# A Machine Learning Approach to Event Analysis in Distribution Feeders Using Distribution Synchrophasors

Alireza Shahsavari[†], Mohammad Farajollahi[†], Emma Stewart[‡], Ed Cortez[§], and Hamed Mohsenian-Rad[†]
[†]Department of Electrical and Computer Engineering, University of California, Riverside, CA, USA
[‡]Grid Integration Group, Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[§]Riverside Public Utilities, Riverside, CA, USA

***Abstract*—This paper proposes a *machine learning* (ML) approach to detect, identify, and analyze the events that occur on distribution networks using data streams from real-world distribution-level phasor measurement units (PMUs). First, we develop two statistical event detection methods. One is based on testing absolute values around median and the other one is based on testing residuals on a non-linear estimation. Both methods use *moving windows* as well as *dynamic window size*. This allows us to detect events of different types and durations. Next, we use field expert knowledge to assign labels to the detected events and subsequently develop a multi-class support vector machine classifier to classify power quality events. Finally, we apply the above developed techniques to detect, identify, and analyze the events in a micro-PMU data stream from a real-world test site in Riverside, CA. We particularly study the oscillation events that occur somewhere across the distribution feeder itself, where their impacts are observed remotely by the available micro-PMUs.**

***Keywords*: Machine learning, event detection classification, distribution synchrophasors, non-linear estimation, residual test, oscillation events.**

## I. Introduction

In recent years, power distribution systems are becoming drastically more complex and more dynamic due to the increasing number of distributed energy resources. Therefore, there has been a growing interest in the power engineering community to enhance the level of situation awareness in distribution networks. However, this goal cannot be achieved by the existing traditional distribution-level monitoring systems that are often limited to supervisory control and data acquisition (SCADA) at distribution substation. The typical minutely reporting rate of SCADA systems is no longer sufficient to form the desired level of situation awareness.

Distribution-level phasor measurement units (D-PMUs), a.k.a., micro-PMUs, have emerged recently and gradually become commercially available to enhance both precision and reporting intervals in monitoring distribution networks. Micro-PMUs provide precise time-stamped GPS-synchronized reading of voltage and current phasors; on all three phases and once every 8.333 milliseconds [1]. There is a growing interest among electric utilities to deploy micro-PMUs in their distribution networks for different applications, c.f., [1]–[4].

### A. The Challenges and The Related Literature

The data streams that are generated by micro-PMUs introduce prominent *Big-Data challenges* to the power distribution industry. The key to address these challenges is to turn the micro-PMU data-streams into a set of events that are worth studying [4]–[6]. This of course requires developing new tools and techniques that can detect, identify, and analyze the type of events that occur on distribution networks and recorded by often only a few available micro-PMUs.

The related literature is still evolving. In [7], a semi-supervised event detection method is developed to detect events in micro-PMU data. In [8], the event detection of micro-PMU data is done based on examining the deviations from the expected linear relationship between nodal voltages and line current. In [9], a model-based event detection method is proposed to detect changes in admittance matrix using micro-PMUs data. In [10], a model-based event detection method is developed to detect permanent faults in distribution lines. In [11], an event classifier is designed to classify partially labeled events that are extracted from micro-PMU data. Finally, in [12], event classifiers are developed for malfunctioned capacitor bank switching and malfunctioned voltage regulator events. The transient signatures of these malfunctions are obtained from simulations and not from real-world micro-PMUs data streams.

While the above aforementioned event detection and event classification methods have already taken a great leap forward in this field, there is still a gap in the related literature due to two issues. First, since distribution systems are traditionally less equipped with high-resolution monitoring capabilities, the field expert knowledge for labelling the distribution system events is still limited; therefore, supervised and semi-supervised methods often overlook several events. This issue is also partly due to the natural challenges in reaching out to utility field experts by those who have the expertise to develop the data-driven methods in this field. Second, as for the model-based methods, they are often prone to failure due to lack of accurate and updated models for distribution systems.

### B. Our Approach and Contributions

In this paper, we seek to address the above challenges toward event analysis in micro-PMU data. Specifically, we seek to develop new methods to transform extremely large raw data of micro-PMUs to the type of information that is useful to distribution system operators. First, we propose two

This work was supported by NSF grants 1462530 and 1253516; DoE grant EE 0008001; UCOP under Grant LFR-18-548175; and NASA MIRO grant NNX15AP99A. The corresponding author is H. Mohsenian-Rad, e-mail: hamed@ece.ucr.edu.

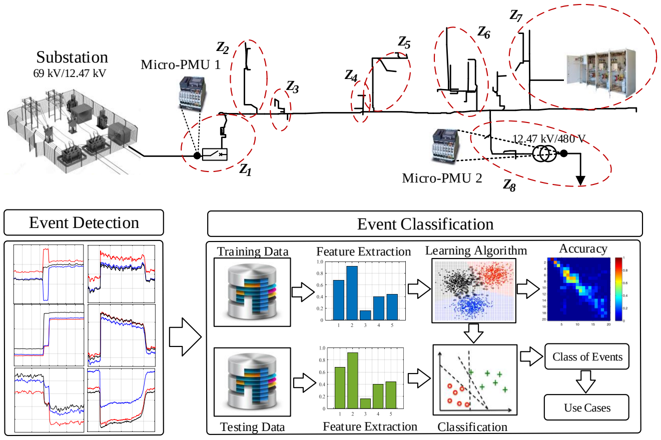

Fig. 1. The real-world distribution feeder that is studied in this paper and the designed Machine Learning application framework. The zones $Z_1$ to $Z_8$ will explained later in Section IV-C.

novel *model-free* event detection methods to pick out events from an ongoing stream of micro-PMU data. The first event detection method is based on testing absolute values around median by considering *moving window* and *dynamic window size*. The second event detection method is based on examining the non-linear relationship between nodal voltage and nodal current injection. For this purpose a non-linear estimator is designed that can detect the events by conducting a residue test. The proposed event detection methods are studied in several test scenarios based on real-world data. Subsequently, the detected events are labeled into three categories based on their root-causes using a novel data-driven event labeling and using the field expert knowledge. Finally, a *multi-class support vector machine* (multi-SVM) classifier is trained to classify the events.

The effectiveness of the developed event detection and classification methods is examined on one week of real-world data, i.e., 1.7418 billion data points, from two micro-PMUs on a distribution feeder in Riverside, CA, see the upper-portion of Fig. 1. One micro-PMU is installed at the feeder-head transformer and one micro-PMU is installed at a load transformer. The proposed overall machine learning framework is also shown in the lower-portion of Fig. 1.

Last but not least, in this paper, we particularly look into the oscillation events that occur somewhere across the distribution feeder itself and their impacts are observed remotely by the available micro-PMUs. This particular use case is of importance in practice because traditionally the oscillations at distribution-level are not investigated much; as opposed to the oscillations at transmission-level.

## II. Two Model-Free Data-Driven Event Detection Methods

### A. Method I: Absolute Deviation Around Median

Let $D_i := [d_1, \ldots, d_n]^T$ denotes a sequence of measurements from a micro-PMU, e.g., current magnitude, where $n$ denotes number of observation samples in data sequence $i$. 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:

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

where $M[\cdot]$ and $|\cdot|$ denote the median and absolute value, respectively. Note that, constant coefficient $\gamma$ denotes the underlying distribution in $D_i$, disregarding the abnormality induced by event. Mostly, it is considered that $\gamma$ is equal to $1/\phi(0.75)$, where $\phi(0.75)$ is the 0.75 quantile of that underlying distribution. A typical value for coefficient $\gamma$ is 1.4826 [13]. There exists an event within data sequence $D_i$ if there is a data point $k = 1, \ldots, n$ for which any of the following inequalities holds:

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

where $\zeta^-$ and $\zeta^+$ denote thresholds to detect the overshoot and undershoot in data sequence, respectively. The threshold in (2) may result in detecting several data points of each event as event. Thus, we need to revise and replace (2) with

$$\begin{aligned} \underline{D_i} &\leqslant M[D_i] - \zeta^- MAD_i \\ M[D_i] + \zeta^+ MAD_i &\leqslant \overline{D_i}, \end{aligned} \tag{3}$$

where $\overline{D_i} := \max\{d_1, \ldots, d_n\}$ and $\underline{D_i} := \min\{d_1, \ldots, d_n\}$. The choice of parameters $\zeta^-$ and $\zeta^+$ and the size of the data sequence window $n$ have impact on the performance of the detection method. While $\zeta^-$ and $\zeta^+$ are often selected empirically, choosing the right window size $n$ is very challenging. One may ask about the possibility of detecting all events in one window with no information about *event duration* as well as *event start time*. In order to overcome these challenges, we propose considering *dynamic window size* as well as *moving window*. On one hand, the dynamic window size can help to compensate the lack of information about event duration. On the other hand, the moving window can help to compensate the lack of information about start time of each event.

Although, in (3), we assure that multiple data points of each event are not detected as events in one window size, the dynamic window size and moving window may result in multiple detection of each event in different windows. Thus, we define an indicator function $\mathbb{I}(\cdot)$, such that If $\mathbb{I}(D_i) = 1$ if the condition in (3) holds for data sequence $D_i$; and if $\mathbb{I}(D_i) = 0$ otherwise. The pseudo-code of the proposed data-driven event detection is presented in Algorithm 1.

The impacts of applying dynamic window size and moving window are shown on 30 seconds of micro-PMU data in Fig. 2. The data sequence includes two major events at $t_1$ and $t_2$. Fig. 2(a) shows the case where the window size is fixed at five seconds, i.e., $n = 600$ micro-PMU samples. In this case, none of the events are detected. In Fig. 2(b), the event detection is conducted for five window sizes $n = 120, 360, 600, 840, 1080$ with considering moving window equal to the half of the window sizes, $\tau = n/2$. Accordingly, there are 10 upper-bound margins and 10 lower-bound margins as shown in Fig. 2(b). As it can be seen from this figure, both events are detected by considering dynamic window and moving window. Thus, both dynamic window sizes and moving windows are necessary to assure detecting *all* events.

**Algorithm 1** Data-Driven Event Detection

```
Require: $\mathbb{I} = [0, \dots, 0]^T_{N\times 1}$, $\gamma = 1.4826$, $n \geqslant 0$, $\tau \geqslant 0$
  for all window $n$ do
    for all shift value $\tau$ do
      $MAD_i \leftarrow \gamma \cdot M\left[|D_i - M[D_i]|\right]$
      $\underline{D_i} \leftarrow \min\{d_1, \dots, d_n\}$
      if $\underline{D_i} \leqslant M\left[D_i\right] - \zeta^- MAD_i$ then
        if $\mathbb{I}(D_i) = 0$ then
          $\mathbb{I}(D_i) \leftarrow 1$
        end if
      end if
      $\overline{D_i} \leftarrow \max\{d_1, \dots, d_n\}$
      if $M\left[D_i\right] + \zeta^+ MAD_i \leqslant \overline{D_i}$ then
        if $\mathbb{I}(D_i) = 0$ then
          $\mathbb{I}(D_i) \leftarrow 1$
        end if
      end if
    end for
  end for
```

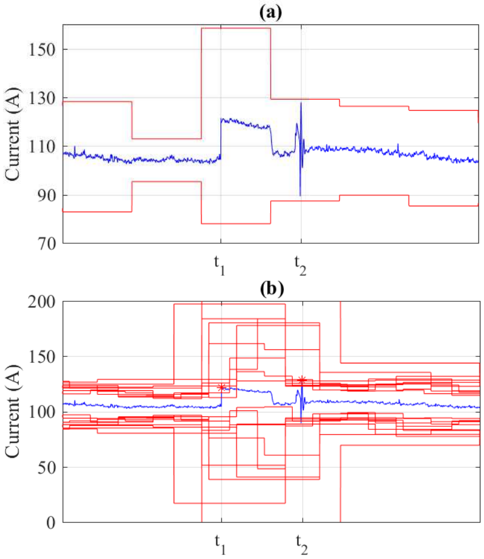

Fig. 2. Effect of moving window and dynamic window size on event detection: (a) static window size without moving window: one event is detected at $t_2$; (b) dynamic window size with moving window: both events are detected.

### B. Method II: Residual Test on Non-Linear Estimation

Let $V_i := [|v_1|, \dots, |v_n|]^T$ and $I_i := [|i_1|, \dots, |i_n|]^T$ denote sequences of voltage magnitude and current magnitude from a micro-PMU, respectively. From Circuit Theory, we know that in steady-state the relation between $|v_k|$ and $|i_k|$ at data point $k = 1, \dots, n$ is as:

$$s_k = |v_k| \cdot |i_k|, \tag{4}$$

where $s_k$ denotes apparent power at downstream of micro-PMU. During the transient of events, the (4) includes higher order harmonics beside the fundamental harmonic.

Let assume that there exists no major event in a window with $n$ data points. Thus, we can assure that the downstream load of micro-PMU is almost constant in $n$ data points. A non-linear estimator can be designed to estimate $I_i$ by solving the following optimization problem:

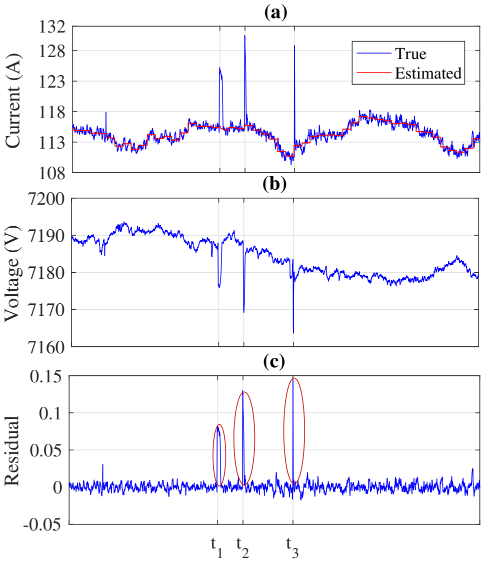

Fig. 3. Event detection by estimating current magnitude: (a) current magnitude; (b) voltage magnitude; (c) residual.

$$\underset{B}{\text{minimize}} \quad \left\| \tilde{I}_i - I_i \right\|_2, \tag{5}$$

where $\tilde{I}_i := [|\tilde{i_1}|, \dots, |\tilde{i_n}|]^T$ is the estimated current magnitude in data sequence $i$, as:

$$\tilde{I}_i = b_1 + \frac{b_2}{V_i}. \tag{6}$$

where $B := [b_1, b_2]^T$ is the regression coefficient vector. If there exists an event in data sequence $i = 1, \dots, n$, e.g., such as load switching event, during transient of the event, (4) does not hold. While, it hold for pre-event and post-event. Consequently, the non-linear estimator, that is designed only fundamental frequency, fails to estimate the $|i_k|$ during transient period of event. Thus, the residues corresponding to event data points are larger than those during pre-event and post-event.

Fig. 3(a) shows the true current magnitude and estimated current magnitude of micro-PMU 1 during 50 seconds. Here, the non-linear estimation is conducted every second, i.e., $n = 120$ micro-PMU data points. The data sequence includes three major events at $t_1$, $t_2$, and $t_3$. From Fig. 3(a) and (b), it seems that all three events are load turning on events. From Fig. 3(a), we can see that the non-linear estimator fails in estimating current magnitude during these major events. Fig. 3(c) shows the residues in estimating current magnitude. As it can be seen, the residues at $t_1$, $t_2$, and $t_3$ are the largest residuals.

Also, Fig. 4(a) shows the true and estimated voltage magnitude of micro-PMU 1 during 50 seconds. The non-linear estimation is conducted every second to estimate the voltage magnitude. The data sequence includes two major events at $t_1$ and $t_2$, the former event is initiated from load switching, while latter one is a voltage step-down event which may be initiated from operating a voltage regulator in upstream-level, see Figs. 4(a) and (b). From Fig. 4(b), we can see that the non-linear estimator fails in estimating voltage magnitude during

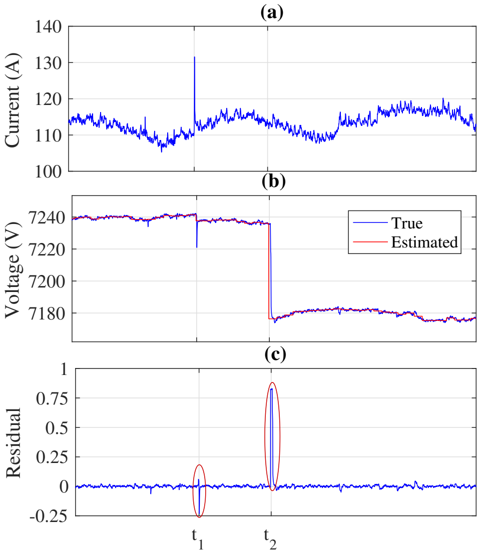

Fig. 4. Event detection by estimating voltage magnitude: (a) current magnitude; (b) voltage magnitude; (c) residual.

these events. Fig. 3(c) shows the residues in estimating voltage magnitude. As it can be seen, the events at $t_1$ and $t_2$ can be detected from residue test.

In Figs. 3 and 4, we detect events by considering a *fixed* threshold on residues or by applying largest normalized residual test. One may ask, what is the advantage of considering a threshold on residues against applying a threshold on data stream? The answer to this question is that the threshold on residues is *fixed*, while if we want to apply a threshold on data sequence, it should be updated for each window to consider the steady state changes. Also, the residual test based event detection method is not sensitive to the estimation window size. For instance, in Fig. 3, we can obtain almost the same residues by considering $n = 1200$ instead of $n = 120$.

## III. Supervised Event Classification

Broadly speaking, the events that are captured by using the event detection methods in Section II can be categories into three classes:

- Class I. Events initiated from upstream of micro-PMU 1, i.e., at transmission level or another distribution feeder: These events are mostly observed as either *sustained steps* or *temporary fluctuations* in feeder-level voltage magnitude and customer-level voltage magnitude. These events has no significant effect on current magnitudes and active powers. The step-down in voltage magnitude may initiated from switching-off a capacitor bank in upstream-level or operating a voltage regulator in upstream-level.
- Class II. Events initiated from downstream of micro-PMU 2, i.e., at customer location that hosting micro-PMU 2: These events are mostly initiated by *load switching* at downstream of micro-PMU 2. Based on the size of the fluctuations in current magnitude, we can conclude that these events are due to turning on large loads at downstream of micro-PMU 2, such as HVAC loads or energy storages.
- Class III. Events initiated from somewhere between the two micro-PMUs across the distribution feeder of interest: These events are mostly observed as temporary fluctuation 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. These events can be due to a wide verity of causes, such as load switching, capacitor bank switching, and protection operation.

### A. Data-Driven Feature Selection

Feature selection is a key step towards developing an automated data-driven mechanism to classify the events. In Machine Learning, features are defined as quantifiable properties of events. In this paper, we propose features in two broad categories as for the purpose of event classification:

- **Single-Stream Features:** These features are quantifiable properties of single data stream, such as features that are derived from voltage magnitude, current magnitude, active power, and reactive power. In this paper, standard deviation and absolute difference are chosen as single-stream features. Thus, the total number of single-stream features is 16.
- **Multi-Stream Features:** These features are derived from any two combination of synchronized data sequences, whether from the same micro-PMU or two micro-PMUs. Here, we consider correlation between any two of the eight data streams as multi-stream features. Thus, the total number of multi-stream features is 28.

### B. Multi-SVM Classifier

In this section, we train a multi-class support vector machine (multi-SVM) classifier to separate the events according to the labels that are introduced in Section III. The SVM was originally developed for binary classification [14]. Consider $m$ events that are detected by using methods in Section II as training samples. For each training sample $i = 1, \ldots, m$, let $X_i$ denote a 44-dimensional vector of extracted features. Also, let $y_i \in \{-1, 1\}$ denote the assigned label for sample $i$. If the training samples are linearly separable, there exists a hyperplane $W^T X + b = 0$ in the $44 \times 44$ feature space that separates two classes, where $W$ is a $44 \times 1$ weight vector, and $b$ is a scalar. When the training samples are linearly separable, there are an infinite number of hyperplanes that separates classes. The SVM yields to find the optimal hyperplane that has the maximum distance between the hyperplane and the set of training samples. In the case that training samples are not linearly separable, there is no such hyperplane that separates all $m$ events. This issue can be resolved by adding some slack variables to the optimization problem. The binary-SVM classifier is formulated as:

$$\begin{aligned} \underset{W,b,\xi}{\text{minimize}} \quad & \frac{1}{2}\|W\|_2^2 + \lambda \sum_{i=1}^{n} \xi_i && \text{(7a)} \\ \text{subject to} \quad & y_i\left(W^T X_i + b\right) \geq 1 - \xi_i, \; i = 1, \ldots, m && \text{(7b)} \\ & \xi_i \geq 0, \; i = 1, \ldots, m, && \text{(7c)} \end{aligned}$$

TABLE I
NUMBER OF DETECTED EVENTS IN CASE I AND CASE II.

| $n$ | 120 | 240 | 360 | 480 | 600 | 720 | 840 | 960 | 1080 | 1200 |
|---|---|---|---|---|---|---|---|---|---|---|
| Case I | 302 | 200 | 197 | 153 | 125 | 125 | 106 | 101 | 95 | 88 |
| Case II | 481 | 303 | 271 | 223 | 206 | 190 | 172 | 151 | 150 | 123 |

TABLE II
NUMBER OF DETECTED EVENTS IN CASE III AND CASE IV.

| $n$ | 120 | 240 | 360 | 480 | 600 | 720 | 840 | 960 | 1080 | 1200 |
|---|---|---|---|---|---|---|---|---|---|---|
| Case III | 302 | +64 | +29 | +13 | +4 | +3 | 0 | 0 | +1 | 0 |
| Case IV | 481 | +50 | +19 | +5 | +5 | +4 | 0 | 0 | 0 | 0 |

where $\xi_i$ denotes slack variable corresponding to training sample $i$. Also, $\lambda$ denotes the tuning parameter used to balance the importance of the margin and the training error. If a training event falls on the correct side but within the margin, then $0 < \xi_i < 1$, while if a point falls on the wrong side of the hyperplane $\xi_i \geq 1$, otherwise $\xi_i = 0$.

We utilize binary-SVM classifier to solve multi-class classification problem by decomposing the problem into several binary classification problems. Several methods have been developed in literature to decompose multi-class classification problem, such as one-against-one (OvO), one-against-all (OvA), and directed acyclic graph SVM (DAGSVM) [14]. In this paper, we use OvA decomposition method. Each binary SVM problem results in a separating hyperplane to separate one of the $c$ classes from the rest of the $c-1$ classes. Accordingly, solving $c$ binary classifications, using (7), results in $c$ separating hyperplanes. The decision on class prediction for testing sample $i$ is made as:

$$y_i = \underset{l=1,\ldots,c}{\arg\max} \left( W_l^T X_i + b_l \right). \tag{8}$$

Thus, in (8), the testing sample $i$ is assigned to class $l$, which has the largest value of the decision function.

## IV. CASE STUDIES

In this section, the proposed event detection methods and event classification method are applied to one week real-world data from two micro-PMUs shown in Fig. 1.

### A. Event Detection in micro-PMU Data Sequence

This section examines the effectiveness of the proposed event-detection methods introduced in Section II, using one day data of current magnitude from micro-PMU 1. Also, we study the effect of the moving window and dynamic window sizes. To such aim, Method I is applied to the following cases:

- Case I. Static window size without moving window;
- Case II. Static window size with moving window;
- Case III. Dynamic window size without moving window;
- Case IV. Dynamic window size with moving window.

In order to compare the above case-studies, the detected events in each case study are compared with those that are detected in Case IV, which includes 564 events. Table I reports the number of the detected events in Case I and Case II with considering 10 static window sizes. For instance, considering $n = 120$ micro-PMU data points, the moving window results in detecting 179 more events. From this table, we can conclude that most of the events are detected in smaller windows. Also, Table II shows the number of *additional* events that are detected in each window size in Cases III and IV. From this table we can conclude that considering moving window results in detecting more events in each dynamic windows.

TABLE III
EVENT DETECTION PERFORMANCE INDEXES FOR METHODS I AND II.

| Method | TP | FP | FN | PPV | TPR | $F_1$ Score |
|---|---|---|---|---|---|---|
| Method I | 541 | 23 | 8 | 0.96 | 0.985 | 0.97 |
| Method II | 546 | 6 | 3 | 0.99 | 0.99 | 0.99 |

Next, we compare the effectiveness of the Method I with Method II, considering dynamic window sizes and moving windows. We define positive predictive value (PPV), true positive rate (TPR), and detection accuracy ($F_1$ score) as [15]:

$$PPV = \frac{TP}{TP+FP}, \tag{9}$$

$$TPR = \frac{TP}{TP+FN}, \tag{10}$$

$$F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR}, \tag{11}$$

where $TP$ denotes number of true positive data points which is the number of events that are correctly identified as events, $FP$ denotes number of false positive data points which is the number of normal data points that are identified as events, and $FN$ denotes number of false negative data points which is the number of events that are identified as normal data points. The performance indexes for both detection methods are reported in Table III. The obtained results shows Method II outperforms the Method I.

### B. Classifier Design

The proposed event detection method based on residual test is applied to one week micro-PMUs data. In total, 5881 events has been detected in one week data. Among the events detected, 961, 1,208, and 3,711 events are labeled in Class I, Class II, and Class III, respectively. We partitioned the events to training dataset and test dataset, where the training data set includes 670 events, among them 143, 223, and 304 events are labeled in Class I, Class II, and Class III, respectively.

A multi-SVM classifier is designed to separate the detected events. According to the (7), the three binary classifier results in three separating hyperplanes in $44 \times 44$ feature space. In order to visually show the separating hyperplanes in feature space, two dominant features among 44 features are selected to train and test classifier. Thus, the separating hyperplanes in $2 \times 2$ feature space are separating lines. Fig. 5(a) shows the training data points in $2 \times 2$ feature space as well as three separating lines obtained from three binary SVMs. Each separating line is somehow designed to separate events of one class from the rest of the events. The confusion matrix corresponding to training dataset is shown in Fig. 6(a). Here, the accuracy of training is 90.6%. Next, the separating hyperplanes are applied to the test dataset, and decision on class prediction is made using (8). Fig. 5(b) shows the test dataset in $2 \times 2$ feature space and separating lines. The overall classifier testing

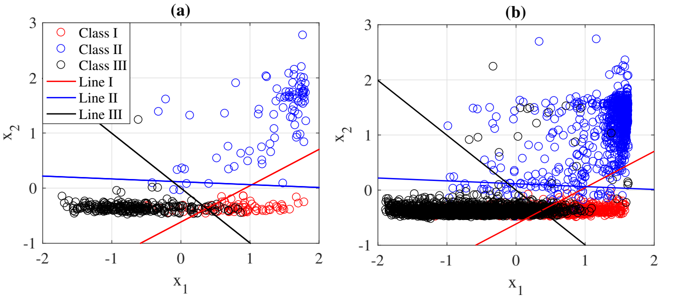

Fig. 5. Target classes in two dimensional feature space and separating lines: (a) training data points; (b) test data points

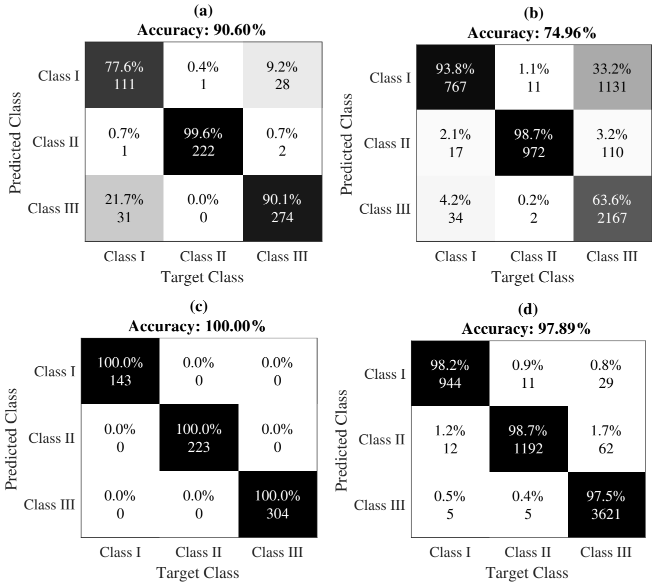

Fig. 6. Confusion matrices obtained by classifier: (a) training data considering two dominant features; (b) test data considering two dominant features; (c) training data considering 44 features; (d) test data considering 44 features

accuracy is 74.96%. Fig. 6(b) shows the confusion matrix for test dataset. As it can be seen, the classifier fails to correctly separate several data points due to not using 44 features.

Next, we design a classifier by considering all 44 features that are introduced in Section III-A. Fig. 6(c) and (d) show the confusion matrices for training and testing datasets, respectively. As it can be seen from these matrices, the accuracy of training dataset is 100%. Also, the classifier accuracy for test data set increases to 97.89% by considering all 44 features. The accuracy of classification for test dataset verifies the performance of the proposed classifier.

## C. Analysis of Oscillation Events

As discussed in Section III, the events in Class III can be due to load switching, capacitor bank switching, protection devices operation. Among all events in this class, there exists several oscillation events as shown in Fig. 7. There is currently a limited understanding of the oscillation events within distribution systems. Thus, in this section, we scrutinize these events to characterize the oscillations on the under-study distribution feeder. We have applied Fourier analysis and Prony analysis to find frequency and damping components of oscillation events. Figs. 8(a) and (b) show frequency and damping components of oscillation events, respectively. As it can be seen, the average oscillation frequency is 5.2 Hz, while the average damping ratio is 2.97%.

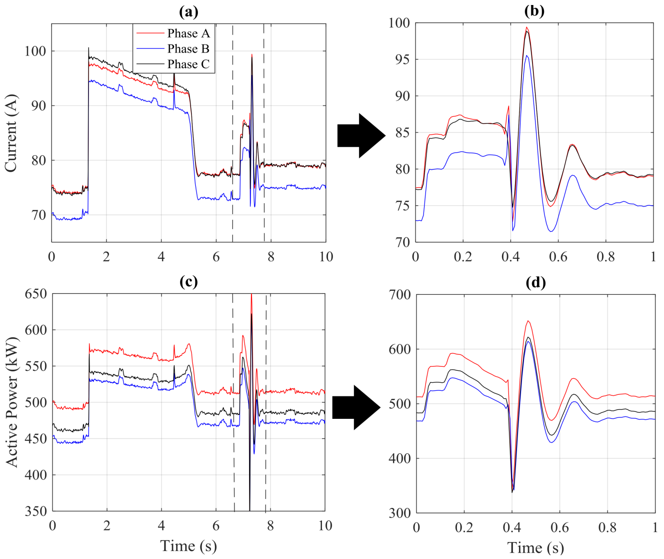

Fig. 7. Oscillation event corresponds to the measurements from micro-PMU 1: (a) and (b) current; (c) and (d) active power.

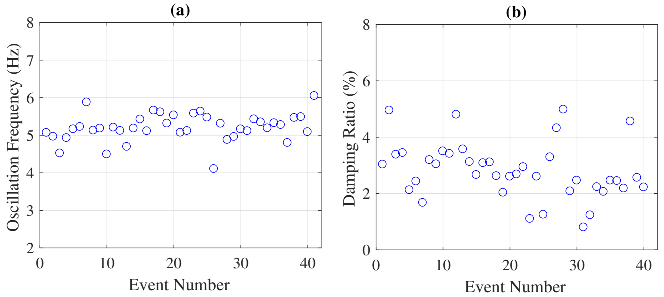

Fig. 8. (a) Frequency component of oscillation event; (b) damping component of oscillation event.

Next, we apply event source location identification (ESLI) method in [3], [16] to identify the location of the detected oscillation events. Here, the 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 [3], [16] for more details. For each individual oscillation event, we calculate the so-called zonal voltage discrepancy $\Delta V_z$ for each zone $z = 1, \ldots, 8$ [3], [16]. For each event, the location of event is identified as the zone with minimum zonal voltage discrepancy. Interestingly, we identified that all oscillation events are initiated from $Z_7$.

## V. Conclusions

This paper develops new methods to resolve Big-Data challenges in transforming unprocessed data of distribution-level PMUs to actionable information. Two novel model-free event detection methods have been proposed to pick out events from an ongoing stream of micro-PMU data. The first event detection method is built on testing absolute values around median by considering moving window and dynamic window size. The second event detection method is based on examining the non-linear relationship between nodal voltage and nodal current injection. The effectiveness of the proposed event detection methods are compared on real-world data from two micro-PMUs on a distribution feeder in Riverside, CA. The obtained results show that the accuracy of event detection by testing residues of non-linear estimation method is higher than the method by examining absolute values around median. Also, a multi-SVM classifier is trained to classify events based on their root-cause locations. The effectiveness of the designed classifier is examined on one week real-world data from two micro-PMUs on a distribution feeder in Riverside, CA. Finally, we particularly look into oscillation events that occurs somewhere in distribution feeder. The results in this paper could be of value to utilities to resolve Big-Data challenges.

## References

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

[2] A. Shahsavari, A. Sadeghi-Mobarakeh, E. Stewart, E. Cortez, L. Alvarez, F. Megala, and H. Mohsenian-Rad, "Distribution grid reliability versus regulation market efficiency: An analysis based on micro-PMU data," *IEEE Trans. on Smart Grid*, vol. 8, no. 6, pp. 2916 – 2925, Jun. 2017.

[3] 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. on Power Systems*, vol. 33, no. 6, pp. 6343 – 6354, Nov. 2018.

[4] A. Shahsavari, M. Farajollahi, E. Stewart, A. von 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 *IEEE PES ISGT*, Washington D.C., Apr. 2017.

[5] 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. of IEEE PES NAPS, Morgantown, WV*, Sep. 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. of IEEE PES NAPS, Morgantown, WV*, Sep. 2017.

[7] Y. Zhou, R. Arghandeh, and C. J. Spanos, "Partial knowledge data-driven event detection for power distribution networks," *IEEE Trans. on Smart Grid*, vol. 9, no. 5, pp. 5152 – 5162, Sep. 2018.

[8] M. Jamei, A. Scaglione, C. Roberts, E. Stewart, S. Peisert, C. McParland, and A. McEachern, "Anomaly detection using optimally-placed $\mu$PMU sensors in distribution grids," *IEEE Trans. on Power Systems*, vol. 33, no. 4, pp. 3611 – 3623, Jul. 2018.

[9] O. Ardakanian, Y. Yuan, R. Dobbe, A. von Meier, S. Low, and C. Tomlin, "Event detection and localization in distribution grids with phasor measurement units," *arXiv preprint arXiv:1611.04653*, 2016.

[10] M. Pignati, L. Zanni, P. Romano, R. Cherkaoui, and M. Paolone, "Fault detection and faulted line identification in active distribution networks using synchrophasors-based real-time state estimation," *IEEE Trans. on Power Delivery*, vol. 32, no. 1, pp. 381–392, Feb. 2017.

[11] Y. Zhou, R. Arghandeh, I. Konstantakopoulos, S. Abdullah, A. von Meier, and C. J. Spanos, "Abnormal event detection with high resolution micro-PMU data," in *Proc. of the IEEE Power Systems Computation Conference*, Genoa, Italy, Aug. 2016.

[12] I. Niazazari and H. Livani, "A PMU-data-driven disruptive event classification in distribution systems," *Electric Power Systems Research*, vol. 157, pp. 251 – 260, Apr. 2018.

[13] P. J. Huber, "Robust statistics." Wiley, 2009, pp. 107–109.

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

[15] A. Shahsavari, M. Farajollahi, E. Stewart, E. Cortez, and H. Mohsenian-Rad, "Situational awareness in distribution grid using micro-PMU data: A machine learning approach," *accepted for publication in IEEE Trans. on Smart Grid*, Jan. 2019.

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