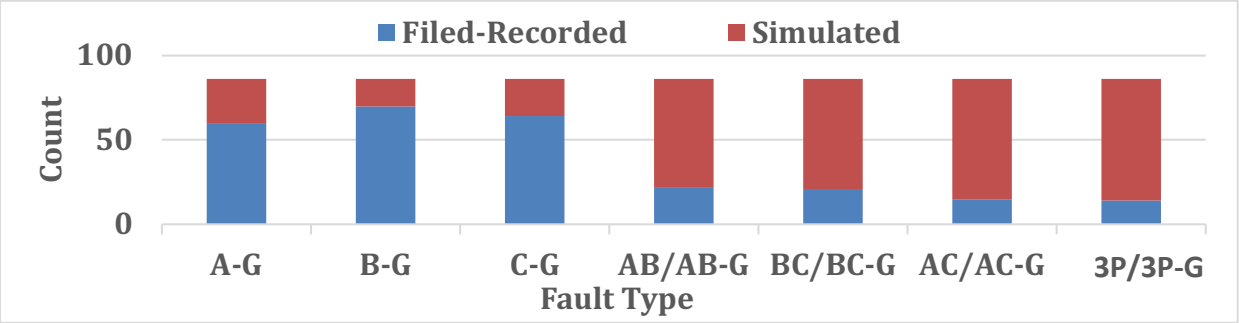

Figure 7. Integrated Data Distribution

5. Data Modeling

In this study, three classification models were considered to evaluate the proposed approach, where the performance of each model was compared before and after data integration. These three classifiers have discussed briefly in this section.

5.1. Classifiers

5.1.1. XGBoost Classifier. The first algorithm used in our experiment is eXtreme Gradient Boosting which is known as XGBoost [19]. XGBoost classifier implemented in the scikit-learn library (version 0.18.1) for Python with default parameters is utilized. XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework [20]. XGBoost consists of multiple classification decision trees. Each line fault is classified separately by each of the decision trees where each tree puts the classified line fault in one of the classes as described in Table1.

5.1.2. SVM Classifier. The second tested algorithm is Support Vector Machine (SVM), invented by Vapnik and Chervonenki [21]. In its most simple type SVM is applied to binary classification cases [22]. However, classifying types of line faults is a multiclass problem. The multiclass problem is broken down into multiple binary classification cases, which is also called one-vs-one. In the one-vs-one approach, each classifier separates cases in two classes, and comprising all one-vs-one classifiers leads to a multiclass classifier [23].

5.1.3. RF Classifier. The last considered ML algorithm is Random Forest Classifier. Random forest is one of the most commonly used ensembles learning algorithms, and it is implemented [24] in scikit-learn library (version 0.22.1) for Python where default parameters are utilized. It consists of multiple classification decision trees. So, each new event is classified separately by each of the decision trees where each tree of the forest gives a unit vote, assigning each input to the most probable class label. The final rule on the class is selected based on maximum votes [25]. The number of trees in the forest and how deep the trees

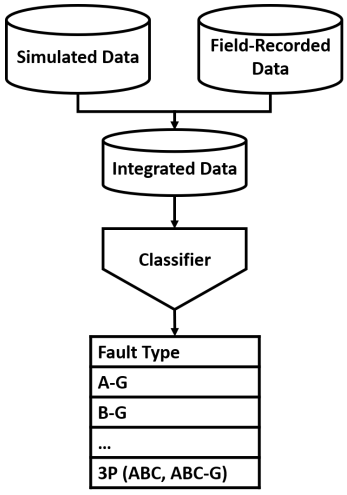

Figure 8. Train a classifier by two sources of data

Table 2. Percentage increases of faults after data integration

Line fault type	Before integration	After integration	Difference In percentage
A-G	60	86	**43 %**
B-G	70	86	**23 %**
C-G	64	86	**34 %**
AB/AB-G	22	86	**291 %**
BC/BC-G	21	86	**310 %**
AC/AC-G	15	86	**473 %**
3P/3P-G	14	86	**514 %**

are reflecting hyperparameters should be chosen for each specific problem. The reason for utilizing RF in our experiment is because RF is a fast method, robust to noise and it is an ensemble that can successfully identify non-linear patterns in the data

6. Classifier Evaluation

The results of evaluating SVM, RF, and XGBoost classifiers trained on filed-recored and on integrated training data are shown in Table 3. The performance of the fault-type classification models on out of sample filed-recorded data is measuresured using Weighted Precision, Weighted_Recall, and F1_score, which are suitable measures for multiclass classification problems [26]. The obtained result provide evidence that training any model on integrated data significantly improved test accuracy. The difference in accuracy among SVM, RF, and XGBoost models was minimal when relying on integrated training data while this was not the case when relying only on filed-recorded data alone (XGBoost was less accurate from SV and RF).

We show SVM results in mode detail for all three cases in Figure 9 (i.e. training on simulated data), Figure 10 (i.e training on filed-recorded data), and Figure 11 (i.e. using integrated training data). These results provide additional evidence that the accuracy using field-recorded PMU data alone was lower when compared to relying on combined field-recorded data enhanced by simulated faults to compensate for the types least present in the field-recorded data.

Figures 12 and 13 depict the precision-recall curve of the SVM classifier for each type of fault using filed-recorded data and integrated data, respectively.

Table 3. Fault classification performance across multiple evaluation metrics

Field-Recorded Data			
Models	**Weighted Precision**	**Weighted Recall**	**F1-score:**
SVM	92.50%	91.70%	90.10%
RF	91.70%	91.73%	90.25%
XGBoost	89.70%	90.44%	89.65%
Integrated Data			
SVM	**97.72%***	96.66%	96.66%
RF	97.70%	96.66%	96.65%
XGBoost	96.69%	96.66%	96.66%

The largest metric values are bolded, while '*' indicates SVM performance are relatively high compared to RF and XGBoost

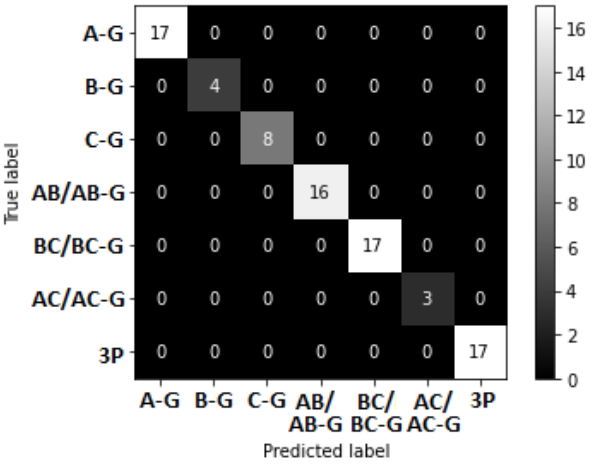

Figure 9. Confusion matrix for SVM trained on simulated data

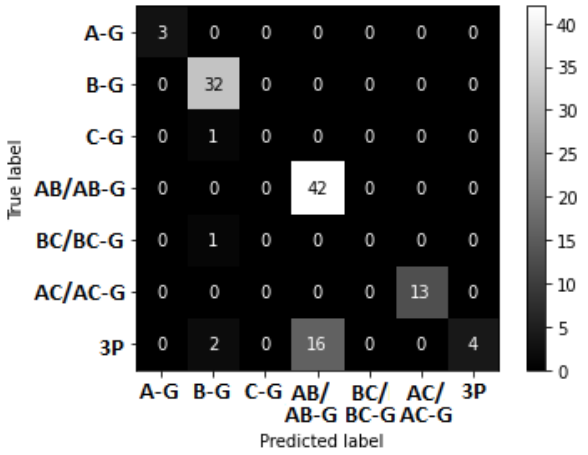

Figure 10. Confusion matrix for SVM trained on field-recorded data

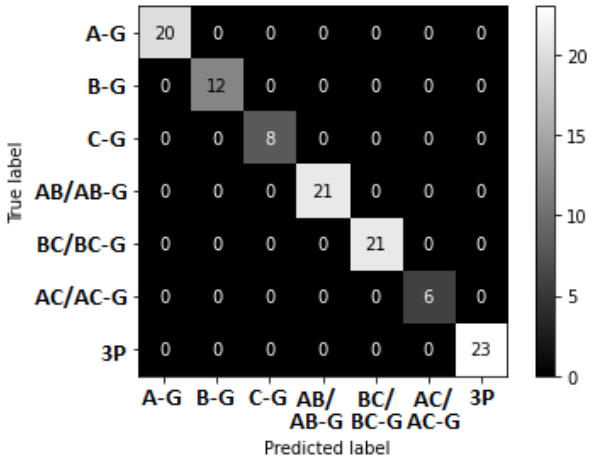

Figure 11. Confusion matrix for SVM trained on integrated data

Micro-average precision shown at Figures 12 and 13 was utilized to compare the overall performance for a seven class problem (increase from 0.773 to 0.993 when using integrated data). Micro-average of precision measures the true positives (TP) and false positives (FP) for each class [29] and computes the aggregated precision score as:

$$\text{Micro\_average\_of\_precision} = \frac{\text{TPsum}}{\text{TPsum} + \text{FPsum}} \quad (15)$$

In both figures precision-recall curves are also shown for each fault type. Performance was much improved when training the model on integrated data for types of faults insufficiently present in field-recorded training data since t data deficiencies were reduced by including simulated cases of these types.

7. Conclusions

In this study, we propose a novel method for line faults classification using machine learning on a large dataset of PMU measurements. In our approach integrated simulated and field-recorded data of three phases voltages were used to classify faults in the electric grid. This data enhancement has helped training more accurate ML models. The evidence is provided in this study that it is beneficial to train classification models using integrated field-recorded PMU data and simulations vs relaying on filed-recorded data alone when certain types of events of interest are insufficiently represented in field-recorded data over the training period.

Disclaimer

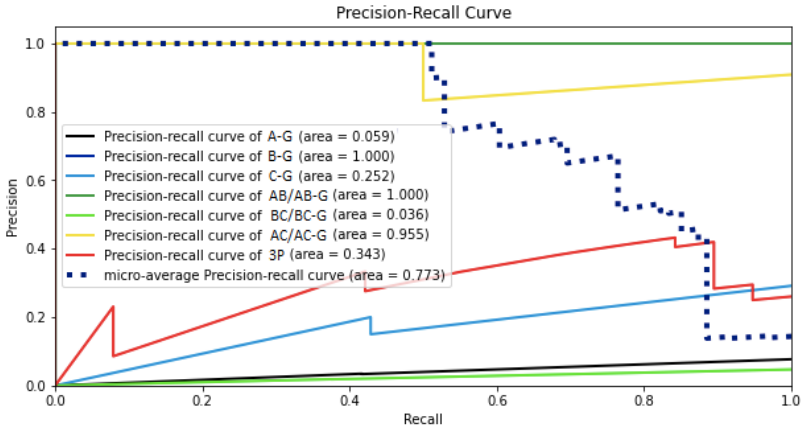

Figure 12. Precision-recall curve of SVM for classifying line fault type in field-recorded data

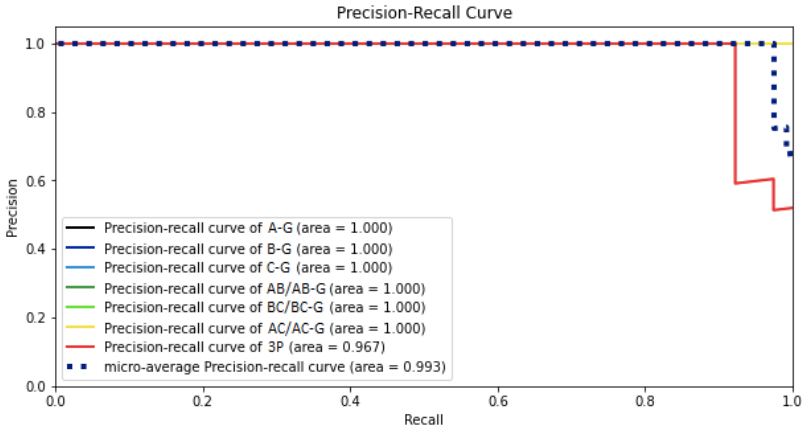

Figure 13. Precision-recall curve of SVM for classifying line fault type using integrated data

service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] M. Kezunovic,S.Meliopoulos, S. Venkatasubramanian, V. Vittal, "Application of Time-Synchronized Measurements in Power System Transmission Networks," Springer, ISBN 978-3-319-06218-1, 2014.

[2] M. Patel, S. Aivaliotis, E. Ellen et al., "Real-time application of synchrophasors for improving reliability", *NERC Report*, Oct 2010.

[3] L. Xie, Y. Chen, P. R. Kumar, "Dimensionality Reduction of Synchrophasor Data for Early Event Detection: Linearized Analysis," IEEE Trans. Power Systems, vol. 29, no. 6, pp. 2784-2794, Nov. 2014.

[4] M. Rafferty, X. Liu, D. M. Laverty, S. McLoone, "Real-Time Multiple Event Detection and Classification Using Moving Window PCA," IEEE Trans. Smart Grid, vol. 7, no. 5, pp. 2537-2548, Sep 2016.

[5] O. P. Dahal, S. M. Brahma, H. Cao, "Comprehensive Clustering of Disturbance Events Recorded by Phasor Measurement Units," *IEEE Trans. Power Delivery*, vol. 29, no. 3, pp. 1390-1397, Jun. 2014.

[6] M. Khan, P. M. Ashton, M. Li, G. A. Taylor, I. Pisica, J. Liu, "Parallel Detrended Fluctuation Analysis for Fast Event Detection on Massive PMU Data," IEEE Trans. Smart Grid, vol. 6, no. 1, pp. 360-368, Jan 2015.

[7] M. Biswal, S. M. Brahma, H. Cao, "Supervisory Protection and Automated Event Diagnosis Using PMU Data," IEEE Trans. Power Delivery, vol. 31, no. 4, pp. 1855-1863, Aug. 2016.

[8] R. Yadav, A. K. Pradhan, I. Kamwa, "Real-Time Multiple Event Detection and Classification in Power System using Signal Energy Transformations," IEEE Trans. Industrial Informatics, vol. 15, no. 3, Mar. 2019, pp. 1521-1531.

[9] M. Biswal, Y. Hao, P. Chen, S. Brahma, H. Cao, and P. DeLeon, "Signal features for classification of power system disturbances using PMU data," Proc. Power Syst. Comput. Conf., Jun. 2016, pp. 1–7.

[10] S. Brahma, R. Kavasseri, H. Cao, N. R. Chaudhuri, T. Alexopoulos, Y. Cui, "Real-Time Identification of Dynamic Events in Power Systems Using PMU Data, and Potential Applications-Models, Promises, and Challenges," IEEE Trans. Power Delivery, vol.32, no. 1, Feb. 2017, pp. 294-301.

[11] D.-I. Kim, T. Y. Chun, S.-H. Yoon, G. Lee, Y.-J. Shin, "Wavelet-Based Event Detection Method Using PMU Data," IEEE Trans. Smart Grid, vol. 8, no. 3, May. 2017, pp. 1154-1162.

[12] M. Cui, J. Wang, J. Tan, A. R. Florita, Y. Zhang, "A Novel Event Detection Method Using PMU Data With High Precision," IEEE Trans. Power Systems, vol. 34, no. 1, Jan. 2019, pp. 454-466.

[13] A. Shahsavari, M. Farajollahi, E. Stewart, E. Cortez, H. Mohsenian-Rad, "Situational Awareness in Distribution Grid Using Micro-PMU Data: A Machine Learning Approach," IEEE Trans. Smart Grid, vol. 10, no. 6, Nov. 2019, pp. 6167-6177.

[14] S. Wang, P. Dehghanian, L. Li, "Power Grid Online Surveillance through PMU-Embedded Convolutional Neural Networks," IEEE Trans. Industrial Applications, vol. 56, no. 2, Mar.-Apr. 2020, pp. 1146-1155.

[15] K. Kashinath, et al. "Physics-informed machine learning: case studies for weather and climate modelling." Philosophical Transactions of the Royal Society A 379.2194 (2021): 20200093.

[16] OSlsoftPIVision,[Online]Available: https://www.osisoft.com/pi-system/pi-core/visualization

[17] NASPI PMU Applications Requirements Task Force, "PMU Data Quality: A Framework for the Attributes of PMU Data Quality and a Methodology for Examining Data Quality Impacts to Synchrophasor Applications," NASPI-2017-TR-002, March 2017.

[18] Apache Spark, [Online] Available: https://spark.apache.org/

[19] Python, [Online] Available: https://www.python.org/Brownlee, Jason. "How to Develop Your First

[20] XGBoost Model in Python." Machine Learning Mastery, 18 Jan. 2021, machinelearningmastery.com/develop-first- xgboost-model-python-scikit-learn/.

[21] "Multiclass & Multilabel Classification with XGBoost [Online].Available:https://gabrielziegler3.medium.com/multiclass-multilabel-classification-with-xgboost.

[22] Prudhvi, Poorna. "Support Vector Machines." Medium, Medium, 13 Oct. 2017, medium.com/@poornaprudhvi/support-vector-machines-bf0242ade522.

[23] "Sklearn.multiclass.OneVsOneClassifier." Scikit, scikitlearn.org/stable/modules/generated/sklearn.multiclass. OneVsOneClassifier.html.

[24] J. Browniee, "One-vs-Rest and One-vs-One for Multi-Class Classification," April 13, 2020. [Online].Available: https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/

[25] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, 2001.

[26] T. Yiu, "Understanding Random Forest," Medium, Towards Data Science, June 12, 2019. [Online] towardsdatascience.com/understanding-random-forest-58381e0602d2.

[27] J. Brownlee, "Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning," Machine Learning Mastery, 2020.

[28] J. Mohajon, "Confusion Matrix for Your Multi-Class Machine Learning Model," May 29, 2020. [Online] Available: https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826.

[29] B. Shmueli, "Multi-Class Metrics Made Simple, Part II: the F1-Score." Medium, Towards Data Science, 3 July 2020, towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1.