Machine Learning Using a Simple Feature for Detecting Multiple Types of Events From PMU Data

Tatjana Dokic[1], Rashid Baembitov[1], Ameen Abdel Hai[2], Zheyuan Cheng[3], Yi Hu[3], Mladen Kezunovic[1], Zoran Obradovic[2]

[1]Electrical & Computer Engineering, Texas A&M University, College Station, USA
kezunov@ece.tamu.edu

[2]Computer and Information Sciences, Temple University, Philadelphia, USA

[3]Quanta Technology, Raleigh, USA

***Abstract*— This paper describes simple and efficient machine learning (ML) methods for efficiently detecting multiple types of power system events captured by PMUs scarcely placed in a large power grid. It uses a single feature from each PMU based on a rectangle area enclosing the event in a given data window. This single feature is sufficient to enable commonly used ML models to detect different types of events quickly and accurately. The feature is used by five ML models on four different data-window sizes. The results indicated a tradeoff between the execution speed and detection accuracy in variety of data-window size choices. The proposed method is insensitive to most data quality issues typical for data from field PMUs, and thus it does not require major data cleansing efforts prior to feature extraction.**

***Index Terms*-- Big data, Event detection, Machine learning, Phasor measurement units, Power system faults, Time series analysis.**

I. INTRODUCTION

Synchrophasor measurements from Phasor Measurement Units (PMUs) are widely used for off-line post-event analysis and improved real-time situational awareness in power system operations. Detecting various types of events quickly and accurately is an important first step. Application of machine learning (ML) techniques to detect events has been extensively investigated in the last few years [1].

Multiple feature extraction methods including Discrete Fourier Transform, Discrete Wavelet Transform, Principal Component Analysis (PCA), Fast variant of Discrete S-Transform (FDST), and domain specific shapelets; using ML algorithms such as k-nearest neighbor (KNN) and Support Vector Machine (SVM) were compared [2,3]. The normalized value of the wavelet coefficient energy is used as the feature to detect events in [4]. Dynamic Programming based Swinging Door Trending was used [5]. Performance of SVM, KNN, and Decision Tree for event detection based on Micro-PMU data was compared for an application in distribution networks [6]. A Pseudo Continuous Quadrature Wavelet Transform was used to generate features for Convolutional Neural Network (CNN) [7]. A specialized method for detection and classification of multiple events was developed based on Signal Energy Transform [8]. The use of Minimum Volume Enclosing Ellipsoid (MVEE) to extract a set of features from recorded PMU signals is reported in [9]. PCA is applied to reduce the dimensionality of the PMU dataset [10,11]. The parallel version of the Detrended Fluctuation Analysis was developed for fast event detection [12]. The FDST is used to extract features in a form of Time Frequency Representation [13]. Hierarchical CNN approach to event detection was recently developed [14]. Transfer learning was used to reduce the required number of labels [15]. SVM and ensemble classifier based on Bagged Trees were also used for classification with the consideration of cyberattacks causing false event detection [16]. Deep Neural Network (DNN) enforced with information loading was used for event detection and classification in a large PMU dataset from the Eastern Interconnect of the U.S.A. [17].

Previous studies have limitations when historical dataset containing thousands of imprecise raw event labels are analyzed. Additional challenges involve measurements with missing or duplicated data, out-of-range data, etc., which violate some of common ML assumptions related to distribution stationarity, smooth mapping of explanatory variables to response, large signal to noise ratio, negligible effects of ignored information and sufficient precision of labels.

Our contribution is a novel method to transform raw PMU signal to a small number of informative features that can be used for ML based event detection. Our feature, based on a rectangle area (RA) capturing the voltage magnitude and frequency measurements within a data window, offers several advantages: 1) quick extraction of events from a large historical dataset by using only one *RA* feature per PMU for a given time window; 2) ML results without the need for additional data preprocessing, subsampling, or cleansing prior to feature extraction; and 3) highly accurate event detection when used with different ML models.

Data preparation is discussed in Section II, methodology in Section III, and results in Section IV, followed by conclusions and references.

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

II. Data Preparation

A. Field PMU data

The data used for this research contains measurements from 38 field PMUs recorded over two years (2016-2017) provided in an Apache Parquet storage format (~6 TB). The data is captured from the Western Interconnection (WI) in the USA that has thousands of buses. The dataset is anonymized by the provider for security reasons by eliminating designation of physical location of PMUs and the grid topological information making it extremely difficult to detect events using various causality correlations. 24 out of 38 PMUs reported data at 60 frames per second (fps), whereas the remaining 14 PMUs at 30 fps. Each PMU measured one positive sequence voltage and current, but some also included three phase phasors, as well as frequency and rate of change of frequency (ROCOF). For this study we use measurements of frequency and positive sequence voltage. The PMUs are connected to several voltage levels in the transmission network. Apache Spark and Python script are used to access and analyze the dataset stored on a high-performance server cluster. An event log for the dataset is also provided with 1-minute resolution labels (Raw Labels).

Another main challenge when dealing with field PMU measurements is data quality issues [18,19]. The PMU dataset we use for this study contains multiple such issues:

- *Missing data*: individual PMU datasets have between 0.7% and 30% of missing positive sequence magnitude and frequency measurements, respectively. Overall, 3% of measurements are missing.
- *Data duplicates* [19]: 14 out of 38 PMUs contain data duplicates whose number varies from day to day and reaches 2.6 x 10^6 frames on some days.
- *Out-of-range data:* This includes all the "impossible values" that would never be seen in a real power system. Frequency measurements from 5 out of 38 PMUs have out-of-range values, ranging from having only 1 wrong point for one PMU, to 4% of the measurement points for another. No out-of-range date points were detected in the voltage magnitude measurements. To detect these out-of-range data points, the following criteria is used: Voltage magnitude > (rated voltage)*2 or voltage magnitude < 0; and Frequency > 70 Hz or < 50 Hz.

B. Event types of interest

Our method is aimed at detecting events from sparsely located PMUs using imprecise event log labeled at 1-minute resolution described by the data provider as: a) forced line outage (such as short-circuit faults causing line outages), b) fundamental frequency deviations, and c) forced transformer outages. The goal is to use the discriminative features of these events to separate them from normal operations. An example of positive sequence voltage magnitude and frequency measurements during a fault related forced line outage event is shown in Fig. 1 where the voltage magnitude dip is the main event characterization. Some local (only on selected PMUs) fluctuations of frequency are also observed. In Fig. 2, measurements during fundamental frequency event are presented. One can observe large frequency drop and a small voltage magnitude dip on all PMUs. Fig. 3 indicates voltage and frequency measurements for a combination of forced transformer and line outages where the voltage magnitude dip and frequency fluctuations are the main characterizations. In all three types of events, significant deviations from the normal operation signal are observed for most of the events, but not always. For some events, the observed deviations in PMU measurements are much smaller, probably due to the large distance between the event location and locations of available PMUs, which makes it very difficult to separate them from normal events. This makes it impossible to build a threshold-based event detection using only the *RA* feature without the ML. Howevere, if we use the *RA* feature as an input to the ML model, very good performances can be achieved as demonstrated in the Results section. Due to the size of the full dataset (~6 TB) on which the method is applied to extract the events, the ML process is made much faster by extracting only one *RA* feature per PMU that captures the product of voltage magnitude difference and frequency difference within a data window. In general, our method can detect any event that contains a significant change in voltage magnitude and/or frequency on at least one of the PMUs. The method was not aimed at detecting oscillation or other events that do not exhibit such properties.

C. Event Labels

The event log includes a total of 1,472 forced events within Line, Frequency, and Transformer categories. The imprecise start time with 1-minute resolution makes it very difficult to correctly chose the data-window position and size to capture the event of interest for feature extraction. To facilitate more accurate learning with a higher quality training data, we performed a visual inspection of most of the PMU measurements indicated by the event log to find the precise start time of each event. Out of 1,472 events, we were able to

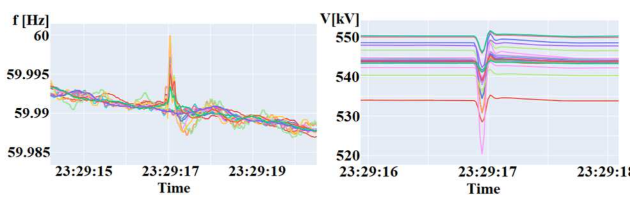

Fig. 1 Frequency (left) and voltage magnitude (right) of a line outage event

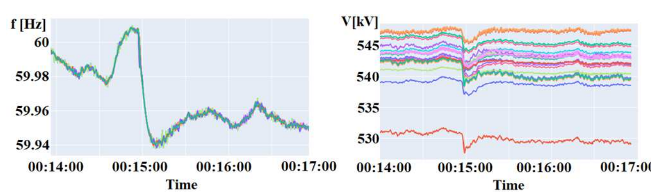

Fig. 2 Frequency (left) and voltage magnitude (right) of a frequency event

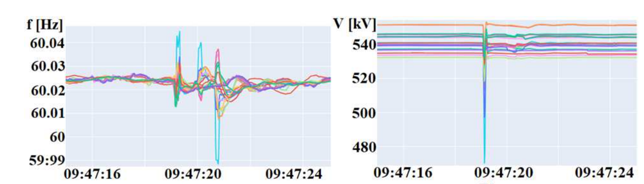

Fig. 3 Frequency (left) and voltage magnitude (right) during a combination of transformer event and line event

visually confirm and labeled more precisely 1,038 events with Cleaned Labels. In Fig. 4, we plot a histogram with the number of events in each time span according to the temporal distance between the event start time reported in the event log and the event start time determined using visual inspection of the PMU data. As it may be observed in Fig. 4, most of the events occurred within the (-6sec, +54 sec) window from the start time reported in the event log. There were 52 (~4%) events occurring outside of this time window. We separate the labels into two categories: Raw Labels created based on the start time from the original 1,472 events taken from the event log, and Cleaned Labels created based on the resulting 1,038 events of the visual inspection of the PMU data. Furthermore, we investigate different time window durations. We are unable to use data-windows shorter than 1-min using Raw Labels due to the uncertainty in the distance between the event log start time and actual event start time as demonstrated in Fig. 4. Because of the overlapping event windows start times (multiple events having a start time within one minute), we are unable to create Cleaned Labels for windows longer than 1-min.

For training of the ML models for event detection, we also need examples of normal operation that do not contain any types of events. We have observed that the event log does not contain all the events "seen" by the PMUs. Thus, we cannot assume that all the time periods between the events in the event log represent normal operation despite the fact that a large percentage does. In our study, the normal operation examples are taken from time periods between the events of the event log in two ways:

1) Not visually inspected: We first exclude the time segments of events based on the event start times and end times in the event log. Then we select all the time segments that are longer than 27 minutes from the remaining time segments. From each of the selected time segments we take two 3-min data-windows exactly from the middle point as the normal operation examples. We do this to ensure that the normal operation examples are at least 10 min away from any event time segment. These examples are not visually inspected to confirm if they truly represent normal operation. This set of normal operation examples is used for training with Raw Labels (1-min and 3-min).

2) Confirmed by visual inspection: We select 923 1-min data-windows from the original PMU dataset as normal operation example based on visual inspection. These normal operation examples are used for training with the Cleaned Labels of all durations. For the cases of time windows of 30-sec and shorter, two separate time windows are extracted from the 1-min examples. The number of labels in each event log in each category and time window selection criteria are presented in Table I.

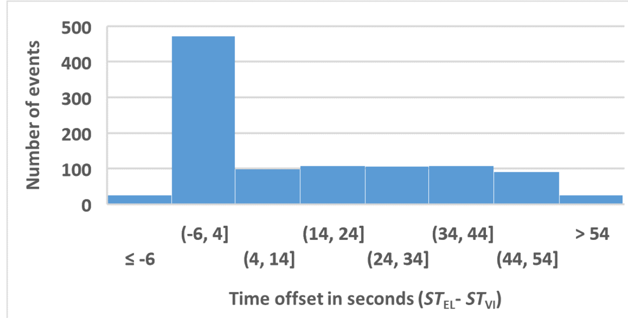

Fig. 4 Histogram of the temporal distance between the event start times of corresponding Raw Labels and Cleaned Labels

Having two sets of labels enables us to compare the performances of the event detection by various ML models trained with: 1) raw event and normal operation labels that may be created without any visual inspection, and 2) cleaned event and normal operation labels that are created after visual inspection. Results for the event detection on Raw Labels show performance of the method when it is automatically applied to the dataset with Raw Labels. Results for the event detection on Cleaned Labels illustrate how the method would perform if the original dataset had much cleaner event labels.

III. METHODOLOGY

Our methodology is selected to deal with specific challenges of the provided dataset, in particular:

- Due to a large amount of historical PMU data, it makes it time consuming to apply extensive data cleansing and feature engineering. Defining computationally simple feature that does not require extensive data cleansing is highly desirable.
- For supervised ML training, labels need to be much more precise and accurate, particularly regarding the start and end time of an event. The visual inspection is needed to understand what differentiates events from normal operation, and confirm exactly when the events actually occurred. We partition the files into smaller time windows to position an event inside the window to capture their respective lengths more precisely to see the impact.
- Events are occurring relatively rarely while the normal operation conditions vary with time constantly, making it hard to differentiate an event from normal operation particularly when all PMUs are far away from an event's location. We use a set of visually confirmed normal operation examples from different time periods to better distinguish events with various normal operation states. To improve the confidence in normal operation selection

Table I. Number of labels per category and window selection method

Event Log	# Event Labels	Event Start Time	Event End Time	# Normal Labels	Normal Selection Method
3-min Raw Labels	1472	ST_{EL} – 1 min	ST_{EL} + 2 min	2548	Randomly selected
1-min Raw Labels	1472	ST_{EL} – 6 sec	ST_{EL} + 54 sec	2548	Randomly selected
1-min Cleaned Labels	1033	ST_{VI} – 5 sec	ST_{VI} + 55 sec	923	Based on the visual inspection
30-sec Cleaned Labels	1038	ST_{VI} – 2 sec	ST_{VI} + 28 sec	1846	Based on the visual inspection
5-sec Cleaned Labels	1038	ST_{VI}	ST_{VI} + 5 sec	1846	Based on the visual inspection

ST_{EL} - start time of the event based on the event log ST_{VI} - start time of the event based on the visual inspection by the domain expert these examples are taken from the measurements that are completely outside the reported events in the event log.

As a result of better understanding the correlation between normal states and events, and between labeled time and event occurrence, we come up with the RA feature as the best approach to selecting the time windows corresponding to the most prominent signal behavior for different types of events. We then use the RA features across different data windows from various PMUs to train variety of conventional state-of-the-art ML models. The outcome, besides contributing to a ML model to detect events, also offers an understanding why proper time-stamping of events matters, and how to deal with the situations with labels characterized by imprecise time-stamps. We do not perform any other cleansing of the raw dataset prior to feature extraction, which saves significant time.

A. Feature Extraction

The underlying premise of this paper is that all considered types of events have one thing in common: a deviation from the normal operation visible on measurements of frequency, voltage magnitude, or both. In the case of normal operation, the difference between f_{min} and f_{max} (V_{min} and V_{max}) within a short time window (~1-min) is small. In case of any of the mentioned event types, this difference becomes larger in the same time window. Based on this experience, we combine the difference between f_{min} and f_{max} and V_{min} and V_{max}, by enclosing the signal into a rectangle area *RA* defined as:

$$RA = (f_{max} - f_{min}) * (V_{max} - V_{min}) \tag{1}$$

where f_{max} and f_{min} are maximum and minimum measured frequency, and V_{max} and V_{min} are maximum and minimum measured positive sequence voltage magnitude, inside the selected time window, for example 1-min window in Fig. 5.

B. Machine Learning Models

The features for a single time window are combined into a feature matrix $\boldsymbol{X}$ where each row contains 38 *RA* features, one corresponding to each observed PMU. The number of instances is equal to number of time windows in the event log (for example, in the case of 1-min Raw Labels, the number of instances is equal to 1472 (events) + 2548 (normal) = 4020. The labels vector "y" is created for each time window as ('1' – in case of a reported event, and '0' – in case of normal operation) and combined into a vector.

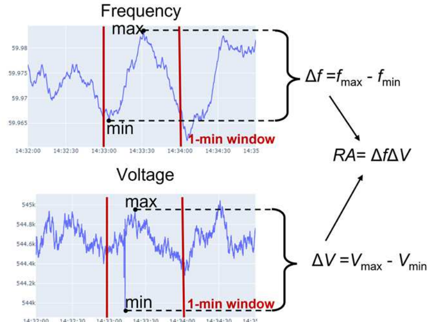

Fig. 5. Extraction of Rectangle Area feature

The feature matrix $\boldsymbol{X}$ and labels vector "y" are then ingested as inputs to a variety of ML algorithms aimed to learn a 38-dimensional function that maps the given time window to one of 2 categories (normal operation or event of interest); $y \in \{0,\ 1\}$. We use the *scikit-learn* library for ML in Python [20]. Four ML algorithms from *scikit-learn* library are tested: Random Forest (RF), K-Nearest Neighbor (KNN), AdaBoost (AB), Gaussian Naive Bayes (GNB). In addition, we also use the Catboost (CB) model from the library available in [21].

C. Response to Data Quality Problems

As mentioned in Sec. III.A. there are several data quality issues with the dataset. Here we provide a discussion on how our method responds to these data quality issues.

Missing data has minor impacts on the data model performance. We only need 2 data points within a single time window to extract our feature. For example, for a 1-min window on a 30 fps we only need 2 out of 1,800 points. In case there are no two points in the time window for a given PMU, the *RA* is zero. The main principle of this method is that smaller *RA* corresponds to normal operation, and larger *RA* corresponds to events. In a case PMU reports zero as an *RA* value, this PMU is simply not participating in the claim that there is an event, and a decision is made based on the remaining PMUs. A similar approach is taken if there are at least two points available within the time window, but they do not capture the event due to large number of missing points. In that case the *RA* for that particular PMU will not be zero, but it will be smaller than values captured by PMUs without missing data. This results in that PMU not participating in the claim that there is an event. Since we have 38 PMUs, and the majority of them are available for all the observed time windows, the impacts of missing data are negligible. The benefit of our method is that it does not require any missing data characterization or replacement, which makes it easier to implement and much faster to execute.

Data duplicates do not affect the data model performance in any way since we extract a single min and max value from each data window. This way duplicated values are automatically discarded during the feature extraction.

Out-of-range data do affect the data model performance since they can cause a high value of *RA* that can be easily mistaken for an event. There are two ways to remove such data: a) from the raw dataset prior to feature extraction, or b) from the pre-processed data after the RA feature extraction. The first method would be more accurate, but it is costlier computationally due to the large volume of the raw dataset. We decided to use the second method. The result is that only negligibly small fraction consisting of 11 values of *RA* needed to be discarded and replaced with zeros. The threshold for removal is RA_T = 20 MVHz = 1000 kV * 20 Hz, based on the maximum range of voltage of 1000kV (2 x 500kV, where 500kV is max rated voltage) multiplied by maximum range of frequency of 20 Hz (50 Hz < f < 70 Hz).

IV. Results

Hyperparameters tuning was performed using a grid search to select the hyperparameters that yield the best performances. The hyperparameters that were set for the five classifiers are listed in Table II.

Table III lists the results for the five classifiers using cross-validation for all five event log types. The split of the data into training and test sets is performed with Stratified K-Folds cross-validation with 5 folds. To ensure that each fold has the same proportion of observations with a given class outcome value, the same number of time windows from each event type were chosen. Five performance measures were reported: Area Under the Receiver Operating Curve (AUC), Area Under the Precision Recall Curve (AUPRC), Precision, Recall, and F-1 score [22].

As observed, from the Table III, the Random Forest outperforms all the other algorithms in most cases. Catboost is a close second place. Gaussian Naive Bayes provides the best precision in most cases. All the models have higher precision than recall. The study of the fundamental algorithmic reasons for such performance is out of the scope of this paper.

Table II. Selected Hyperparameters for the Binary Classifiers

AB(n_estimators=200, learning_rate=0.3)
RF(max_depth=200, n_estimators=150, random_state=42)
GNB(var_smoothing=1e-12)
KNN(n_neighbors=10, weights='distance', leaf_size=100)
CB(verbose=False, learning_rate=0.01, depth=8)

Table III. Model performance on five event log types

Event Log Type	Model	AUC	AUPRC	Preci-sion	Recall	F1-score
3-min Raw Labels	RF	**0.93**	**0.92**	0.90	**0.81**	**0.85**
	CB	0.92	0.91	0.91	0.77	0.83
	KNN	0.91	0.90	0.88	0.79	0.83
	AB	0.89	0.88	0.90	0.72	0.80
	GNB	0.87	0.85	**0.94**	0.54	0.69
1-min Raw Labels	RF	**0.92**	**0.92**	0.95	**0.81**	**0.87**
	CB	0.91	0.91	0.95	0.76	0.85
	KNN	0.90	0.91	0.93	0.78	0.85
	AB	0.89	0.89	0.95	0.71	0.81
	GNB	0.87	0.86	**0.96**	0.60	0.74
1-min Cleaned Labels	RF	**0.98**	**0.99**	0.97	**0.92**	**0.94**
	CB	0.98	0.99	0.98	0.91	**0.94**
	KNN	0.97	0.98	0.97	0.87	0.92
	AB	0.95	0.97	0.97	0.88	0.92
	GNB	0.95	0.96	**0.98**	0.80	0.88
30-sec Cleaned Labels	RF	**0.99**	**0.99**	0.98	**0.93**	**0.95**
	CB	0.98	0.98	0.98	0.91	0.94
	KNN	0.97	0.97	**0.98**	0.85	0.91
	AB	0.96	0.96	0.98	0.87	0.92
	GNB	0.96	0.95	0.97	0.81	0.88
5-sec Cleaned Labels	RF	**0.99**	**0.99**	0.98	**0.95**	**0.96**
	CB	0.99	0.99	**0.99**	0.94	0.96
	AB	0.98	0.98	0.98	0.92	0.95
	KNN	0.98	0.98	0.98	0.89	0.94
	GNB	0.98	0.97	0.98	0.89	0.93
RF: Random Forest, CB: Catboost, KNN: K-Nearest Neighbor, AB: AdaBoost, GNB: Gaussian Naive Bayes						

A. Impact of the Quality of Labels

We can observe from Table III that the accuracy obtained by using Cleaned Labels is significantly better than the accuracy obtained using the Raw Labels in the event log. For example, on the same window size equal to 1-min, the use of Raw Labels results in 0.92 AUC while the use of Cleaned Labels results in 0.98 AUC.

We perform a more detailed comparison between two cases: 1-min Raw Labels, and 1-min Cleaned Labels to check how the quality of labels affects the performance. We ran another set of experiments using an 80/20 split of the dataset, where 80% of the instances are used for training and 20% are used for testing. This experiment is used to generate confusion matrices [22] (Fig. 6) and ROC curves (Fig. 7) for 1-min Raw Labels and 1-min Cleaned Labels. Confusion matrices are included only for RF that is the best performing model overall as presented in Table III.

We can observe from Fig. 7 that there is a more significant drop in Recall as compared to Precision in the case of Raw Labels. As observed from Fig. 6, the model is more likely to misclassify normal operation as an event (false positives), than the other way around (false negatives). This is a desirable outcome for the detection where it is important to detect as many events as possible and picking up some mislabeled normal operation is a smaller issue because this can be eliminated during classification. The classification into different types of events is outside of scope of this paper and is left for future work.

B. Impact of Time Window Selection

Next, we analyze how the accuracy is affected by the change in the time window size. When we compare two cases

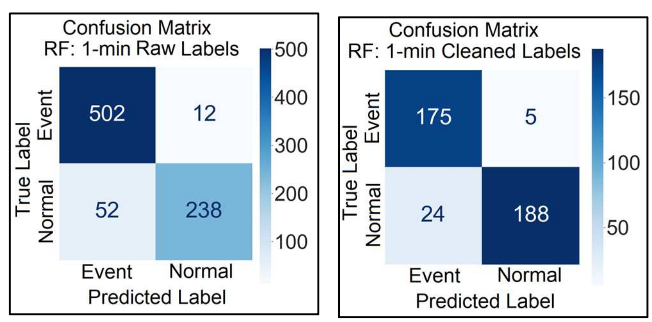

Fig. 6 Confusion matrix for Random Forest for 1-min Raw Labels (left) and 1-min Cleaned Labels (right)

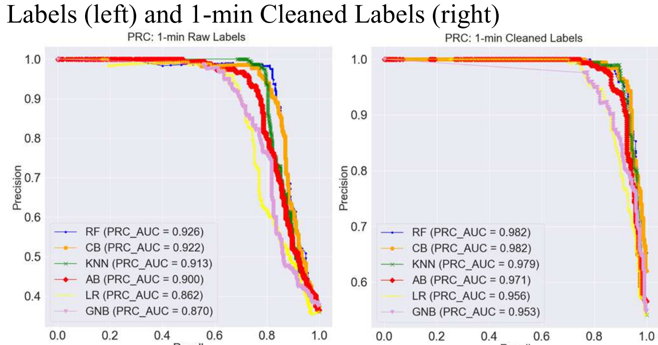

Fig. 7 Precision Recall Curve for 1-min Raw Labels (left) and 1-min Cleaned Labels (right)

with Raw Labels taken from the provided event log, we can conclude that the 3-min window results in better accuracy as compared to the 1-min window. This can be explained by the number of recorded events that are inside the 3 min window around the event log start time, but still outside of 1-min window. As demonstrated in Fig. 4 in Sec. III B there were at least 52 events (out of the inspected 1,038) that are outside of a 1-min time window. In case of these events, the detection model using 1-min labels is receiving a "normal operation" signal with an "event" label. By cross validation of the results from Table III we can observe that the AUPRC is very comparable between two cases (1-min and 3-min for Raw Labels).

Cleaned Labels demonstrate consistency in improving accuracy with shorter time windows. The visual inspection is performed to make sure that the start of the event is within the selected time window. The increase in accuracy from 0.981 in case of 1-min labels to 0.992 in case of 5-sec labels can be explained by the smaller fluctuation of normal operation within a shorter time window.

Conclusion

We propose a novel waveform feature extraction method that makes the ML models more efficient when detecting various types of events in large datasets of field-recorded PMU measurements. Since extracting events from such large data sets is time consuming and prone to errors due to imprecise labels, we observe several benefits of our method:

- A single Rectangle Area feature used as an input to the ML algorithms makes the detection of events in a large historical PMU dataset computationally very efficient.
- No data cleansing is required prior to feature extraction and only minor cleansing of out-of-range data is required after feature extraction to achieve ML performance.
- The Random Forest ML algorithm combined with the RA feature provides the best accuracy (0.924-0.991) depending on the window size and quality of labels.
- A reasonably high accuracy of detecting events with Raw Labels (0.931) can be significantly improved by learning from the Cleaned Labels (0.991).
- If the method to create labels is more precise than what the SCADA event log provides, the RA feature makes event detection using ML faster.

Disclaimer

References

[1] North American SynchroPhasor Initiative, "Data Mining Techniques and Tools for Synchrophasor Data," NASPI White Paper, NASPI-2018-TT-007, January 2019.

[2] M. Biswal, et al., "Signal features for classification of power system disturbances using PMU data," Proc. Power Syst. Comput. Conf., Jun. 2016, pp. 1–7.

[3] S. Brahma, et al., "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.

[4] D.-I. Kim, et al., "Wavelet-Based Event Detection Method Using PMU Data," IEEE Trans. Smart Grid, vol. 8, no. 3, May. 2017, pp. 1154-1162.

[5] M. Cui, et al., "A Novel Event Detection Method Using PMU Data With High Precision," IEEE Trans. Power Systems, vol. 34, no. 1, Jan. 2019, pp. 454-466.

[6] A. Shahsavari, et al., "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.

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

[8] R. Yadav, et al., "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] O. P. Dahal, et al., "Comprehensive Clustering of Disturbance Events Recorded by Phasor Measurement Units," IEEE Trans. Power Delivery, vol. 29, no. 3, pp. 1390-1397, Jun. 2014.

[10] L. Xie, et al., "Dimensionality Reduction of Synchrophasor Data for Early Event Detection: Linearized Analysis," IEEE Trans. Power Systems, vol. 29, no. 6, pp. 2784-2794, Nov. 2014.

[11] M. Rafferty, et al., "Real-Time Multiple Event Detection and Classification Using Moving Window PCA," IEEE Trans. Smart Grid, vol. 7, no. 5, pp. 2537-2548, Sep 2016.

[12] M. Khan, et al., "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.

[13] M. Biswal, et al., "Supervisory Protection and Automated Event Diagnosis Using PMU Data," IEEE Trans. Power Delivery, vol. 31, no. 4, pp. 1855-1863, Aug. 2016.

[14] M. Pavlovski, et al., "Hierarchical Convolutional Neural Networks for Event Classification on PMU Measurements," in IEEE Trans. Instrumentation and Measurement, vol. 70, pp. 1-13, no. 2514813., 2021.

[15] A. A. Hai et al., "Transfer Learning for Event Detection from PMU Measurements with Scarce Labels," in IEEE Access, vol. 9, pp. 127420-127432, 2021.

[16] R. Ma, et al., "A PMU-Based Data-Driven Approach for Classifying Power System Events Considering Cyberattacks," IEEE Systems Journal, Vol. 14, No. 3, pp. 3558-3569, Sep. 2020.

[17] J. Shi, et al., "Power System Event Identification Based on Deep Neural Network with Information Loading," in IEEE Trans. Power Systems, vol. 36, no. 6, pp. 5622-5632, Nov. 2021.

[18] C. Huang, et al., "Data quality issues for synchrophasor applications Part I: a review," J. Mod. Power Syst. Clean Energy 4, p. 342–352, 2016.

[19] 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.

[20] Scikit-learn, [Online] Available: https://scikit-learn.org/stable/

[21] L. Prokhorenkova, et al., "CatBoost: Unbiased Boosting with Categorical Features," in Advances in Neural Information Processing Systems 31, 2018.

[22] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation," Journal of Machine Learning Technologies, vol. 2, 2008.