Situational Awareness in Distribution Grid Using Micro-PMU Data: A Machine Learning Approach

Alireza Shahsavari, *Student Member, IEEE*, Mohammad Farajollahi, *Student Member, IEEE*,
Emma M. Stewart, *Senior Member, IEEE*, Ed Cortez, and Hamed Mohsenian-Rad, *Senior Member, IEEE*

***Abstract*—The recent development of distribution-level phasor measurement units, also known as micro-PMUs, has been an important step toward achieving situational awareness in power distribution networks. The challenge however is to transform the large amount of data that is generated by micro-PMUs to actionable information and then match the information to use cases with practical value to system operators. This open problem is addressed in this paper. First, we introduce a novel *data-driven event detection* technique to extract events from the extremely large collection of raw micro-PMU data. Subsequently, a *data-driven event classifier* is developed to effectively classify power quality events. Importantly, we use field expert knowledge and utility records to conduct an extensive *data-driven event labeling*. Moreover, certain aspects from event detection analysis are adopted as additional features to be fed into the classifier model. In this regard, a multi-class support vector machine classifier is trained and tested over 15 days of real-world data from two micro-PMUs on a distribution feeder in Riverside, CA, USA. In total, we analyze 1.2 billion measurement points and 10,700 events. The effectiveness of the developed event classifier is compared with prevalent multi-class classification methods, including *k*-nearest neighbor method as well as decision-tree method. Importantly, two real-world use-cases are presented for the proposed data analytics tools, including remote asset monitoring and distribution-level oscillation analysis.**

***Index Terms*—Machine learning, distribution synchrophasors, situational awareness, event detection, event classification, big-data.**

I. Introduction

THE PROLIFERATION in distributed energy resources, electric vehicles, and controllable loads has introduced new and unpredictable sources of disturbance in distribution networks. This calls for developing new monitoring systems that can support achieving situational awareness at distribution-level; thus, allowing the distribution system operator to make the best operational decisions in response to such disturbances.

Traditionally, there have been three major challenges in achieving situational awareness in power distribution systems. First is the lack of high resolution measurements. Metering in distribution systems is often limited to supervisory control and data acquisition (SCADA) at substations with minutely reporting intervals. As for smart meters, their report measurements once every 15 minutes or hourly. Second is the lack of accurate and up-to-date models for most practical distribution circuits. Third, due to the lower voltage and the larger number and variety of utility and customer equipment, distribution systems are subject to a huge number of events on a daily basis.

The first challenge above has recently been resolved by the advent of micro-PMUs [1]. A typical micro-PMU is connected to single- or three-phase distribution circuits to measure GPS time-referenced magnitudes and phase angles of voltage and current phasors at 120 readings per second. Since 2015, several micro-PMUs have been installed at pilot test sites in the state of California, including some in the city of Riverside [2].

This paper makes use of real-world micro-PMU data from a feeder in Riverside, CA, see Fig. 1. It seeks to address the second and the third challenges listed above. Specifically, we propose a novel *model-free* situational awareness framework for power distribution systems to turn micro-PMU data in to actionable information for tangible use cases. This is done by introducing a novel *data-driven event detection* technique as well as a novel *data-driven event classification* technique. Event detection is applied to eight non-linearly dependent data streams for each micro-PMU, including voltage magnitude, current magnitude, active power, and reactive power. Event classification is done by extracting the inherent features of detected events, and by constructing an algorithm that can learn from and make predictions of various events. The main contributions in this paper can be summarized as follows.

1) A novel situational awareness framework is introduced for power distribution systems using micro-PMU data, that is model-free; it works by going through a sequence of *event detection*, *event classification*, and *event scrutinization* efforts to transform the large amount of measurement data from micro-PMUs to information that are useful for distribution system operators.
2) The approach in this paper makes use of field expert knowledge and utility records in order to conduct an

Manuscript received August 30, 2018; revised December 14, 2018; accepted January 24, 2019. Date of publication February 12, 2019; date of current version October 30, 2019. This work was supported in part by UCOP under Grant LFR-18-548175, in part by DoE under Grant EE 0008001, and in part by NASA MIRO under Grant NNX15AP99A. Paper no. TSG-01268-2018. *(Corresponding author: Hamed Mohsenian-Rad.)*

A. Shahsavari, M. Farajollahi, and H. Mohsenian-Rad are with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: hamed@ece.ucr.edu).

E. M. Stewart is with the Infrastructure Systems, Cyber and Physical Resilience, Lawrence Livermore National Laboratory, Livermore, CA 94550 USA.

E. Cortez is with Riverside Public Utilities, Riverside, CA 92501 USA.

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

Digital Object Identifier 10.1109/TSG.2019.2898676

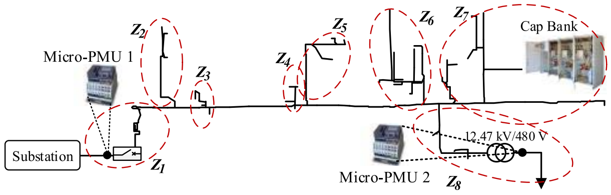


Fig. 1. The real-world distribution feeder that is studied in this paper.

extensive *data-driven event labeling* for micro-PMU data. The detected events are labeled according to *event zone* and *event type*. As for the event detection phase prior to event labeling, our approach is comprehensive; it involves *moving windows* to help compensate the lack of information about the *start time* of each event. It also involves *dynamic window sizes* to help compensate the lack of information about the *duration* of each event.

3) Different feature selection approaches and different classification methods are examined and compared, including multi-SVM, k-nearest neighbor, and decision-tree, with considering certain aspects of events from micro-PMUs, e.g., *uneven* datasets and features of *multi-stream* signals. It is shown that the use of the proposed detection features, such as detection window and detection indicator, is critical, regardless of the method of classification. It is also observed that multi-SVM is a better classifier compared to k-NN and DT in this particular application domain, whether or not we use the detection features.
4) Two important real-world use-cases are proposed and investigated, namely *remote asset monitoring* and *distribution-level oscillation analysis*. The first use-case allowed us to measure the internal phase imbalance in a 900 kVAR capacitor bank as well as a potential malfunction in its Volt/VAR controller. The second use-case also allowed us to identify the source location and the frequency of a class of oscillation events that occur on the understudy distribution feeder.

The early studies on micro-PMUs focused on innovative case studies, in [2]–[7]. In [8], a model-based event detection method is proposed to detect changes in the admittance matrix of the distribution grid using micro-PMUs data. However, model-based techniques are often prone to failure due to lack of model accuracy, particularly in case of detecting power quality events. Detecting partially-labeled events in micro-PMU data is proposed in [9]–[12]. However, given the complexity of distribution systems, it is difficult for utilities to pre-determine the variety of distribution-level events [13]. Thus, relying on expert knowledge and labeling events at event detection phase may inevitably result in overlooking some events. To resolve this issue, we propose a model-free event detection approach to capture *unlabeled* data.

In the context of event classification in power distribution systems, prior studies have classified the various causes of fault events [14]–[16], and power quality events [17], [18]. However, broadly speaking, the current literature is still limited when it comes to studying large-scale real-world micro-PMU data sets; therefore, the challenges that may arise in practical event classification problems are yet to be understood and addressed. Finally, in [19], classifiers are trained to identify malfunctioned capacitor bank switching and malfunctioned on-load tap-changer switching events using data from hardware-in-the-loop simulations. The transient signatures of these malfunctions are derived by simulating different test systems and test scenarios. In contrast, the labeling in this paper is done by using real-world data combined with field knowledge from utility staff and utility event logs.

II. Data-Driven Event Detection

Let $D_i := [d_1, \ldots, d_n]^T$ denote a sequence of measurements from a micro-PMU, such as current magnitude on one phase, where n is the number of observation samples in the sequence. Subscript i is the index of the data sequence within the overall micro-PMU data stream. We define MAD_i as the median absolute deviation (MAD) in data sequence D_i as follows:

$$MAD_i = \gamma \cdot M[|D_i - M[D_i]|], \tag{1}$$

where $M[\,\cdot\,]$ and $|\cdot|$ denote median and absolute values. A typical value for coefficient γ is 1.4826 [20]. In this study, we detect an event within data sequence D_i if there exists a data point $k = 1, \ldots, n$ for which any of the following holds:

$$\begin{aligned} d_k &\leq M[D_i] - \zeta^- MAD_i \\ M[D_i] + \zeta^+ MAD_i &\leq d_k, \end{aligned} \tag{2}$$

where ζ^- and ζ^+ denote the threshold to detect overshoot and undershoot in the data sequence, respectively. Here, $M[D_i] - \xi^- MAD_i$ and $M[D_i] + \xi^+ MAD_i$ denote the lower-bound and the upper-bound margins for data sequence D_i, respectively. We define an indicator function $\mathbb{I}\{\cdot\}$ such that $\mathbb{I}\{D_i\} = 1$, if the condition in (2) holds for data sequence D_i; and $\mathbb{I}\{D_i\} = 0$ otherwise. Note that, the above approach to detect an event in a micro-PMU data sequence is a statistical anomaly detection technique which uses the absolute deviation around median test. Other statistical anomaly detection methods could also be used, such as the extreme studentized deviate test, or the standard deviation around mean value test, see [20].

The choice of parameters ζ^- and ζ^+ and the size of the data sequence window n have impact on the performance of the detection method. While ζ^- and ζ^+ are often selected empirically, choosing the right window size n is very challenging. In fact, we observed that it may *not* be possible to detect all events based on only one value for parameter n.

In order to overcome the above challenges, we propose to use a *dynamic window size* as well as a *moving window* such that we can detect as many events as possible. On one hand, the *dynamic window size* can help to compensate the lack of information about the *duration of each event*. On the other hand, the *moving window* can help to compensate the lack of information about the *start time of each event*.

The impact of applying dynamic window sizes and moving windows is shown in Fig. 2. Here, the entire data stream takes 100 seconds. Three major events can be visually detected, with start time stamps t_1, t_2, and t_3. We can see that different events have different natures and different lengths. The first event is

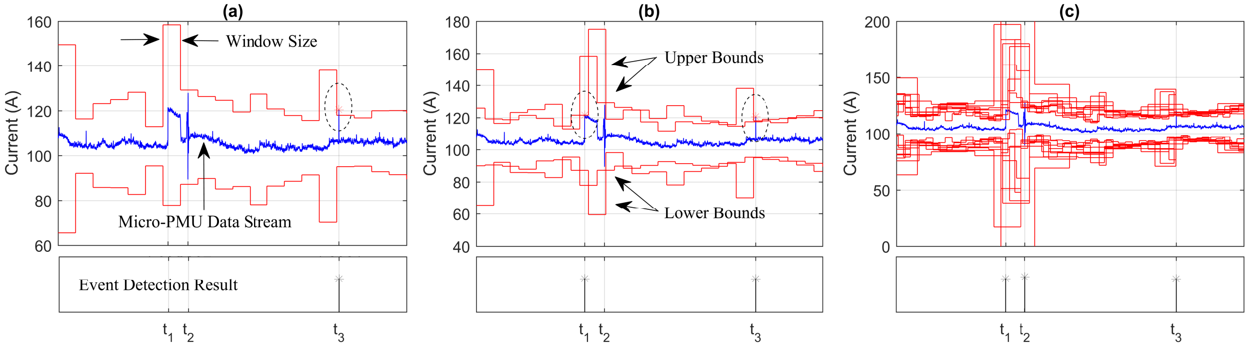


Fig. 2. Effect of *moving window* and *dynamic window size* on event detection: (a) static window size without moving window: one event is detected at t_3; (b) static window size with moving window: another event is detected at t_1; (c) dynamic window size with moving window: all three events are detected.

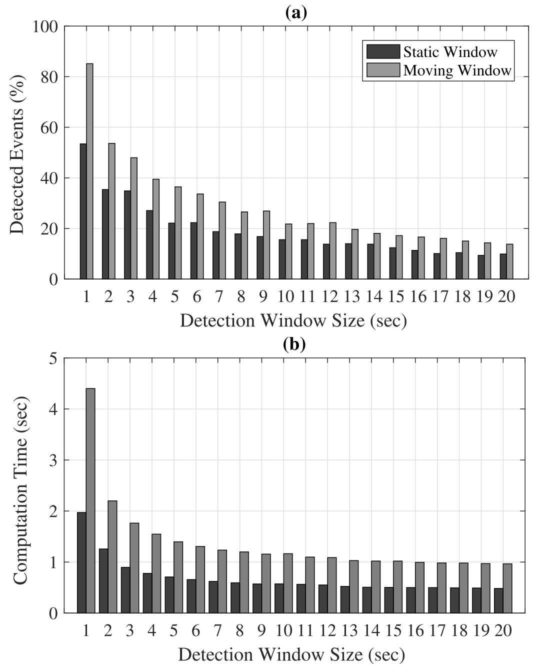


Fig. 3. (a) The percentage of detected events. (b) Computation time.

long lasting. It can be detected either at its step-up edge or at its step-down edge; or both. The second event includes some transient oscillations. The third event is a momentary spike.

Fig. 2(a) shows the case where the window size is fixed at $n = 600$ micro-PMU samples, i.e., five seconds. Therefore, in the 100 seconds of data shown in this figure, there exist $20 = 100/5$ upper-bound and lower-bound margins of the form in (2). Only the third event at time t_3 is detected in this case. Fig. 2(b) shows the case where there is a second window of the same size, a moving window, that is shifted by 300 samples, i.e., half of the window size. Therefore, besides those 20 upper-bound and lower-bound margins that we saw in Fig. 2(a), there are additional 20 upper-bound and lower-bound margins in this figure. Accordingly, for each micro-PMU sample, there exist two upper-bound margins and two lower-bound margins in Fig. 2(b). The new upper-bound and lower-bound margins in Fig. 2(b) can detect the first event at time t_1. Fig. 2(c) shows the windows of $n = 120, 360, 600, 840, 1080$ samples for their sizes, where each window is also moved by half of its own size. The window sizes are 1, 3, 5, 7, and 9 seconds, respectively. Note that, for each micro-PMU sample in this figure, there exist 10 upper-bound margins and 10 lower-bound margins. As it can be seen from this figure, by applying both moving windows and dynamic window sizes, we can detect *all three* events.

Both dynamic window sizes and moving windows are necessary to assure detecting *all* events. This point is illustrated in Fig. 3(a). The percentage of correctly identified events versus the window size are shown in this figure; for both static and moving window types. The micro-PMU data stream in this example takes one day and includes 564 events. We can see that the use of moving window is always more effective than the use of a static window. No single window size can detect all events. However, collectively, a combination of different window sizes and moving windows can detect all 564 events. Also, Fig. 3(b) shows the computation time corresponding to static windows and moving windows for different window sizes. As it can be seen in this figure, in each window size, the computation time of the moving windows is twice of the static window. Also, as expected, the computation time decreases as the detection method is applied to wider window sizes.

III. Event Labeling and Feature Selection

Given the events that are captured by using the event detection method in Section II, in this section, we conduct a comprehensive event labeling and feature selection approach.

A. Two-Layered Event Labeling

One can label power system events based on different aspects of their characteristics. Here, we seek to label the events according to *event zone* and *event type*.

1) Layer 1 Labeling Based on Event Zone: In this first layer of classification, each event can take one of the below labels:

- Class I. Events initiated from upstream of micro-PMU 1, i.e., at transmission-level or another distribution feeder;
- Class II. Events initiated from downstream of micro-PMU 2, i.e., at customer location that hosting micro-PMU 2;
- Class III. Events initiated from somewhere between the two micro-PMUs across the distribution feeder of interest.

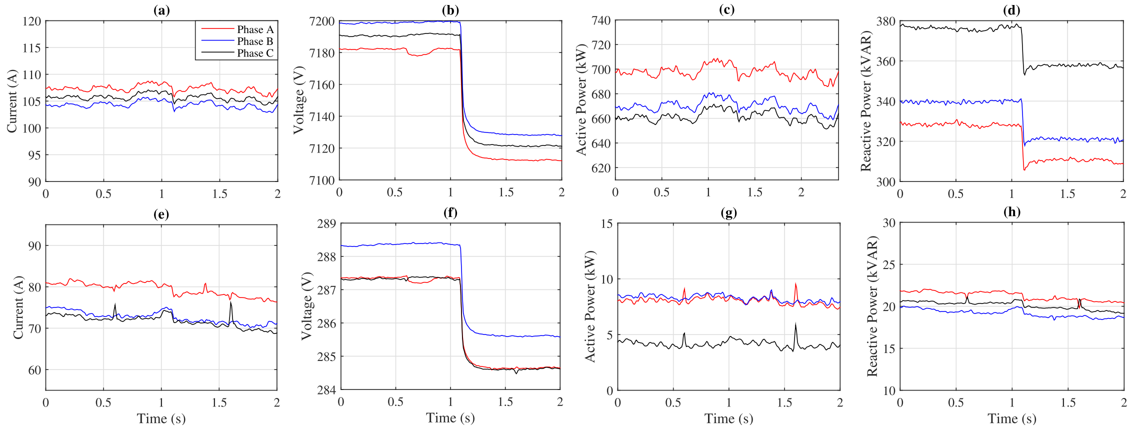


Fig. 4. An example Class I event: (a) and (e): current, (b) and (f): voltage, (c) and (g): active power, (d) and (h): reactive power. First row corresponds to the measurements from micro-PMU 1. Second row corresponds to the measurements from micro-PMU 2.

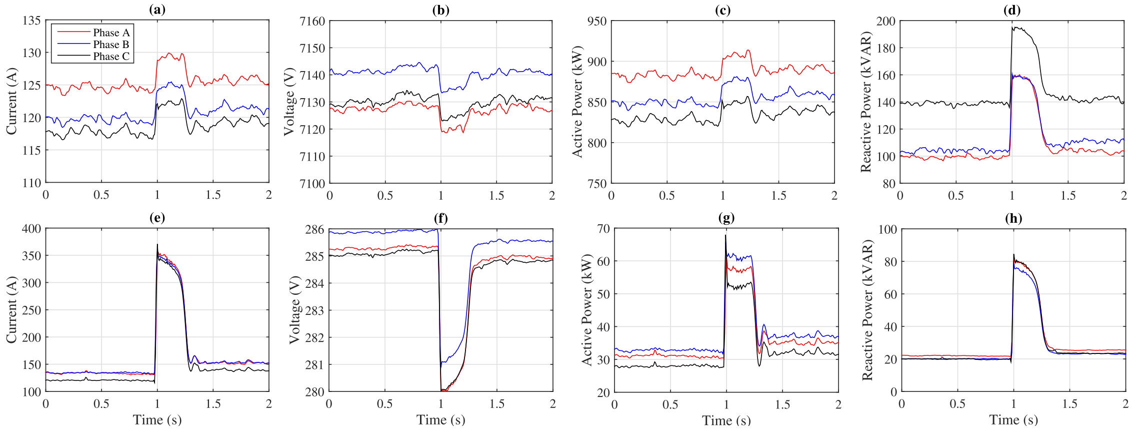


Fig. 5. An example Class II event: (a)-(h) are defined the same way as in Fig. 4.

An example for a Class I event is shown in Fig. 4. Class I events often appear as sustained steps or temporary fluctuations in voltage magnitudes at both feeder-level, seen by micro-PMU 1, and customer-level, seen by micro-PMU 2. However, they do *not* cause any major change in the current magnitudes. Class I events could be due to transformer, capacitor bank, generator, or load switching at sub-transmission or transmission networks. They could also be due to momentary faults on another neighbouring distribution feeder, see [2].

An example for a Class II event is shown in Fig. 5. Class II events often appear as sustained steps or temporary fluctuations in voltage magnitude, current magnitude, active power, and reactive power at customer level, seen by micro-PMU 2. Depending on the size of the event, the event signature is noticeable also in the measurements at the feeder level, seen by micro-PMU 1. Class II events could be due to load switching, such as motor and HVAC loads, DER switching, such as PVs and batteries, among other customer-level causes [3].

An example for a Class III event is shown in Fig. 6. Class III events often appear as sustained steps or temporary fluctuations in voltage magnitude, current magnitude, active power, and reactive power at feeder-level, seen by micro-PMU 1. They may also affect the voltage magnitude at customer-level, seen by micro-PMU 2. However, they do *not* have a major impact on current magnitude, active power, and reactive power at customer-level. Class III events can be due to a wide verity of causes, such as distribution-level transformer and capacitor bank switching, lateral fuse blowing, primary protection operation, load switching, DER switching, etc.

2) Layer 2 Labeling Based on Event Type: Each event can be further labeled based on its type. This can be done for all events, whether they are in Classes I, II, or III. However, labeling the type of Class I events is not of interest; because distribution-level PMUs are not intended to investigate transmission-level events. Labeling the type of Class II events is not of great interest either; because the customer that hosts micro-PMU 2 is being monitored directly. In fact, it is only Class III events that are of interest to be further classified; because those are the events that occur across the distribution feeder. Thus, in this section, we define a second layer for labeling Class III events, as follows.

- Class III.A. Capacitor bank switching; see Fig. 6;
- Class III.B. Distribution-level oscillation, see Fig. 7;
- Class III.C. Other events, see Fig. 8.

The above distinction is based on the fact that capacitor bank switching is an important event in distribution systems and the subject of several studies [2], [21]. Oscillation events too are important. There is currently a limited understanding of the oscillation events within distribution systems [1].

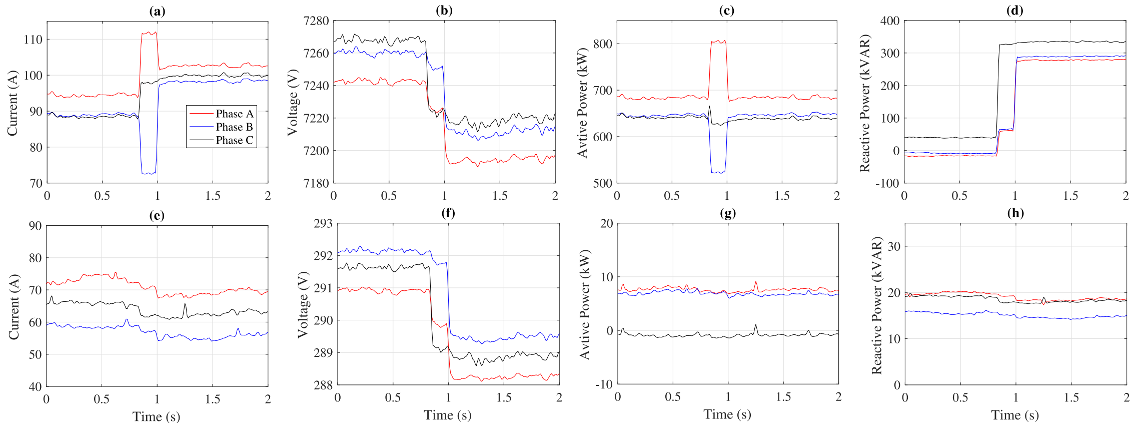


Fig. 6. An example Class III event, i.e., Class III.A: (a)-(h) are defined the same way as in Fig. 4.

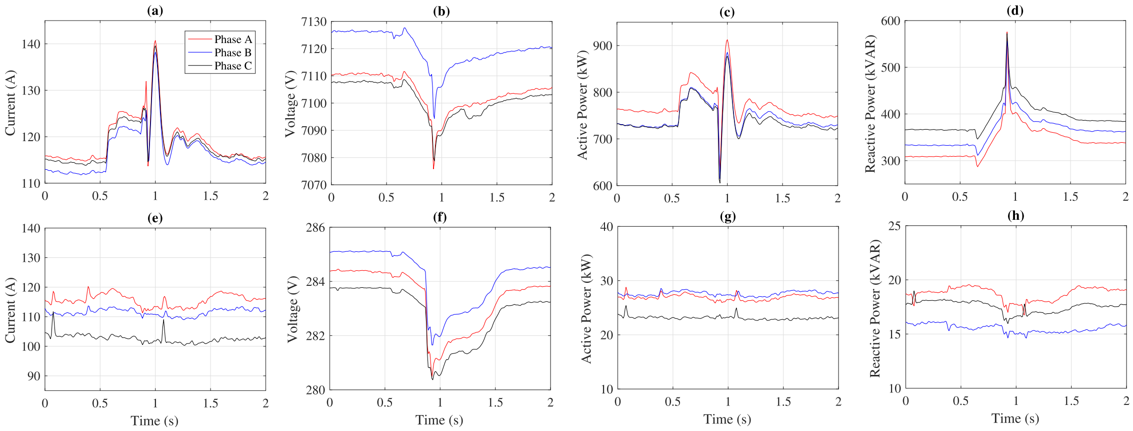


Fig. 7. An example Class III event, i.e., Class III.B: (a)-(h) are defined the same way as in Fig. 4.

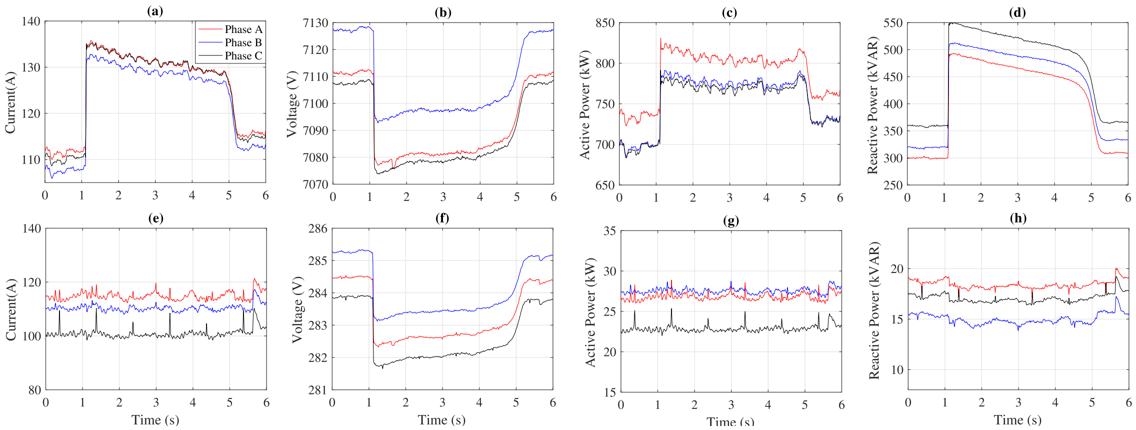


Fig. 8. An example Class III event, i.e., Class III.C: (a)-(h) are defined the same way as in Fig. 4.

B. Feature Selection

A crucial task in any classification problem that involves Machine Learning is to choose adequate quantifiable features that can help distinguish classes. In this paper, we propose the features in Table I, which consists of three broad categories.

- *Single-Stream Features:* These are quantifiable properties that are derived from single data streams $D_i \in \{I, V, P, Q\}$. They could be obtained by applying the mean, standard deviation, median, difference, or other operators to each of such single data streams within the detected window. In this study, we use standard deviation and absolute difference. Note that, notations d_1 and d_n denote the first and the last data samples in D_i.
- *Multi-Stream Features:* These are quantities that are defined for various combinations of two synchronized data sequences $D_i, D_j \in \{I, V, P, Q\}$, whether from the same micro-PMU or two micro-PMUs. Different operators could be applied to the data streams. Here, we use

TABLE I
PROPOSED FEATURES FOR CLASSIFICATION

Feature	Feature Description		Number
Single-stream	Statistics	$\mathrm{std}(D_i)$	8
	Difference	$\lvert d_n - d_1 \rvert$	8
Multi-stream	Correlation	$corr(D_i, D_j)$	28
Detection	Detection Window	ω	1
	Detection Indicator	$\mathbb{I}\{D_i\}$	8

the correlation between any two of the eight available data sequences to construct the multi-stream features.

- *Detection Features:* The way that an event is detected can itself carry useful information to classify the event. We use the following detection features for classification: 1) the smallest window at which the event was detected, denoted by ω; and 2) the binary detection indicators $\mathbb{I}\{D_i\}$ for $D_i \in \{I, V, P, Q\}$, for both micro-PMUs.

IV. DATA-DRIVEN EVENT CLASSIFICATION

We now use the event labeling and feature selection strategies in Section III to train different types of event classifiers.

A. Binary-SVM Classifier

Consider m events that are detected by using the method in Section II. We use these events to train an SVM classifier. For each training event $i = 1, \ldots, m$, let X_i denote the 53×1 vector of extracted features, where $53 = 28 + 3 \times 8 + 1$, as in Table I. Also, let y_i denote the assigned label for event i.

When it comes to binary classification, there are only two types of labels. We define $y_i \in \{-1, 1\}$, where $y_i = -1$ is the label for the first class; and $y_i = 1$ is the label for the second class. Let $W^T X + b = 0$ denote a separating hyperplane in the 53×53 feature space that separates the two classes, where W is a 53×1 coefficients vector, and b is the intercept. The SVM training problem seeks to find the optimal hyperplane that has the maximum total distance between the two classes across the training samples. If the training samples are not linearly separable, we should add some slack variables so as to turn the SVM into a *soft margin SVM*, which is formulated as:

$$\underset{W,b,\xi}{\text{minimize}} \quad \frac{1}{2}\|W\|_2^2 + \lambda \sum_{i=1}^{n} \xi_i \tag{3a}$$

$$\text{subject to} \quad y_i\big(W^T X_i + b\big) \geq 1 - \xi_i, \quad i = 1, \ldots, m \tag{3b}$$

$$\xi_i \geq 0, \; i = 1, \ldots, m, \tag{3c}$$

where ξ_i is a slack variable corresponding to training event i. If $y_i = -1$, then constraint (3b) requires that $W^T X_i + b \leq -1 + \xi_i$; and if $y_i = 1$, then constraint (3b) requires that $W^T X_i + b \geq 1 - \xi_i$; thus, making $W^T X + b = 0$ a separating hyperplane with a soft SVM margin of length $1 - \xi_i$ on both sides. Parameter λ is a tuning parameter. If the extracted features of a training event results in a point that falls on the correct side of the separating hyperplane with respect to the label of the event, then $0 \leq \xi_i < 1$; otherwise $\xi_i \geq 1$, see [22], [23].

B. Multi-SVM Classifier

A multi-class classification problem can be decomposed into several binary classification problems. This can be done by using methods such as one-against-one (OvO), one-against-all (OvA), directed acyclic graph SVM (DAGSVM), and binary tree of SVM (BTSVM) [22], [23]. In this paper, we use OvA decomposition. We construct c binary SVM problems, where c is the number of classes. Each binary SVM problem obtains a separating hyperplane to separate one of the c classes from the rest of the $c - 1$ classes. We have $c = 3$ for both Layer 1 and Layer 2 classification; see Section IV-A. In total, six sets of separating hyperplanes are trained; three sets for each layer. Once the training process is complete, the decision on class prediction for testing event i is made as [22]:

$$y_i = \underset{l=1,\ldots,c}{\arg\max}\big(W_l^T X_i + b_l\big). \tag{4}$$

In (4), we say that event i is predicted to belong in class l, which has the largest value of the decision function.

An alternative training separating hyperplanes in the form of $W_l^T X + b_l$ is to use non-linear classifiers, such as separating quadratic planes [24]. However, our experimental results based on real-world data have shown that there is no advantage in using nonlinear classifiers which are computationally more complex. What matters the most is to choose the right classification features, i.e., as in Table I, as we will further discuss in Section V.

C. K-NN and Decision-Tree Classifiers

There are other classifiers that one can consider for this study. One example is the *k-nearest neighbors* (k-NN) classifier [25]. Another example is the *decision-tree* (DT) classifier [26]. The k-NN method classifies an unknown sample based on the known labels of its k-closet, e.g., in the Euclidean sense, neighbors [25]. As for the DT classifier, a decision tree is constructed by creating branches as conjunctions of features as well as leaves as class labels. Then, a test data sample is classified based on branches conjunctions [26]. These additional methods are not discussed here in details due to space limitation. However, detailed performance comparisons are provided across these methods later in Section V.

D. Metrics to Compare Different Classifiers

First, consider the binary classifiers as in Section IV-B. The correctness of each classifier can be evaluated by computing the following four quantities: *True Positive (TP)*, which is the number of events that are correctly classified to be inside of the target class; *True Negative (TN)*, which is the number of events that are correctly classified to be outside of the target class; *False Positive (FP)*, which is the number of events that are incorrectly classified to be inside of the target class; and *False Negative (FN)*, which is the number of events that are incorrectly classified to be outside of the target class. Accordingly, for each binary classifier, we can calculate the following five standard performance evaluation metrics [27]:

$$TPR = \frac{TP}{TP + FN}, \tag{5}$$

$$FPR = \frac{FP}{FP + TN}, \tag{6}$$

$$PPV = \frac{TP}{TP + FP}, \tag{7}$$

$$FOR = \frac{FN}{FN + TN}, \tag{8}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{9}$$

where TPR, FPR, PPV, FOR, and MCC stand for the true positive rate, false positive rate, positive predictive values, false omission rate, and Matthews correlation coefficient, respectively. It should be noted that, in some machine learning literature, such as in [27], TPR, PPV, and FPR are also known as *recall*, *precision*, and *fall-out* metrices, respectively. The overall accuracy of a binary classifier can be assessed also by using the following metric:

$$F_1\text{Score} = \left(\frac{TPR^{-1} + PPV^{-1}}{2}\right)^{-1}. \tag{10}$$

Next, consider the multi-class classifier, as in Section IV-C. The recall, precision, and F_1*score* for multi-class classifier can be calculated by using either *Macro-averaging* or *Micro-averaging* [27]. Macro-averaging simply normalizes the sum of all metrics. Thus, Macro-averaging does not consider the number of events in each class. Micro-averaging however computes the metrics from sum of TP, TN, FP, and FN values of all classes. Thus, Micro-averaging takes the frequency of classes into consideration. Accordingly, one can prove that if Micro-averaging is used, then recall, precision, and F_1Score, all become equal, as follows:

$$\text{Recall} = \frac{\sum_{l=1}^{c} TP_l}{\sum_{l=1}^{c}(TP_l + FN_l)}. \tag{11}$$

V. Case Studies

The proposed event detection and event classification methods are applied to data from the two micro-PMUs in Fig. 1, during 15 days in July 2016. In total, we analyzed 1.2 billion measurement points, and 10,700 events. Only 1% of the measurements demonstrated any considerable event. Among the events detected, 1802, 2228, and 6670 events are labeled in Class I, Class II, and Class III, respectively. Among the 6670 Class III events, 27 events are labeled as Class A and 43 events are labeled as Class B. The training dataset includes 4.09% and 4.06% of all Layer I and Layer II data, respectively.

A. Classifier Design: A Preliminary Illustrative Example

In this section, a multi-SVM classifier is designed to separate the events in Layer I. Recall from Sections III and IV that the resulting three separating hyperplanes are in the 53 × 53 feature space that cannot be visualized. Therefore, in order to develop an example that is easy to illustrate, we use only the three most dominant features, out of 53, so as to visualize the separating hyperplanes in a 3×3 space, as shown in Fig. 9. In this figure, x_1 is the correlation coefficients between reactive power of the two micro-PMUs, x_2 is the standard deviation of the current magnitude of micro-PMU 1, and x_3 is the standard deviation of active power at micro-PMU 2. The total number of training events is 438. We can see that the events are properly separated across the three classes. For instance, Hyperplane I separates events corresponding to Class I from the rest of the events. The overall classifier training accuracy is 91%.

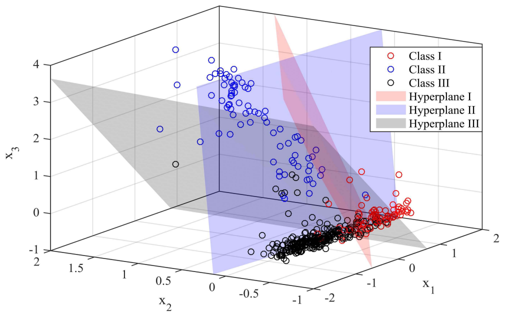


Fig. 9. Target classes and separating hyperplanes of Layer I in a 3×3 feature space. The circles indicate training data points.

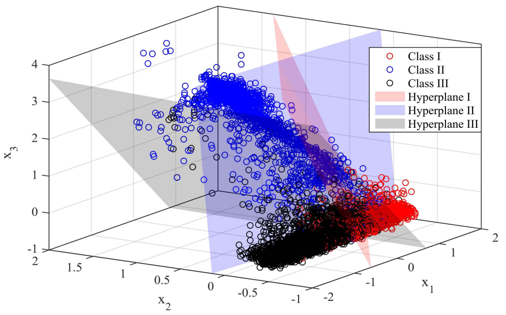


Fig. 10. Predicted classes of Layer I obtained by the use of the separating hyperplanes in Fig. 9. The circles indicate test data points.

Next, the above separating hyperplanes are applied to the test dataset, and the decision on class prediction is made using (4). Fig. 10 shows the predicted classes across 10262 test events. The overall classifier testing accuracy is 89%. Thus, several events in this preliminary example are *not* predicted in the right classes, mainly due to not using all 53 features.

B. Classification Results and Impact of Detection Features

In order to demonstrate the importance of features, in this section, we separately examine the following two cases:

- *Case 1:* Classification *without* detection features.
- *Case 2:* Classification *with* detection features.

Interestingly, the overall multi-SVM classifier training accuracy is 100% in both cases; not shown here. However, when it comes to using the classifiers to identify the classes for *test events*, the performance is considerably better for *Case 2* than *Case 1*. The confusion matrices for the multi-SVM, k-NN, and DT classifiers are shown in Fig. 11, for both Case 1 and Case 2. Parameter k for the k-NN classifier is set to 3 based on an exhaustive search. All results are based on the data for test events. Each confusion matrix shows the *recall* metric in percentage for binary-classifiers as well as overall classification *recall* using Micro-averaging.

Tables II to IV show the performance metrics in percentage for binary classifiers in Layer I. We can make two important observations from Fig. 11 and Tables II to IV. First, the performance is always better in Case 2 compared to Case 1. In other words, the use of detection features is indeed critical, regardless of the method of classification. Second, the multi-SVM classifier outperforms the k-NN classifier and the DT

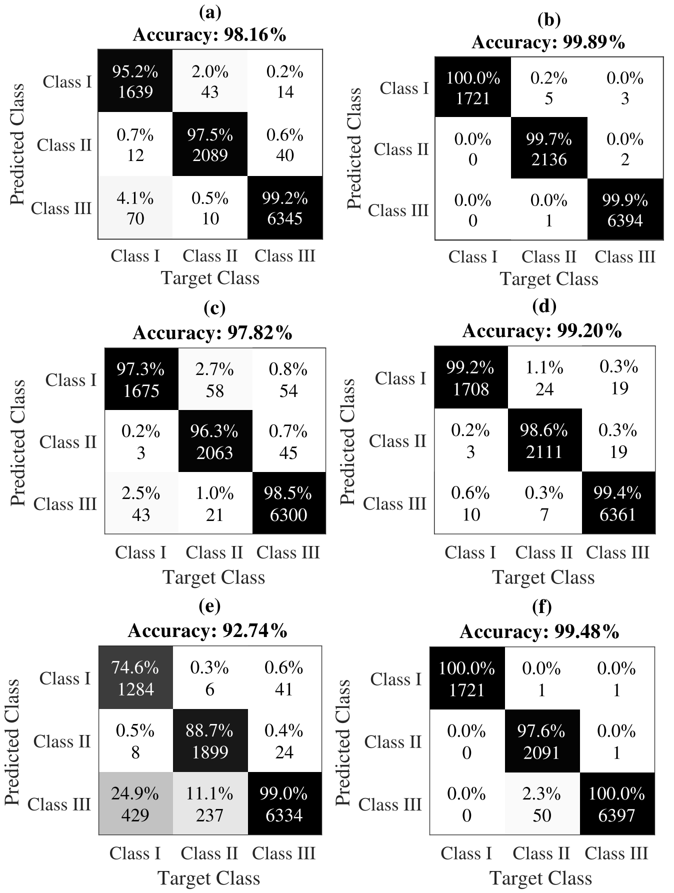


Fig. 11. Confusion matrix for test data of Layer I classification, i.e., with respect to event zone, obtained by various classifiers: (a) multi-SVM classifier, *Case 1*; (b) multi-SVM classifier, *Case 2*; (c) k-NN classifier, *Case 1*; (d) k-NN classifier, *Case 2*; (e) DT classifier, *Case 1*; (f) DT classifier, *Case 2*.

TABLE II
PERFORMANCE METRIC IN PERCENTAGE CORRESPONDING TO THE FIRST BINARY-CLASSIFIERS IN LAYER I CLASSIFICATION

Classifier	Case	TPR	FPR	PPV	FOR	MCC	F_1Score
SVM	1	95.23	0.66	96.63	0.95	95.12	95.93
	2	100	0.09	99.53	0	99.72	99.76
k-NN	1	97.23	1.31	93.73	0.54	94.59	95.49
	2	99.24	0.50	97.54	0.15	98.06	98.38
DT	1	74.60	0.55	96.46	4.89	52.35	84.14
	2	100	0.02	99.88	0	99.93	99.94

TABLE III
PERFORMANCE METRIC IN PERCENTAGE CORRESPONDING TO THE SECOND BINARY-CLASSIFIERS IN LAYER I CLASSIFICATION

Classifier	Case	TPR	FPR	PPV	FOR	MCC	F_1Score
SVM	1	97.52	0.64	97.57	0.65	96.90	97.54
	2	99.71	0.02	99.90	0.07	99.76	99.81
k-NN	1	96.31	0.59	97.72	0.96	96.23	97.01
	2	98.55	0.27	98.96	0.38	98.43	98.76
DT	1	88.65	0.39	98.34	2.91	91.77	93.24
	2	97.61	0.01	99.95	0.62	98.46	98.77

TABLE IV
PERFORMANCE METRIC IN PERCENTAGE CORRESPONDING TO THE THIRD BINARY-CLASSIFIERS IN LAYER I CLASSIFICATION

Classifier	Case	TPR	FPR	PPV	FOR	MCC	F_1Score
SVM	1	99.15	2.07	98.75	1.40	97.21	98.95
	2	99.92	0.02	99.98	0.12	99.87	99.95
k-NN	1	98.45	1.65	98.99	2.53	96.62	98.72
	2	99.40	0.44	99.73	0.97	98.86	99.56
DT	1	98.98	17.24	90.48	1.99	85.05	94.54
	2	99.96	1.29	99.22	0.05	98.92	99.59

TABLE V
PERFORMANCE METRIC IN PERCENTAGE CORRESPONDING TO THE FIRST BINARY-CLASSIFIERS IN LAYER II CLASSIFICATION

Classifier	TPR	FPR	PPV	FOR	MCC	F_1Score
SVM	100	0.11	78.12	0	88.33	87.71
k-NN	100	0.04	89.28	0	94.46	94.33
DT	0	0	N/A	0.39	N/A	N/A

classifier in both cases. In other words, the multi-SVM classifier is a better choice in this study, whether or not we use the detection features.

Based on the above results, for the rest of this paper, we always include the detection features, i.e., we use Case 2. It should be noted that the training accuracy of multi-SVM classifier is 100% in both cases. The results however are not shown here due to space limitation.

C. Classification Results for Second Layer

Unlike in Section V-A and V-B, where our focus was on Layer I classification, i.e., with respect to event zone, in this section, we examine the performance for Layer II classification, i.e., with respect to event type. The Layer II events that we observed in the real-world micro-PMU data demonstrated a very *uneven* distribution across different classes. Only 2 and 4 events out of 271 Class III training events belong to Class III.A and Class III.B, respectively. Also, only 0.4% and 0.6% of events in the test dataset are in Class III.A and Class III.B, respectively.

Figs. 12(a) and (b) show the confusion matrices corresponding to the training data and the test data for Layer II, respectively. The *recall* metric for the binary-classifiers and multi-class classifier are presented in confusion matrices. Similarly, Figs. 12(c) and (d) show the confusion matrices of the k-NN classifier and DT classifier for Layer II events. Due to space limitation, the confusion matrices are shown only for the test data. Parameter k for the k-NN classifier is set to 3.

The above results verify the performance of the proposed classifiers in separating uneven datasets. Also, Tables V to VII report the performance metrics corresponding to binary-classifiers of Layer II. From these results, we can conclude that the performance of the k-NN classifier is slightly better than the multi-SVM classifier in this particular case; although, there is a caveat about the k-NN classifier, which we will explain in the next paragraph. As for the DT classifier, we can see in the last rows of Tables V to VII that it fails to classify the events in Class III.A. Such events are incorrectly classified as Class III-B events.

We saw in the previous paragraph that the k-NN method can perform slightly better than the multi-SVM method. However, there is a catch, such performance is highly sensitive to the choice of parameter *k*. What was shown earlier was in fact the best possible result for the k-NN method. To see this, the impact of parameter *k* on the overall classification performance

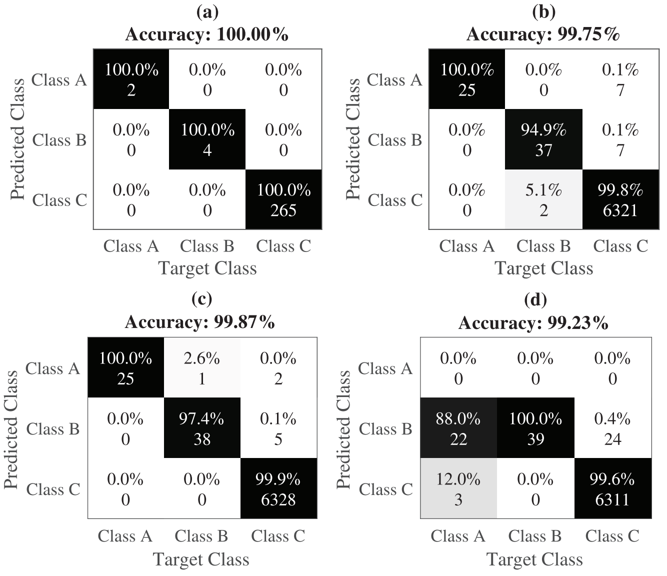


Fig. 12. Confusion matrix for Layer II classification, i.e., with respect to classifying event type: (a) multi-SVM classifier, training data; (b) multi-SVM classifier, test data; (c) k-NN classifier, test data; (d) DT classifier, test data.

TABLE VI
PERFORMANCE METRIC IN PERCENTAGE CORRESPONDING TO THE SECOND BINARY-CLASSIFIERS IN LAYER II CLASSIFICATION

Classifier	TPR	FPR	PPV	FOR	MCC	F_1Score
SVM	94.87	0.11	84.09	0.03	89.25	89.15
k-NN	97.43	0.07	88.37	0.01	92.74	92.68
DT	100	72.32	45.88	0	67.49	62.90

TABLE VII
PERFORMANCE METRIC IN PERCENTAGE CORRESPONDING TO THE THIRD BINARY-CLASSIFIERS IN LAYER II CLASSIFICATION

Classifier	TPR	FPR	PPV	FOR	MCC	F_1Score
SVM	99.77	3.12	99.96	18.42	88.77	99.87
k-NN	99.88	0	100	9.85	94.89	99.94
DT	99.62	4.68	99.95	28.23	82.51	99.78

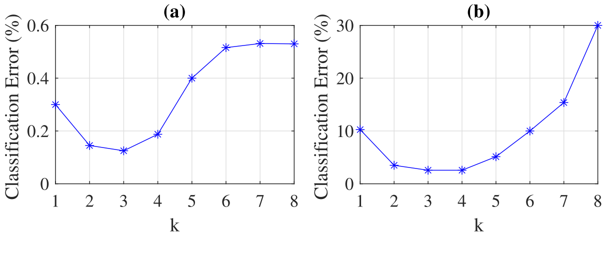


Fig. 13. Error for the k-NN versus parameter *k*, in classifying Layer II events: (a) overall classification, (b) classifying Class III.B.

as well as the performance in classifying Class III.B are shown in Fig. 13(a) and (b), respectively. As it can be seen for both cases, the minimum error is achieved by setting $k = 3$. The results are poor in other choices of *k*. In particular, the error in classifying Class III.B can be very high if parameter *k* is not carefully selected; thus, the results for the k-NN method are not as robust as those for the multi-SVM method.

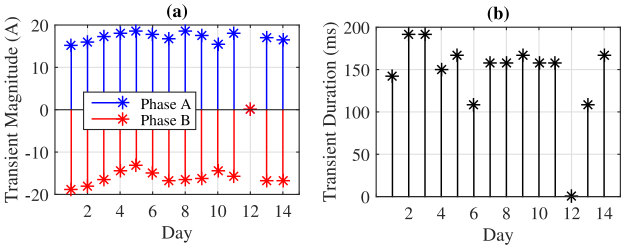


Fig. 14. Capacitor bank switch-off events for the first application in Section V-D: (a) transient current magnitude, (b) duration of transition.

D. First Application: Remote Asset Monitoring

In this section, we scrutinize the Class III.A events that are obtained by the method in Section V-C to achieve situational awareness by monitoring the capacitor bank. The cap bank in this study is rated 900 kVAR. It is switched by a vacuum circuit breaker (VCB). The VCB is controlled by a Volt-VAR controller, which switches on and off based on per-phase low-voltage and high-voltage override thresholds. The event classification method in Section V-C identified 25 cap bank switching events out of the total 10,700 events being examined.

Typically, for a wye-floating capacitor bank, the strategy for switching-off is to open contacts in two steps; first, opening one phase at zero-crossing of its current; second, opening the two other phases at quarter of a cycle later, at 90° relative to zero-crossing of the first phase [21]. Fig. 6(a) in Section III shows that there is an *overshoot* and an *undershoot* in current magnitude of phase A and phase B, respectively, while phase C experiences an ideal switching with no transient in current magnitude. Also, in Fig. 6(d), reactive power of phase C increases due to capacitor bank switching-off on phase C, while after about 200 msec dead-time, reactive power increases on phases A and B. Thanks to the event detection approach in Section II, this issue is further studied across 25 switching-off events that occurred across two weeks. The results are shown in Fig. 14. We can confidently conclude that the capacitor bank switching is ideal during its first step, but there is always about 20% overshoot and undershoot transients in current magnitude lasting for 100-200 msec during its second step.

The unbalanced or underrating operation of the capacitor bank was also investigated during the two weeks of this study. Fig. 15 shows the change in reactive power compensation across three phases due to switch-on events. Reactive power compensation is 288 kVAR on Phase A, 291 kVAR on Phase B, and 286 kVAR on Phase C. The results are similar during the switching on events are omitted due to space limitation.

E. Second Application: Event Source Location Identification

In this section, we scrutinize the Class III.B events that are obtained by our proposed method in Section V-C to characterize the oscillations on the under-study distribution feeder. We combine our event detection and classification results with the event source location identification (ESLI) method in [4] and [28] to pinpoint the location of each oscillation event. The

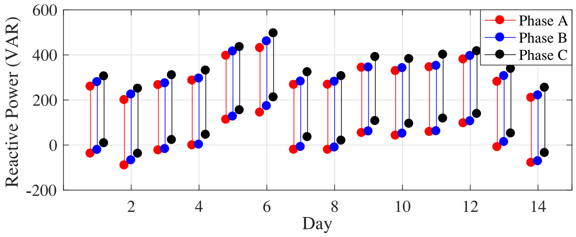


Fig. 15. Change in reactive power during capacitor bank switching.

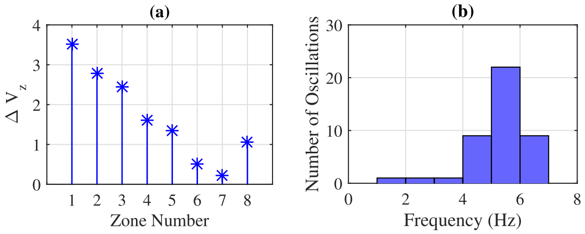


Fig. 16. Oscillation events: (a) zonal voltage discrepancy, (b) frequency.

granularity of location identification is necessarily limited to 8 zones, as marked on Fig. 1, due to the limited number of micro-PMU installations; see [4] for more details. Note that, Z_1 and Z_8 are the *upstream-level* and *customer-level* zones, i.e., they correspond to Class I and Class II, respectively.

In total, 43 oscillation events are identified during the two weeks of this study. For each individual oscillation event, ESLI calculates the so-called *zonal voltage discrepancy* ΔV_z for zones $z = 1, \ldots, 8$; see [4]. An example for applying ESLI to one of the oscillation events is shown in Fig. 16(a); where the location of the event is zone Z_7, because it has the smallest zonal voltage discrepancy. Interestingly, we identified that all the 43 oscillatory events occurred in zone Z_7. By applying the Fourier analysis, we obtained the frequency of all oscillations at this location. Their histogram is shown in Fig. 16(b).

VI. Conclusion

The goal of this paper was to start from a stream of raw micro-PMU data and turn them into information for tangible use cases for power distribution systems. First, a novel model-free event detection technique is proposed to pick out valuable portion of data from micro-PMU data streams from a real-world test site in Riverside, CA. In total, 10,700 events were detected and examined. Subsequently, a novel data-driven event labeling technique was combined with different methods of classification to classify the detected events at two layers. Interestingly, we concluded that adopting classification features from the detection process can considerably improve the overall classification accuracy. Finally, two real-world uses cases were investigated, namely for remote asset monitoring and distribution-level oscillation analysis. The results in this paper could be of value to utilities and system operators.

This study can be extended to active distribution networks with higher penetration of DERs. While we expect our approach to perform well with detecting the increased number of events that may occur due to the increased number DERs, it may not be easy to label all such new events due to the limited information about the resources and equipment that are owned and operated by customers.

References

[1] A. V. Meier, D. Culler, A. McEachern, and R. Arghandeh, "Micro-synchrophasors for distribution systems," in *Proc. IEEE ISGT*, Washington, DC, USA, Feb. 2014, pp. 1–5.

[2] H. Mohsenian-Rad, E. Stewart, and E. Cortez, "Distribution synchrophasors: Pairing big data with analytics to create actionable information," *IEEE Power Energy Mag.*, vol. 16, no. 3, pp. 26–34, May/Jun. 2018.

[3] A. Shahsavari *et al.*, "Distribution grid reliability versus regulation market efficiency: An analysis based on micro-PMU data," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2916–2925, Nov. 2017.

[4] M. Farajollahi, A. Shahsavari, E. 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.

[5] A. Shahsavari, M. Farajollahi, E. Stewart, A. Van-Meier, L. Alvarez, E. Cortez, and H. Mohsenian-Rad, "A data-driven analysis of capacitor bank operation at a distribution feeder using micro-PMU data," in *Proc. IEEE PES ISGT*, Washington, DC, USA, Apr. 2017.

[6] A. Shahsavari, M. Farajollahi, E. Stewart, C. Roberts, F. Megala, L. Alvarez, E. Cortez, and H. Mohsenian-Rad, "Autopsy on active distribution networks: A data-driven fault analysis using micro-PMU data," in *Proc. IEEE PES NAPS*, Morgantown, WV, USA, Sep. 2017.

[7] A. Shahsavari, M. Farajollahi, E. Stewart, C. Roberts, and H. Mohsenian-Rad, "A data-driven analysis of lightning-initiated contingencies at a distribution grid with a PV farm using micro-PMU data," in *Proc. IEEE PES NAPS*, Morgantown, WV, USA, Sep. 2017.

[8] O. Ardakanian *et al.*, "Event detection and localization in distribution grids with phasor measurement units," in *Proc. IEEE PES Gen. Meeting*, Chicago, IL, USA, Jul. 2017, pp. 1–5.

[9] 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.

[10] M. Jamei, A. Scaglione, C. Roberts, E. Stewart, S. Peisert, C. McParland, and A. McEachern, "Anomaly detection using optimally placed μPMU sensors in distribution grids," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 3611–3623, Jul. 2018.

[11] Y. Zhou, R. Arghandeh, H. Zou, and C. J. Spanos, "Nonparametric event detection in multiple time series for power distribution networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1619–1628, Feb. 2019.

[12] Y. Zhou, R. Arghandeh, I. Konstantakopoulos, S. Abdullah, A. Meier, and C. Spanos, "Abnormal event detection with high resolution micro-PMU data," in *Proc. IEEE Power Syst. Comput. Conf.*, Genoa, Italy, Jun. 2016.

[13] P. K. Ray, N. Kishor, and S. R. Mohanty, "Islanding and power quality disturbance detection in grid-connected hybrid power system using wavelet and S-transform," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1082–1094, Sep. 2012.

[14] L. Xu and M.-Y. Chow, "A classification approach for power distribution systems fault cause identification," *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 53–60, Feb. 2006.

[15] N. S. Coleman, C. Schegan, and K. N. Miu, "A study of power distribution system fault classification with machine learning techniques," in *Proc. IEEE PES NAPS*, Charlotte, NC, USA, Oct. 2015, pp. 1–6.

[16] J. Zhang, Z. He, S. Lin, Y. B. Zhang, and Q. Q. Qian, "An ANFIS-based fault classification approach in power distribution system," *Int. J. Elect. Power Energy Syst.*, vol. 49, pp. 243–252, Jul. 2013.

[17] T. Radil, P. M. Ramos, F. M. Janeiro, and A. C. Serra, "PQ monitoring system for real-time detection and classification of disturbances in a single-phase power system," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 8, pp. 1725–1733, Jun. 2008.

[18] S. Mishra, C. N. Bhende, and B. K. Panigrahi, "Detection and classification of power quality disturbances using S-transform and probabilistic neural network," *IEEE Trans. Power Del.*, vol. 23, no. 1, pp. 280–287, Jan. 2008.

[19] I. Niazazari and H. Livani, "A PMU-data-driven disruptive event classification in distribution systems," *Elect. Power Syst. Res.*, vol. 157, pp. 251–260, Apr. 2018.

[20] P. Huber, *Robust Statistics*. Hoboken, NJ, USA: Wiley, 2009, pp. 107–109.

[21] W. Sima, M. Zou, Q. Yang, M. Yang, and L. Li, "Field experiments on 10 kV switching shunt capacitor banks using ordinary and phase-controlled vacuum circuit breakers," *Energies*, vol. 9, no. 2, pp. 88–102, Jan. 2016.

[22] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[23] G. Madzarov, D. Gjorgjevikj, and I. Chorbev, "A multi-class SVM classifier utilizing binary decision tree," *Informatica*, vol. 33, no. 2, pp. 233–241, 2009.

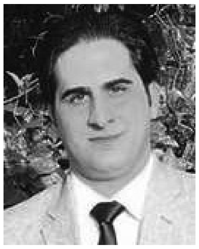
[24] S. Suthaharan, "Machine learning models and algorithms for big data classification," in *Integrated Series in Information Systems*, vol. 36. New York, NY, USA: Springer, 2016.

[25] N. García-Pedrajas, J. A. R. del Castillo, and G. Cerruela-García, "A proposal for local k values for k-nearest neighbor rule," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 470–475, Feb. 2017.

[26] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, and S. Mishra, "Decision tree and SVM-based data analytics for theft detection in smart grid," *IEEE Trans. Industrial Informatics*, vol. 12, no. 3, pp. 1005–1016, Jun. 2016.

[27] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, Jul. 2009.

[28] M. Farajollahi, A. Shahsavari, and H. Mohsenian-Rad, "Location identification of distribution network events using synchrophasor data," in *Proc. IEEE PES NAPS*, Morgantown, WV, USA, Sep. 2017.



Alireza Shahsavari (S'17) received the M.Sc. degree in electrical engineering – power systems from the University of Tehran, Tehran, Iran, in 2013. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of California at Riverside, CA, USA. His research interests include data analysis in smart grid and distribution system planning, operation, and monitoring. He was a recipient of the NASA MIRO FIELDS Fellowship in Big-Data Analysis from NASA JPL in 2017.

Mohammad Farajollahi (S'15) received the B.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2014, the M.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, in 2016. He is currently pursuing the Ph.D. degree with the University of California at Riverside, CA, USA. His research interests include power system planning, operation, reliability, as well as optimization. He is specifically working on applications of micro-PMUs and data analysis in distribution system monitoring.

Emma M. Stewart (M'08–SM'14) received the undergraduation degree in electrical and mechanical engineering from the University of Strathclyde, Glasgow, U.K., in 2004 and the Ph.D. degree in electrical engineering in 2009. She joined BEW Engineering in 2009, where she was a Senior Engineer with the Transmission and Distribution Team, and she led distribution studies for interconnection and high renewable penetration studies for large utilities. She joined Lawrence Berkeley National Laboratory in 2009, where she was the Deputy Group Leader for grid integration. She joined the Global Security Team with Lawrence Livermore National Laboratory, where she is the Deputy Associate Program Leader for Infrastructure Systems.

Ed Cortez received the B.Sc. degree in Electrical Engineering, California State Polytechnic University, Pomona, CA, USA. He is a Principal Electrical Engineer with the City of Riverside Public Utilities. He is responsible for managing a staff of over 25 employees in the Communications, System Planning, Substation Engineering and Protection groups. He directs and supervises the deployment and integration of electric systems to support the business processes related with advanced grid technologies and system reliability. He is also responsible for the development and implementation of goals, objectives, policies, and priorities of the electric utility.

Hamed Mohsenian-Rad (S'04–M'09–SM'14) received the Ph.D. degree in electrical and computer engineering from the University of British Columbia, Vancouver, BC, Canada, in 2008. He is currently an Associate Professor of electrical engineering with the University of California at Riverside, Riverside, CA, USA. His research interests include modeling, data analysis, and optimization in power systems. He was a recipient of the National Science Foundation CAREER Award, the Best Paper Award from the IEEE Power and Energy Society General Meeting, and the Best Paper Award from the IEEE International Conference on Smart Grid Communications. He was an Editor of the IEEE TRANSACTIONS ON SMART GRID and the *IEEE Power Engineering Letters*.