PNNL-27812

Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

FY18 Discovery through Situational Awareness: Anomalies, Oscillations, and Classification

August 2018

BG Amidan
JD Follum
T Yin
N Betzsold

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <http://www.ntis.gov/about/form.aspx>
Online ordering: http://www.ntis.gov

This document was printed on recycled paper.
(8/2010)

PNNL-27812

FY18 Discovery through Situational Awareness: Anomalies, Oscillations, and Classification

BG Amidan
JD Follum
T Yin
N Betzsold

August 2018

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Executive Summary

This technical report is a summary of work performed under the DOE GMLC program under the project titled "Discovery through Situational Awareness". The focus of this work was to apply big data, statistical techniques to PMU data to do the following: 1) baseline PMU data and find anomalies with a focus on phase angle pairs, 2) identify the frequencies of dominant modal oscillations between different parts of a power system, and 3) identify and classify frequency events in PMU data.

This work was performed on PMU data from the Western Interconnect. Many of these algorithms will be applied to wide area phase angle pair data on the Eastern Interconnect in a tool called ESAMS (Eastern Interconnect Situational Awareness Monitoring System). ESAMS was built by EPG (Electric Power Group) and PNNL and is currently being installed on the Eastern Interconnect.

The results discussed in this report were made from 14 PMUs from data ranging from Oct 2016 to May 2018. The PMU data were measured 60 times each second. Fourteen phase angle pairs were included, with reactive and active power being calculated at 14 sites. All results have been de-identified for this report.

The anomaly-detection methodologies applied in the ESAMS tool are described in Section 2. A baseline of PMU data is created using the last 120 days of data. This baseline describes typical behavior as understood from the data. The baseline is then applied to the next day's data and anomalies from that baseline are identified. Plots are created to help describe what is anomalous during that specific moment in time. These algorithms are currently being installed on the Eastern Interconnect. FY19 work will include discussing the results and findings from the ESAMS tool.

A new method of identifying the frequencies of dominant modal oscillations between different parts of a power system is described in Section 3. The method is based on a novel spectral estimator that is robust to the effects of forced and ringdown oscillations. As a result, the ambient-only spectral estimate that it produces provides insight into baseline system behavior in terms of inter-area oscillations. This information can be used to prepare for the deployment of a mode meter or to better understand how system changes impact inter-area modes of oscillation. The method also identifies when large system events occur, so it can serve as a useful complement to methods specifically designed for that purpose. The methods described in Section 3 were also integrated into the ESAMS tool.

A list of 199 frequency events provided by Bonneville Power Administration were used to create algorithms to detect frequency events and to create classification rules that can be used to identify future frequency events. Section 4 discusses the performance of 7 different event detection methods. Accuracy percentages ranged from 93.59% to 99.16%, with most methods being very fast (see Table 4-1). GBM (Gradient Boosting Machine) performed well and quickly. These 7 different methods were also tested to determine how well each built classification rules on training data and then applied those rules to identify future frequency events. Accuracy percentages ranged from 89.09% to 99.9%, with all methods performing quickly (see Table 4-2). GLM (General Linear Model) was viewed as the top performer when

considering accuracy and processing time. Because the list of events consisted of 80% of them as active power events and only 20% as faults, all methods struggled to identify events as faults. Small faults were especially difficult to identify. A separate algorithm was created to help with this. Additional work will be done in FY19 on event detection using more data and more events and refinements to the methods.

Acknowledgements

The Pacific Northwest National Laboratory (PNNL) work was funded by the U.S. Department of Energy. The guidance provided by DOE Project Manager, Joseph Eto, is greatly appreciated. The input, data, and support provided by members of the Eastern Interconnect (PJM, MISO, NYISO, and ISONE) are also appreciated.

1.0 Introduction

This technical report summarizes work that was done in FY18 under the DOE GMLC program under the task titled "Discovery Through Situational Awareness" (GMLC0070). Mathematical and statistical algorithms were used to do the following: 1) Find anomalies in PMU data, with a focus on phase angle pairs, 2) identify the frequencies of dominant modal oscillations between different parts of a power system, and 3) identify and classify frequency events in PMU data.

PMU data from 14 sites on the western interconnect were used for these investigations. About 20 months of data were available for analyses (Oct 2016 to May 2018). All site locations have been de-identified for this report.

This paper is organized as follows –

- Section 2 discusses the anomaly detection methodology that is being implemented in the ESAMS (Eastern Interconnect Situational Awareness Monitoring System) software.
- Section 3 describes a new methodology that can be used to identify the frequencies of dominant modal oscillations between different parts of a power system. This work was recently presented at the PMAPS 2018 conference.
- Section 4 investigates using PMU data to identify frequency events and then using those data characteristics to classify and identify future frequency events.

2.0 Anomaly Detection Methodology Used for ESAMS

The ESAMS (Eastern Interconnect Situational Awareness Monitoring System) anomaly detection calculation is a multi-step process. It is a data-driven approach that uses past data to create a baseline of behavior and then compares new data to that baseline, to determine if anomalous behavior has occurred. This section goes step by step through the process and explains the methodology used. A flowchart is given to further show the order and flow of the process.

2.1 Reading the Data

Currently the functions read csv formatted PMU data. There is also the capability to read PDAT format (PDAT is a proprietary binary format used at BPA). The data must have a date / time stamp in the first row, and the variable names in the first column. These names need to be such that the type of variable (i.e. frequency, voltage) can be deciphered. Each data file should contain 1 minute of data and the data should be at either a 30 Hz or 60 Hz rate. Missing values are acceptable and should be entered as NA or NaN. Currently the function only keeps the phase angle pair differences and discards the other variables. Future analyses could be expanded to include some or all of these other variables or data from other sources.

2.2 Calculating Features from the Data

Once a one minute PMU data file is read in, features are extracted for each variable of interest (for now phase angle pair differences). The features capture characteristics about the data stream, including value, variability of the value, rate of change, and variability in the rate of change. These features are calculated by fitting a quadratic regression equation for every second of data, where time is the independent variable (x) and the value for the variable of interest is the dependent variable (y). The intercept (representing a smoothed calculation of the value), slope (representing the rate of change), quadratic term (representing the rate of the rate of change), and noise (representing how well the data fits a quadratic regression equation) are calculated for each second. This results in 60 measures of each of these four regression coefficients for a complete minute. These measures are then summarized by calculating the mean and standard deviation of each. At this time the rate of rate of change (quadratic term) and the noise coefficients are not included in any analyses, so the resulting features for each variable for each minute of data are: 1) the mean value; 2) the standard deviation of the values; 3) the mean rate of change; and 4) the standard deviation of the rates of change. These features are saved each day into what is called a signature matrix and stored in the features database. This matrix includes all 1440 minutes of the day on the rows and the four features for each variable of interest (phase angle pair differences) on the columns. These signature matrices are used in all future analyses discussed in the forthcoming steps.

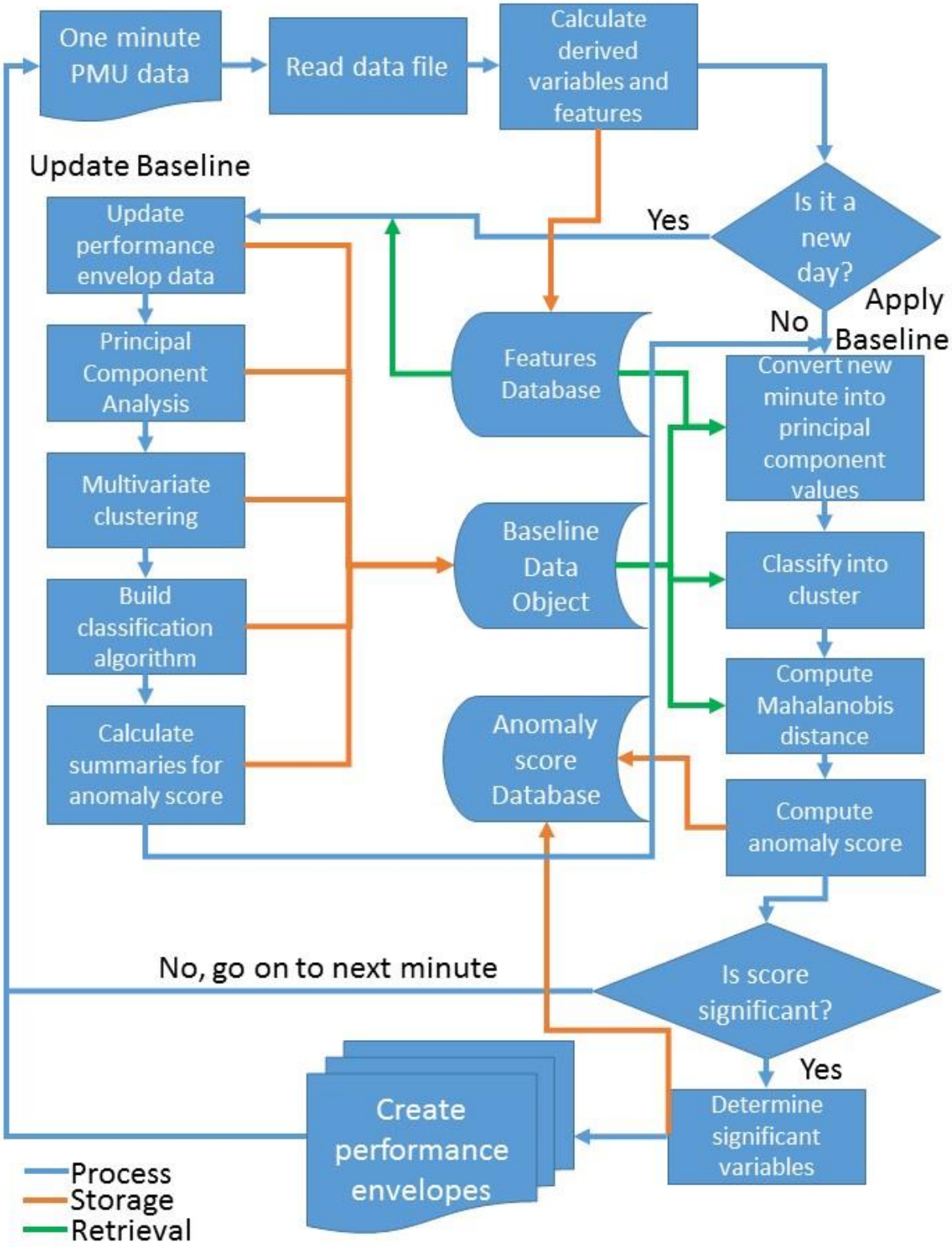
2.3 Creating a Baseline from Past Data

Each evening, after the processing for a day is done (typically at midnight), the "baseline" is calculated. The baseline relies on multivariate statistical techniques to capture how the data are typically behaving. The baseline captures all the necessary elements to calculate the anomaly score for new data. The baseline is calculated using the last 120 days of signature matrices. If there are less than 120 days of data available then it uses what is available. No baseline is created until at least one day of data is available. The full 120 day matrix will have 172,800 rows (assuming that every minute during that timeframe had data available). The baseline consists of 5 steps, which are detailed below.

The first step is to update the performance envelope gray background for each variable with the data from the last 120 days. The gray background represents the values for each variable that have been observed during each minute of the day. Darker the gray indicates that more days had that value during that minute of the day. These matrices with counts in them are stored and then used when plots are needed. Performance envelopes are further discussed in a section below.

The second step is to perform principal component analysis (PCA) on the data (the 120 days of signature matrices). First the original, signature data are centered by subtracting the mean and scaled by dividing by the standard deviation within each column, with the mean and standard deviation of each column stored because they will be needed when applying the baseline to future data. PCA uses an orthogonal transformation on this centered and scaled data to convert it into linearly uncorrelated variables called principal components. These principal components represent linear combinations of the actual data that create or explain the most variance. Each principal component looks at a completely different way to make a linear combination with maximum variance when compared to the previous principal components. The principal components that account for at least 90% of the total variation in the original data are kept for analyses in the next steps. The rest of the principal components are considered to be noise and are removed from analyses. This step removes the collinearity that exists from the many variables that are highly correlated. This means that variables that are highly correlated will all make up the same principal component so that effect will only be represented once.

The third step performs multivariate statistical clustering (often called unsupervised learning) on the principal components. This takes each minute of data (each row) and groups it with similar minutes that have similar principal component values. These groupings make up the clusters. These clusters remain static for the next day of processing, but then can change when the baseline is updated. The number of clusters is an arbitrary decision, so it was decided that the number of clusters should equal the third root of the number of minutes in the dataset, rounded up (with a full dataset of 120 days in the baseline, this amounts to 56 clusters). Those minutes that are in sparsely populated clusters or even in a cluster by themselves (a singleton) are considered to be more atypical. *K*-means is used as the clustering algorithm; however, a different algorithm could be plugged in if desired. The proportion of minutes that fall into each cluster are calculated and these proportions are part of the anomaly score calculation.


One minute PMU data
Read data file
Calculate derived variables and features
Update Baseline
Update performance envelop data
Principal Component Analysis
Multivariate clustering
Build classification algorithm
Calculate summaries for anomaly score
Yes
Is it a new day?
No
Apply Baseline
Features Database
Baseline Data Object
Anomaly score Database
Convert new minute into principal component values
Classify into cluster
Compute Mahalanobis distance
Compute anomaly score
No, go on to next minute
Is score significant?
Yes
Determine significant variables
Create performance envelopes
Process
Storage
Retrieval

The fourth step creates a classification algorithm (often called supervised learning) using the principal components and the cluster assignments. This algorithm creates classifier rules which represent static definitions for each cluster. The rules are applied to a new data point (a minute) to determine which cluster that minute of data best belongs to. Cluster analyses are used to create the groups (clusters), while classification is used to define each group and then classify each new minute into a cluster. Linear discriminant analysis (LDA) is used as the classification algorithm; however, a different algorithm could be plugged in if desired.

The last step is to create the last pieces necessary to calculate the anomaly score. The Mahalanobis distance for each minute is calculated from the principal components. Mahalanobis distance measures how far a multivariate data point is from the center (mean). A Gamma distribution is fit to the Mahalanobis distances because it is a good fit for the right-skewed nature of the distances. The parameters for the Gamma are stored, as they are needed for the anomaly score calculations.

Outputs from these five steps make up the baseline and are used so that anomaly scores can be calculated for each new minute of data.

2.4 Applying the Baseline to New Data

As soon as a new minute of data becomes available, an anomaly score is calculated for that minute. This is done by first calculating the features for the new minute, using the methodology discussed above. Then, the features are converted into the principal components, using the principal component linear combinations that were updated in the baseline from the past 120 days. The classifier from the baseline is then used to classify the new minute into which cluster (group) the new minute of data is most like. The Mahalanobis distance is now calculated for the new minute with respect to the means and standard deviations calculated from the data used in the baseline. This distance value is then translated into a probability using a Gamma distribution and the Gamma parameters provided by the baseline. If this new minute is more anomalous, then it results in a value that is quite far from the typical values, receiving a probability value near 0. This probability is the first input into the anomaly score equation.

The second input comes from the clustering (grouping) results. The proportion of minutes that shared the same cluster as the new minute is calculated. For example, if there are 172,800 minutes used to get the baseline (1440 minutes x 120 days), and 17,000 of those minutes are in the cluster in which the new minute best classified into, then the cluster proportion for the new minute would be 0.0984 (17,000/172,800).

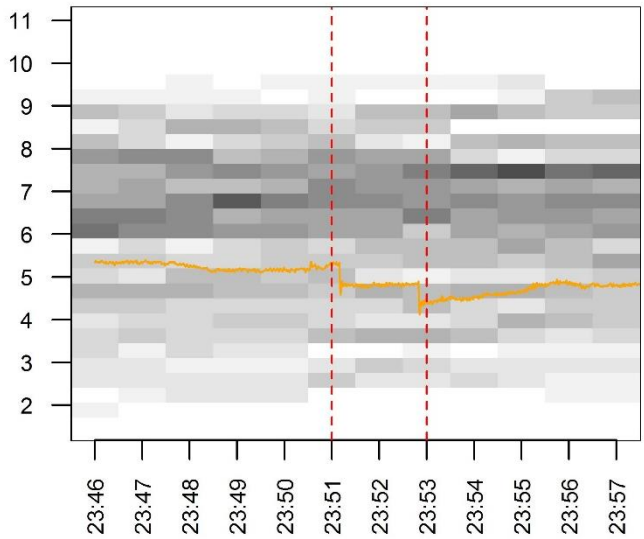
The anomaly score is calculated using the following equation: $Score = -log_e(p) - log_e(c)$, where p is the Gamma probability and c is the cluster proportion. Anomaly scores are always positive, with larger scores indicating that the minute is more unusual or anomalous. Scores generally range from a little bigger than 0 to values in the 20s. At this point, the cutoff for indicating a minute is anomalous is a score of 17 or more.

It is important to note that the new data, at this time, is not used to update the calculations for the baseline (principal components and clustering). The current baseline is not changed until the end of the day, in which the new data will be included in the updating of the baseline (principal components and clustering).

2.5 Performance Envelope Plots to Show Anomalous Moments

After an anomalous moment has passed, a performance envelope plot is created for those phase angle pairs that contribute most to the anomalous moment. An anomalous moment could be just 1 minute long, or many minutes long. The example plot below is an anomalous moment of duration 2 minutes. The algorithm combines all anomalous moments that are within 10 minutes of each other into one anomalous moment and then 5 minutes after that, the plot is produced. The plot contains the five minutes before the anomaly, the time during the anomaly, and then 5 minutes after the anomaly. The orange line shows the actual data for that time period. The time period between the two vertical, dotted, red lines represents when the anomalous moment occurred. The gray, background pixels represents the values that the specific phase angle pair had during the time that is represented on the x axis for the past 120 days. The darker the gray pixel, the more times that value was seen during that time period.

The example performance envelope below shows an anomalous moment that is 2 minutes long. In the first minute (23:51), there is a sharp drop in phase angle pair difference and then, in the second minute (23:52), it happens again. This plot is for a de-identified phase angle pair; however, in the ESAMS tool, it will be identified.



3.0 Oscillations

Reliable operation of a large power system requires that the system's electromechanical modes are sufficiently understood and monitored. The need for effective monitoring was clearly demonstrated in the August 10, 1996 breakup of the Western Electricity Coordinating Council (WECC) system. At that time, the existing models did not reflect the actual system dynamics (Kosterev, Taylor and Mittelstadt). Had effective monitoring been in place, system operators could have been warned that the system's damping was declining to dangerous levels.

Largely in response to the 1996 outage, the oscillatory modes of the WECC system are now monitored (Kosterev, Burns and Leitschuh). In systems where dynamics are less constraining than voltage stability or thermal limits, modes tend to be less understood. This is particularly true in large systems operated by several different entities. The large size makes model-based approaches difficult to implement, and the benefit of measurement-based methods is limited if synchrophasor data cannot be shared among different entities to achieve a wide-area view. Though this has been a severe limitation in the past, data sharing efforts such as those led by the Eastern Interconnection Data Sharing Network (Eastern Interconnection Data Sharing Network, Inc.) and Peak Reliability (Peak Reliability) are leading to an ever-wider view of power systems. Even in systems that are not currently dynamically constrained, changes in system configuration, generation mix, and typical line flows may lead to problems with modal oscillations over time. The data shared between entities can be leveraged to understand a system's modes to provide early identification of changes and potential problems.

In this section, a new method of identifying the frequencies of dominant modal oscillations between different parts of a power system is described. The method relies on an extension to the Modified Daniell-Welch (MDW) nonparametric spectral estimator proposed in (Follum and Tuffner, A multi-channel method for detecting periodic forced oscillations in power systems) and updated in (Follum, Tuffner and Agrawal, Applications of a New Nonparametric Estimator of Ambient Power System Spectra for Measurements Containing Forced Oscillations). While maintaining the original algorithm's ability to reject spectral content due to forced oscillations (FOs), the new algorithm also rejects wide-band spectral content due to ringdown oscillations that occur after large system events. The purely ambient spectral estimates that result can be used to identify dominant electromechanical modes at peaks in the spectra. Over time, these results can be used to detect changes in the modes. The method can be applied to many signals with relative ease because, as a nonparametric estimator, it does not require careful tuning. Following the naming convention of (Follum, Tuffner and Agrawal, Applications of a New Nonparametric Estimator of Ambient Power System Spectra for Measurements Containing Forced Oscillations), the spectral estimator is termed the Robust Modified Daniell-Welch (RMDW) estimator because it reliably estimates ambient spectra, even when other oscillations are present in the data.

Nonparametric spectral estimation has previously been used to gain insight into the electromechanical modes of power systems. However, continued installation of phasor measurement units (PMUs), improved data availability and quality, and increased data sharing is leading to longer and broader datasets that will benefit from more advanced analysis approaches. Thus, this report also discusses the use of the *k*-means clustering algorithm (MacQueen) to make analysis of large datasets practical. The

clustering method is used for two distinct purposes: 1) to distinguish between dominant electromechanical mode frequencies and 2) to identify system events. Though the main motivation for detecting system events is to mitigate their impact on spectral estimates, the detection results could also be useful to system engineers. The method can also be used to validate the results of applications designed specifically to detect system events.

The rest of this section is organized as follows. Background on spectral estimation and clustering techniques leveraged in the proposed method is provided in Section 3.1. The RMDW spectral estimator is then described in Section 3.2, followed by a description of the mode identification procedure in Section 3.3. Section 3.4 contains results from experiments conducted on simulated and measured data.

3.1 Background

The ambient spectral estimator proposed in this report is an extension of the novel spectral estimator proposed in (Follum, Tuffner and Agrawal, Applications of a New Nonparametric Estimator of Ambient Power System Spectra for Measurements Containing Forced Oscillations). A description of the base algorithm is provided in Section 3.1.1, before the new RMDW estimator is described in Section 3.2. Background on the *k*-means clustering algorithm, which is also used in the RMDW estimator, and an associated metric is provided in Section 3.1.2.

3.1.1 The Modified Daniell-Welch Spectral Estimator

The objective of spectral estimation is to capture the power of a signal as a function of frequency. The MDW estimator proposed in (Follum, Tuffner and Agrawal, Applications of a New Nonparametric Estimator of Ambient Power System Spectra for Measurements Containing Forced Oscillations) combines and alters elements from the Welch and Daniell spectral estimators to retain spectral content from ambient and ringdown oscillations while rejecting spectral content from FOs. For the applications discussed in (Follum, Tuffner and Agrawal, Applications of a New Nonparametric Estimator of Ambient Power System Spectra for Measurements Containing Forced Oscillations), these characteristics were ideal.

The MDW estimator, like many nonparametric spectral estimators, is based on the simple periodogram given by

$$\hat{\phi}_s(\omega_r) = \frac{1}{M}\left|\sum_{m=0}^{M-1} y_s(m)e^{-j\omega_r m}\right|^2, \tag{3.1}$$

where $y_s(m)$ denotes a length M segment of data. The term inside the absolute value is the Discrete Fourier Transform (DFT) and

$$\omega_r = \frac{2\pi r}{M}, 0 \le r \le \frac{M}{2} \tag{3.2}$$

is the sampled angular frequency with range $\omega_r \in [0, \pi]$. Consider the set of N_m frequencies

$$\Omega_\rho = \left\{\omega_\rho | r - \frac{N_m - 1}{2} \leq \rho \leq r + \frac{N_m - 1}{2}\right\} \tag{3.3}$$

and denote the median operator as $med\{\cdot\}$. Then the median-filtered simple periodogram, indicated with a $'$, is given by

$$\hat{\phi}_s'(\omega_r) = med\{\hat{\phi}_s(\Omega_\rho)\} \tag{3.4}$$

The median filter is applied in the frequency domain to suppress sharp peaks due to FOs. The MDW estimator is obtained by averaging the median-filtered simple periodograms from overlapping segments of data to reduce variance:

$$\hat{\phi}_{MDW}(\omega_r) = \frac{1}{Q \times S} \sum_{s=1}^{S} \hat{\phi}_s'(\omega_r) \tag{3.5}$$

where S is the number of segments and

$$Q = \sum_{j=1}^{\frac{1}{2}(N_M+1)} (N_m - j + 1)^{-1} \tag{3.6}$$

is a scaling term. See (Follum, Tuffner and Agrawal, Applications of a New Nonparametric Estimator of Ambient Power System Spectra for Measurements Containing Forced Oscillations) for further details.

3.1.2 Clustering

Clustering is the process of partitioning N points (vectors) into k sets. An early approach to selecting these clusters based on the distance of each point from the center (centroid) of each set is *k*-means clustering (MacQueen). In this work, the specific implementation known as Lloyd's algorithm was utilized (Lloyd). An outline of the algorithm's steps, which follows the description in (MathWorks), is as follows:

1. Initialize the centroids as a random set of k points
2. For each point:
    - Compute the distance to each centroid
    - Assign the point to the cluster with nearest centroid
3. Recalculate each centroid as the mean of the cluster's points
4. Repeat steps 2-3 until cluster assignments remain unchanged

Typically, step 2 is implemented based on Euclidean distance. However, a different distance measure is used in the RMDW algorithm, as described in Section 3.2.

Note that the final partitions are sensitive to the set of points selected in step 1. Further, the number of clusters k must be specified at the outset, but is unknown for some applications. Thus, it is common to rerun the algorithm several times to select k and ensure that a set of "poor" initial centroids do not result in poor performance. To choose the final partition across these trials, the silhouette metric is utilized.

The silhouette was proposed as a graphical aid for interpreting and validating the partitioning of data into clusters (Rousseeuw). It applies equally well for the wide array of clustering algorithms. The graphical component is not considered in this paper to allow the algorithms to be automated, but the metric's numerical value remains useful.

To express the metric, begin by denoting points in cluster C_j as $x_j(n)$ for $j = 1,2,\dots,k$ and $n = 1,2,\dots,N_j$. Thus, there are k clusters and cluster C_j has N_j points. For point $x_j(n)$, the term

$$a\left[x_j(n)\right] = \frac{1}{N_j - 1}\sum_{\substack{l=1 \\ l\neq n}}^{N_j} D\left[x_j(n)\,, x_j(l)\right] \tag{3.7}$$

where $D\left[x_j(n)\,, x_j(l)\right]$ is the distance between points $x_j(n)$ and $x_j(l)$, captures the average dissimilarity between the point and all other points in the same cluster. A small value indicates the point is a good fit for the cluster. The term

$$b\left[x_j(n)\right] = \min_{\substack{i=1,2,\dots,k \\ i\neq j}} \left\{ \frac{1}{N_i - 1}\sum_{\substack{l=1 \\ l\neq n}}^{N_i} D\left[x_j(n)\,, x_i(l)\right] \right\} \tag{3.8}$$

is the minimum average dissimilarity between the point and all points in other clusters. A large value indicates that the point would not fit well into another cluster. The silhouette for point $x_j(n)$ is given by

$$s\left[x_j(n)\right] = \frac{b\left[x_j(n)\right] - a\left[x_j(n)\right]}{\max\left\{a\left[x_j(n)\right], b\left[x_j(n)\right]\right\}} \tag{3.9}$$

and has the range $-1 \leq s\left[x_j(n)\right] \leq 1$. The average of $s\left[x_j(n)\right]$ across all points, denoted as $\hat{s}$, provides an indication of how well the data was clustered, with values near one indicating good clustering (Rousseeuw). Thus, the partition resulting in the largest $\hat{s}$ across all values of k and all trials is selected as the clustering result. Note that (3.9) can only be computed for $k \geq 2$. If $\hat{s} < \gamma$ for some user-selected constant γ, then all the data is assumed to belong to a single cluster.

In the following two sections, *k*-means clustering and the silhouette metric are leveraged to estimate ambient spectra and identify dominant electromechanical modes from the estimates. The proposed methods could be implemented with other clustering methods. Initial results do not indicate that more advanced methods are needed, but advantages of other methods could be a topic for future work.

3.2 The Robust Modified Daniell-Welch Spectral Estimator

Recall that the median-filtered simple periodograms $\hat{\phi}_s'(\omega_r)$ given by (3.4) are calculated from overlapping segments of data. Thus, they will take on similar values under ambient and FO conditions (recall that spectral content from FOs is suppressed by the median filter). However, if a ringdown is present in the data, the periodograms from segments containing the ringdown will have significantly larger values over a relatively wide frequency range. The RMDW estimator operates by examining the $\hat{\phi}_s'(\omega_r)$ terms in (3.5) and removing those with abnormally large values to mitigate the effects of ringdowns.

To enable this ability, the algorithm attempts to partition the $\hat{\phi}_s'(\omega_r)$ into two clusters. The clustering involves S points, each corresponding to one of the median-filtered simple periodograms. Let R denote the number of frequency bins ω_r under consideration. Then each point is in R-dimensional space.

Within the RMDW algorithm, a non-Euclidean distance is utilized. To motivate its use, consider the toy example presented in Figure 3-1 and Table 3-1. The figure contains three median-filtered simple periodograms and the table contains distances between them (recall that each periodogram is viewed as a point). Though $\hat{\phi}_3'(\omega_r)$ tends to be higher, perhaps indicating the presence of a ringdown, the Euclidean distances are all similar. As an alternative, consider the "directional" distance defined for vectors $\bar{v}$ and $\bar{q}$ as

$$D = \left| \sum_{r=0}^{R-1} \bar{v}(r) - \bar{q}(r) \right| \tag{3.10}$$

The term *directional* is used to indicate that in a plot such as Figure 3-1, the measure accentuates consistently positive or negative differences across dimensions, i.e., frequency bins. Indeed, the directional distances in Table 3-1 reveal that $\hat{\phi}_3'(\omega_r)$ tends to be larger than the other periodograms. Referring to (3.10) as a distance is a misnomer, but it points to the value's use in place of the more typical Euclidean distance.

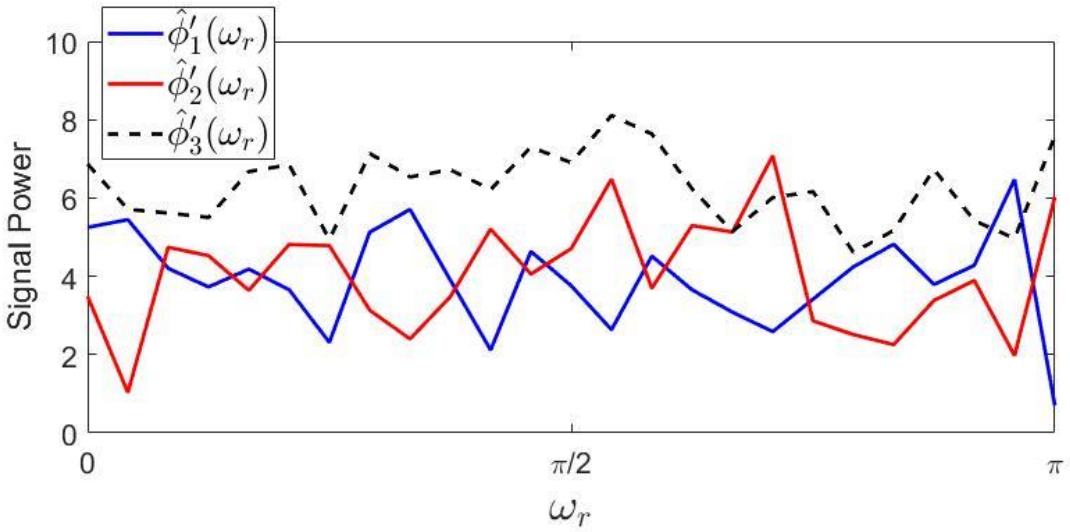


Figure 3-1. Examples of median-filtered simple periodograms with distances listed in Table 3-1.

Table 3-1. Distances between the example median-filtered simple periodograms in Figure 3-1.

Pair	Euclidean	Directional
$\hat{\phi}'_1(\omega_r) \Leftrightarrow \hat{\phi}'_2(\omega_r)$	12.6	2.4
$\hat{\phi}'_1(\omega_r) \Leftrightarrow \hat{\phi}'_3(\omega_r)$	14.4	58.5
$\hat{\phi}'_2(\omega_r) \Leftrightarrow \hat{\phi}'_3(\omega_r)$	13.3	56.2

Using the directional distance, the $\hat{\phi}'_s(\omega_r)$ are partitioned into two clusters. Let Φ_C denote the length S_C set of common $\hat{\phi}'_s(\omega_r)$ generated from ambient and FO data, and let Φ_R denote the length S_R set of $\hat{\phi}'_s(\omega_r)$ with large values indicating the presence of a ringdown. The associated spectral estimates that follow from (3.5) are then

$$\hat{\phi}_C(\omega_r) = \frac{1}{Q \times S_C} \sum_{\phi \in \Phi_C} \phi(\omega_r) \tag{3.11}$$

$$\hat{\phi}_R(\omega_r) = \frac{1}{Q \times S_R} \sum_{\phi \in \Phi_R} \phi(\omega_r). \tag{3.12}$$

After clustering, a set of requirements is evaluated to ensure that the clustering reflects the presence of a ringdown. The heuristic set of requirements used to produce the results in this paper follow:

1. $\hat{s} > 0.8$
2. $S_R < \frac{S_C}{3}$
3. $\hat{\phi}_R(\omega_r) > \hat{\phi}_{MDW}(\omega_r)$ at more than $\frac{R}{2}$ frequency bins
4. $\hat{\phi}_C(\omega_r) > \hat{\phi}_{MDW}(\omega_r)$ at more than $\frac{R}{2}$ frequency bins

Requirement 1 helps ensure that two distinct clusters are present: one for ambient/FO conditions and one for ringdown conditions. The second requirement reflects that ringdowns are relatively short compared to data lengths appropriate for the algorithm, so Φ_R should be a significantly smaller set than Φ_C. Requirements 3 and 4 reflect that ringdowns typically increase signal power over a wide frequency range and reinforce that the clusters should be clearly separated from each other when viewed as spectra. If all requirements hold, then the RMDW estimate is given by

$$\hat{\phi}_{RMDW}(\omega_r) = \hat{\phi}_C(\omega_r).$$

The time-domain data segments associated with Φ_R can be reviewed for the presence of a system event, which serves as a useful biproduct of the RMDW algorithm. If any of the requirements fail, then

$$\hat{\phi}_{RMDW}(\omega_r) = \hat{\phi}_{MDW}(\omega_r)$$

In each case an estimate of the ambient spectra is produced, regardless of the presence of FOs or a ringdown in the data. This characteristic makes the RMDW algorithm useful for identifying dominant electromechanical modes, as described in the next section.

3.3 Dominant Mode Identification

Dominant electromechanical modes are apparent in ambient spectra as broad peaks. The peaks in the true spectra are centered at the frequencies of the modes. For example, the model used to generate simulation data, which will be described more fully in Section 3.4.1, has modes with frequencies listed in Table 3-2. Note that these modal frequencies correspond to peaks in the true signal spectrum depicted in Figure 3-2. For a given signal, the largest peak corresponds to the most observable mode in the signal. Modes that are highly observable at many points in the system tend to be some of the most important for consideration. In the approach proposed in this section, these dominant system modes are identified based on spectral estimates from the RMDW algorithm.

Table 3-2. Electromechanical modes of the miniWECC model used to generate simulation data.

Mode	Frequency (Hz)	Damping (%)
1	0.22	5.0
2	0.37	6.0
3	0.51	8.7
4	0.69	5.8

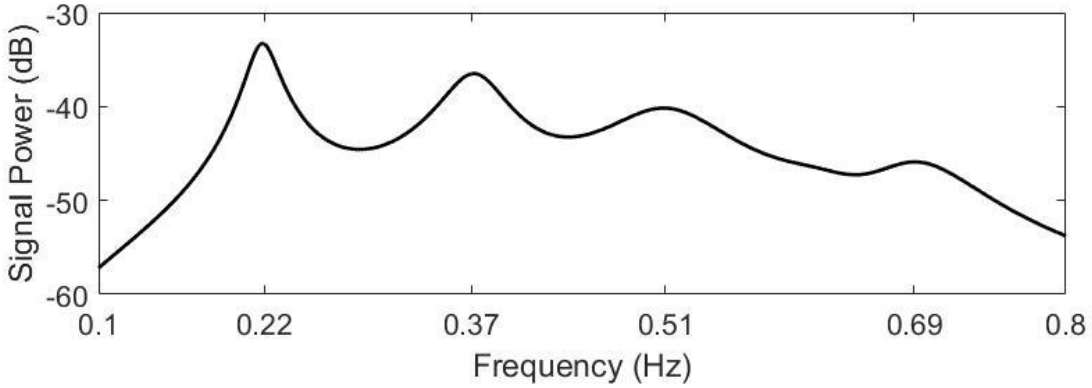


Figure 3-2. Spectrum of the signal used for simulation studies under ambient conditions. Note that peaks correspond to the modes in Table 3-2.

Spectral estimates from the RMDW algorithm are based on a relatively long window of data, say 30 minutes, and updated on some regular interval, say every minute. For each spectral estimate, the frequency and signal power corresponding to the highest point in the spectrum is recorded. At the end of a day, 1440 such records will have been collected. Due to system configuration and operation changes, more than one mode may be captured. Using the simulation model as an example, records could be split between estimates near 0.22 Hz and 0.37 Hz. To identify multiple dominant modes observable in a signal, the *k*-means method is again employed.

At the end of each day, the recorded frequency estimates are clustered, with each cluster corresponding to a dominant mode. In this application, the typical Euclidean distance is used. Without prior knowledge of the number of dominant modes observable in a signal, it is necessary to perform the clustering for several values of k. Multiple trials are also considered for each k, as described in Section 3.1.2. For the results presented in Section 3.4, five trials were conducted for $k = 2,3,4$. If the average silhouette $\hat{s} < 0.9$, a single dominant mode was assumed. To account for the random nature of the frequency estimates, the clustering is performed a second time, after removing outliers from each initial cluster.

At the end of the clustering procedure, the dominance of mode j is given by

$$Dom_j = \frac{N_j}{\sum_{i=1}^{k} N_i}. \tag{3.13}$$

The range of frequency estimates in the cluster reflects the variation in the mode throughout the day. Estimate error contributes to the range as well. The range of signal power values associated with the frequencies also indicates the relative dominance of different modes that are observable in the signal.

The results from the dominant mode identification algorithm could be used to gain initial high-level information about a system's modes, to prepare for the deployment of a mode meter, or to relate changes in a system's dynamics to parameters such as season, weather, line flow, and system configuration. In the

following section, experimental results are presented to demonstrate the viability of the proposed methods.

3.4 Experimental Results

To evaluate the performance of the methods proposed in Sections 3.2 and 3.3, experiments were conducted with simulated and measured power system data. In each test, data was available at 60 samples per second. Spectral estimates were generated based on 30 minutes of data broken up into 200 second windows with 100 seconds of overlap. The median filter had an order of 7, approximately 3 times the main lobe width of the rectangular window (see (Follum and Tuffner, A multi-channel method for detecting periodic forced oscillations in power systems) for discussion). Frequencies between 0.1 Hz and 1 Hz were considered to cover the typical range of electromechanical modes.

The signals selected for analysis were differences between voltage angles in different parts of the systems. Subtracting one voltage angle from another tends to suppress common signal components and accentuate oscillatory interactions between the areas. A first-order difference filter was applied to convert the angle signals to measures of frequency deviation about nominal with units of Hz.

3.4.1 Simulation Results

Simulation data were generated using the miniWECC, a simplified dynamic model of the WECC system (Trudnowski, Kosterev and Undrill). It contains 34 generators, 19 load buses, 115 AC lines, 2 DC lines, and a 1400 MW dynamic brake representative of the installation at Bonneville Power Administration's Chief Joseph substation. For data generation, the model was linearized about an operating point. Random modulation was applied at each load bus to excite the system's dynamics and produce ambient noise. A FO was induced by injecting a square wave with a 0.6 Hz fundamental frequency at a generator bus to model a stable limit cycle. The FO was present throughout the 30-minute simulation and was too small to be apparent in time-domain data. A ringdown was initiated at the simulation's 15th minute by inserting the dynamic brake for 0.5 seconds. The resulting ringdown is depicted in Figure 3-3.

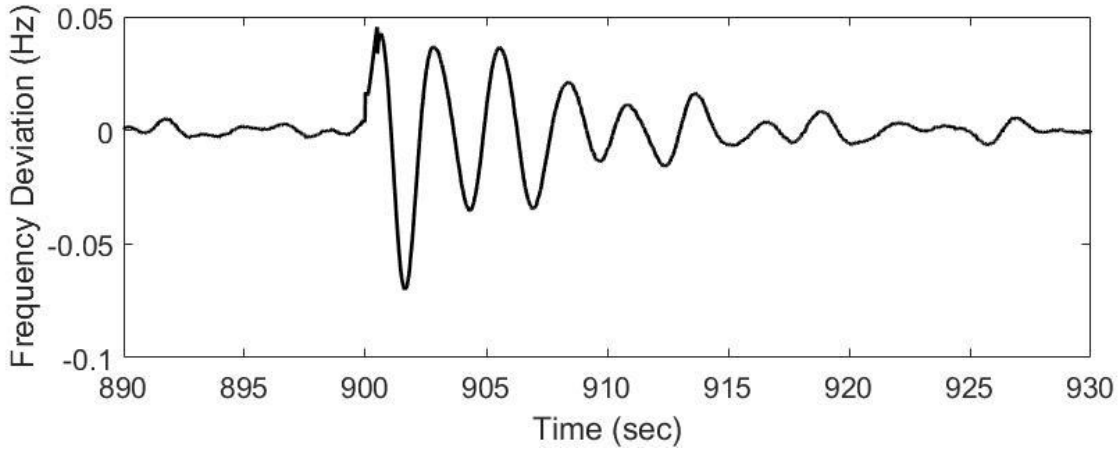


Figure 3-3 . Example simulation data including a ringdown.

Results from the classic Welch, previously developed MDW, and new RMDW spectral estimators are presented in Figure 3-4. Because a model was used, the true power spectral density (PSD) of the system's ambient noise is available for comparison. The classic Welch periodogram does not account for the presence of the ringdown or the FO. The ringdown adds significant energy to the signal, causing the Welch estimate to be much higher than the PSD at all frequencies. Because the brake insertion primarily excited the 0.37 Hz mode, its peak exceeds that of the 0.22 Hz mode. If the Welch estimate was used to identify dominant modes as in Section 3.3, the 0.22 Hz mode would not be listed as dominant for the 30 minutes that the ringdown remained in the analysis window. Further, the Welch periodogram contains a large peak due to the FO at 0.6 Hz, which could distort the dominant mode identification results for an even longer period.

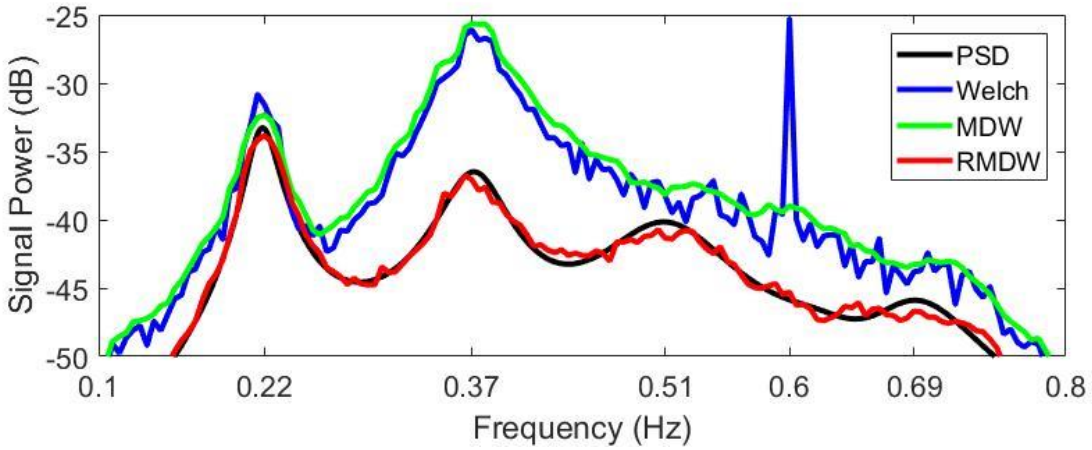


Figure 3-4 . Spectral estimates from simulated data. Note that the RMDW estimate most closely matches the true ambient spectrum.

Next, consider the MDW estimate. The MDW estimator was designed to suppress spectral content due to FOs while retaining the effects of ringdowns. The ringdown causes the MDW estimate to be higher than the PSD across the frequency range of interest, just like the Welch periodogram. However, the MDW estimator removes the FO peak at 0.6 Hz, as expected.

Finally, consider the RMDW estimate. By automatically removing data segments containing the ringdown and suppressing the FO's spectral content, the RMDW estimate closely matches the true PSD. Results from the dominant mode identification process described in Section 3.3 implemented with the RMDW estimator would reveal that under ambient conditions the 0.22 Hz mode is dominant in the selected signal. This information could be useful for implementing a mode meter or noticing changes in the system's dynamics.

3.4.2 Measurement Results

To verify performance under real-world conditions, the proposed methods were applied to synchrophasor measurements from the WECC system. The results reported here were generated from six days' worth of

data collected over seven calendar days in October 2017. From the synchrophasor measurements, 14 voltage angle pairs were constructed to examine interactions between different areas of the system.

The RMDW estimator was used to implement the dominant mode identification algorithm described in Section 3.3. Results for the seven calendar days are presented in Table 3-3. Note that for the first four days, the dominant mode for most angle pairs was near 0.33 Hz. The dominance metric in (3.13) was near unity for most of the channels. For the latter three days, the dominant mode transitioned to near 0.25 Hz, again with dominance metrics near unity. Both of these frequencies are well known to correspond to modes in the WECC system. Further investigation by the Bonneville Power Administration revealed that the change in mode observability coincided with a major transmission line being taken out of service.

Table 3-3. Estimates of dominant mode frequencies across seven days. Note the distinction between days 1-4 and days 5-7.

	Frequency (Hz)						
Pair	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
1	0.655	0.350	0.355	0.650	0.345	0.260	0.255
2	0.335	0.330	0.325	0.330	0.245	0.260	0.255
3	0.335	0.330	0.320	0.325	0.245	0.260	0.260
4	0.335	0.330	0.320	0.325	0.245	0.260	0.255
5	0.335	0.330	0.320	0.325	0.245	0.260	0.255
6	0.750	0.750	0.740	0.815	0.800	0.820	0.855
7	0.335	0.330	0.320	0.325	0.255	0.268	0.265
8	0.335	0.325	0.320	0.325	0.250	0.265	0.260
9	0.335	0.330	0.320	0.325	0.255	0.265	0.265
10	0.335	0.330	0.320	0.325	0.250	0.265	0.260
11	0.335	0.330	0.320	0.325	0.250	0.265	0.260
12	0.340	0.330	0.320	0.325	0.250	0.265	0.260
13	0.340	0.330	0.325	0.325	0.255	0.265	0.265
14	0.350	0.355	0.390	0.380	0.420	0.43	0.405

Recall that the RMDW provides accurate ambient spectral estimates by detecting and removing event data from consideration. During the analysis, events with widespread impact (detected in at least 10 of the 14 signals) were recorded for further review. Six such events were detected, five of which were clearly visible as ringdowns in time-domain data. An example of one of the events is presented in Figure 3-5. For the signal highlighted in red, spectral estimates from 30 minutes surrounding and including the event are presented in Figure 3-6. Highest peaks are indicated with dots. Due to the ringdown and a FO, the highest peaks in the Welch and MDW estimates are near 0.75 Hz. By removing the ringdown and suppressing the FO, the RMDW identifies the mode near 0.25 Hz as the most dominant under ambient conditions.

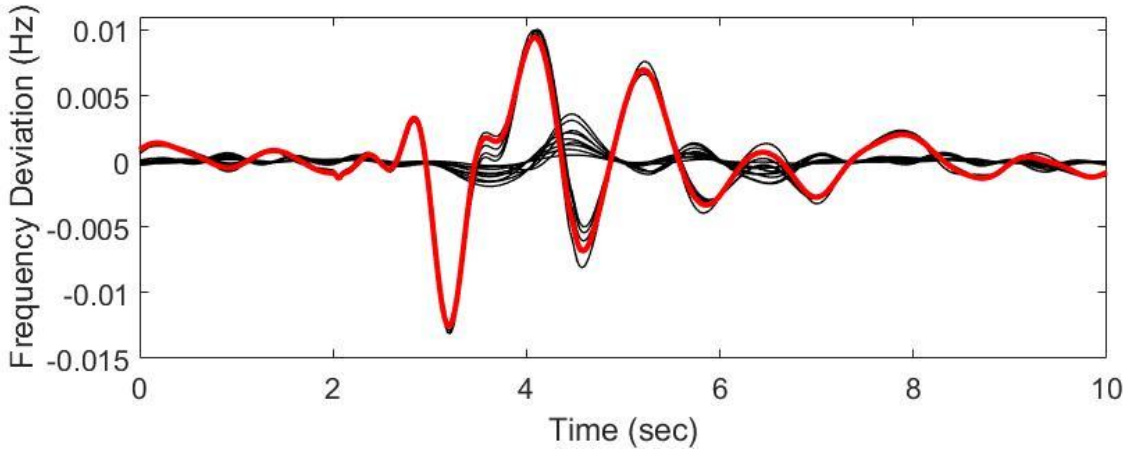


Figure 3-5. Ringdown event detected by the RMDW estimator. Spectral estimates for the signal highlighted in red are presented in Figure 3-6.

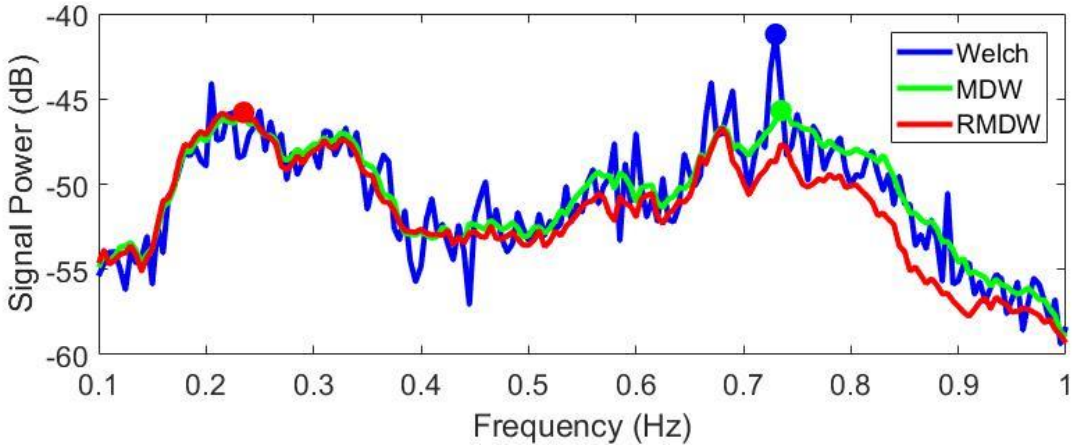


Figure 3-6. Spectral estimates for the signal in Figure 3-5 highlighted in red. Highest peaks in each estimate are indicated with dots.

The results presented in this section demonstrate that the proposed methods can be practically applied with power system measurements. The RMDW estimator provides accurate estimates of ambient spectra that can be used to identify the most observable system modes in signals. Transitions in modal observability and significant system events can be identified with the method, making it a useful companion to more traditional event detection algorithms.

Estimates of ambient spectra based on synchrophasor measurements provide a wealth of information about the system's electromechanical modes. Dominant modes can be identified and changes in their characteristics can be monitored. Along with ambient noise, measurements contain forced and ringdown oscillations that perturb conventional spectral estimators, making it difficult to consistently track the ambient spectrum. The RMDW spectral estimator proposed in this paper solves the problem by automatically distinguishing between ambient and transient conditions and suppressing spectral content from FOs. The new method can be used to identify dominant system modes and to detect system events.

4.0 Classification

4.1 Frequency Events

A list of 199 frequency events from October 2016 to May 2017 was obtained from Bonneville Power Administration (BPA). These events were detected by BPA using existing tools. There are two main types of frequency events consisting of active power events and faults. The frequency event list includes the number of sites that were affected by the event as well as the largest magnitude drop detected, which indicates the severity of the event. Most of the listed events can be observed visually in the time domain. Figures 4-1 to 4-3 show examples of a normal minute, a minute with an active power event, and a minute with a fault, respectively.

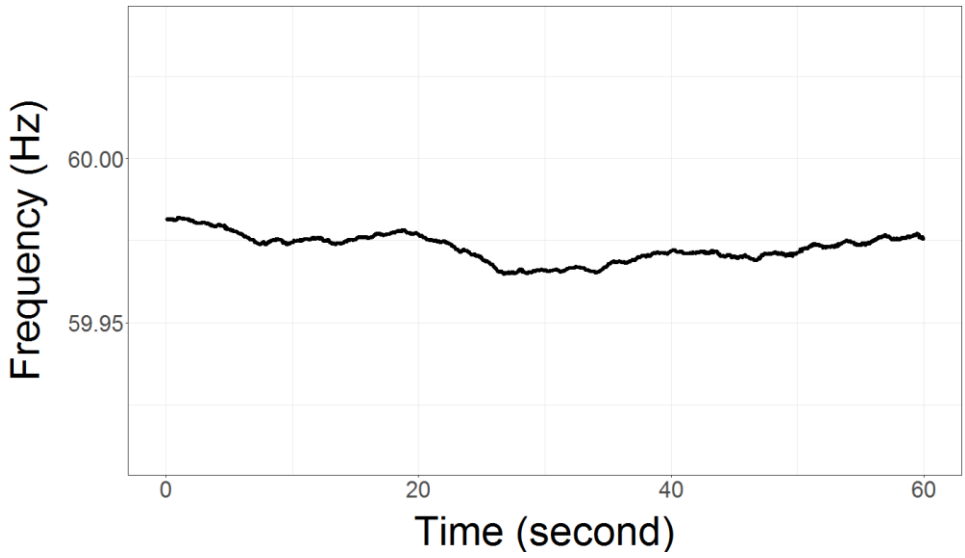


Fig. 4-1. Frequency over a minute for an example under normal conditions.

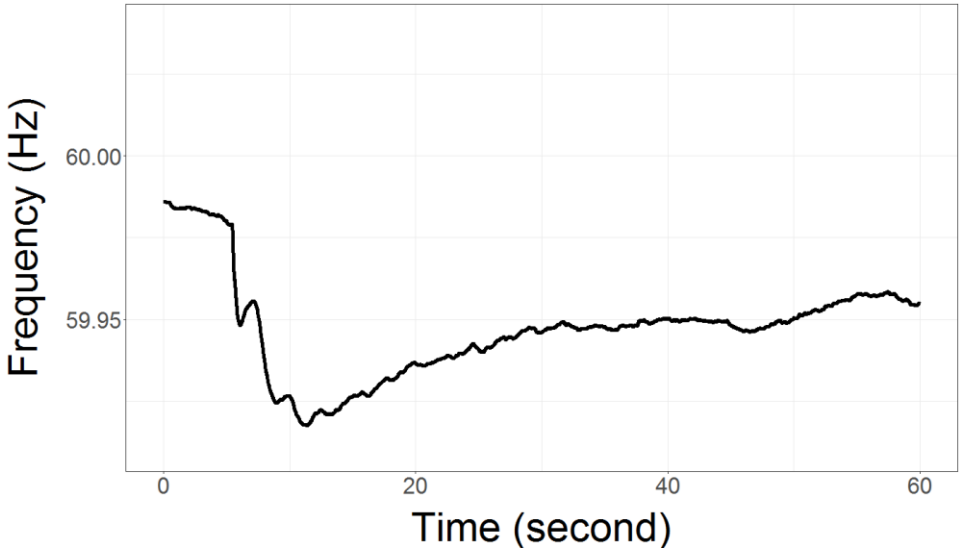


Fig. 4-2. Frequency over a minute for an example with an active power event.

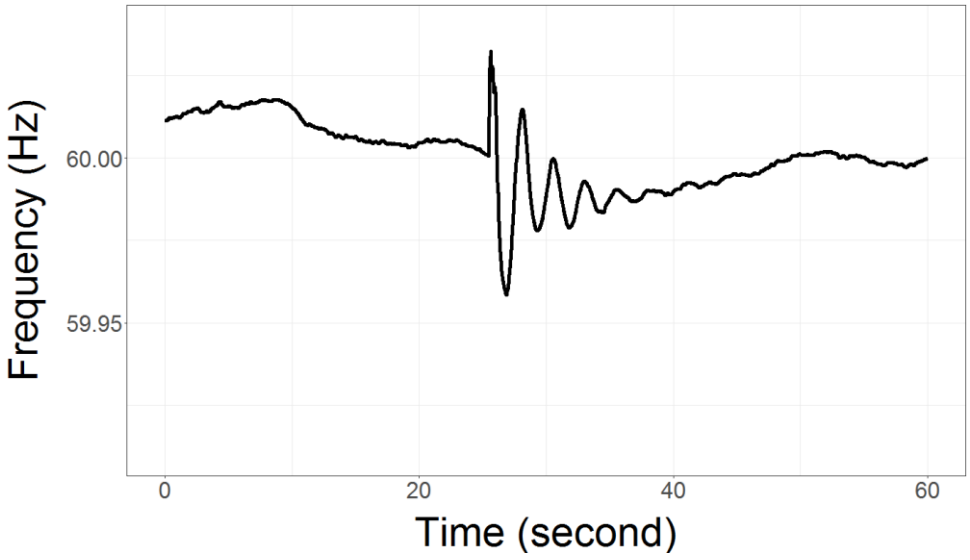


Fig. 4-3. Frequency over a minute for an example with a fault.

Exploratory data analysis shows that all of the frequency signals are highly correlated in our dataset. The correlation and cross-correlation are usually above 99.99%. Therefore, choosing one of them as the primary signal is adequate for the main task in this study. The data for several of the 199 events are completely missing. Therefore, such events were not included in the study. Also, several events are not visible for the recorded time. This might be due to human mistakes during the time recording process. There are some small events that only affect 3 or 4 sites as opposed to most events that affect more than 40 sites. Such events are usually small faults. They are not included in the training data, because the main task of the trained algorithm is to detect large events. However, this study also includes a complementary algorithm that detects small faults and tests whether the final model can capture such small events. This idea is discussed in Section 4.2.

Other main data wrangling tasks include signal selection, missing data imputation, and window size selection. One frequency signal is chosen as the primary signal for the study. An important question is whether incorporating more signals other than the frequency signal can improve the performance in the machine learning algorithms. The computation time will definitely increase, but adding other types of signals could also boost accuracy. There are five major categories other than frequency, namely, the current magnitude, voltage magnitude, paired phase angle difference, active power, and reactive power. Figure 4-4 is an example minute of data showing an active power event in six different categories of signals. The first signal from each category is chosen and added to the training dataset to see if the performance of the machine learning algorithms based on six signals will be better than the one based solely on one frequency signal. There are missing data within some minutes. For the events with little missing data within the minute, the previous value is imputed to the missing spot.

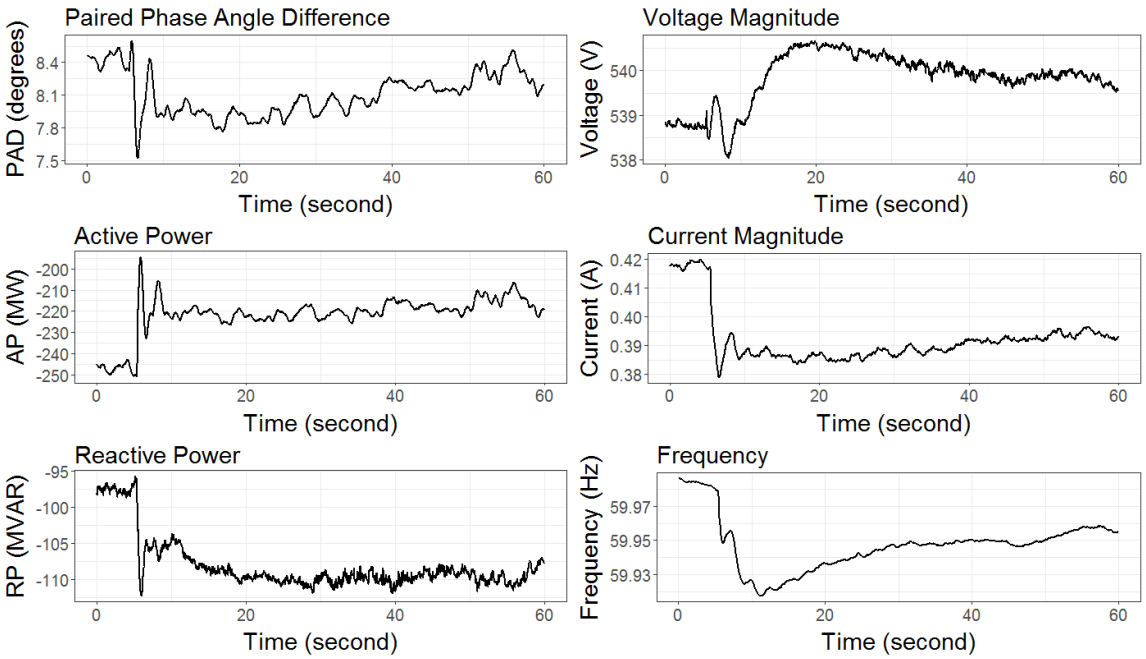


Fig. 4-4. Example of an active power event in six categories of signals.

Since all the recorded frequency events have duration of less than one minute, a one-minute window size is chosen for the training data. All the training data are formed according to the following rule. If the event happens after 30 seconds within a minute, move the window 30 seconds forward. In this way, all the events are completely present in the training data. This is especially meaningful for the events that occurred after 50 seconds within one minute. At the testing stage, when screening the model through the unseen days, the events can happen anywhere within the one-minute window. It is interesting to see if the model can flag events that happen very late within the one-minute window. This issue is discussed in Section 4.3. There are 164 events with good quality data. A total of 336 normal minutes with good data are randomly selected throughout the dataset. These 500 minutes of data form the dataset for selecting the best machine learning algorithm.

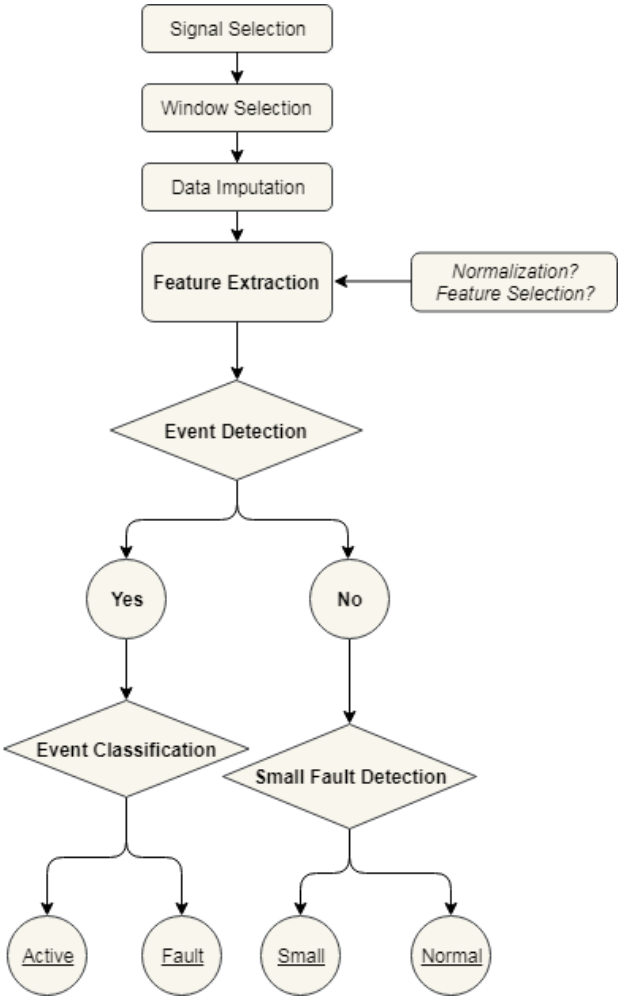


Fig. 4-5. Flowchart of classifying an incoming minute.

4.2 Methodology

Figure 4-5 shows the methodology in this study for detecting and classifying events in a new incoming minute. Signal selection, window selection, and data imputation are a part of data wrangling. After that, feature extraction is a vital component before feeding the extracted features to the machine learning algorithms. Whether to select and normalize the features are also important topics discussed in this section. If an event is detected, then event classification algorithm will be used to determine the type of event just occurred. If no event is detected by the main machine learning algorithm, the small fault detection algorithm will be used to see if there is a small fault within the minute. If not, this minute will be classified as normal.

For time series data mining, an important issue is feature extraction. The feature extraction method should be simple enough to avoid massive computational time, but still capture adequate information to perform accurate prediction. Three methods of feature extraction are investigated in this study. The first method is a simple 5-statistics summary approach (mean, maximum, minimum, median, and variance).

The second method is the 16 signature elements discussed in Section 2.2 (Amidan and Ferryman). The main idea of this method is to fit a quadratic regression line to every second of the data of the form:

$$a + bx + cx^2 + d, \qquad [4.1]$$

where a, b, and c are the estimated coefficients of the regression fit and d is the Mean Square Error (MSE) of the fit. For the whole minute of data, there are 60 a's, b's, c's, and d's, for which the mean, maximum, minimum, and standard deviation are calculated. Therefore, there are 4 statistics for each coefficient and the MSE, thereby forming the 16 (4×4) signature elements. The third method is based on the same idea as the second one, but in addition to the mean, maximum, minimum, and standard deviation, the median, the skewness and kurtosis are also calculated, resulting in 28 (4×7) signature elements.

Seven machine learning algorithms are used in this study, namely Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (ANN), Adaptive Boosting (AdaBoost), Gradient Boosting Machine (GBM), and Generalized Linear Model (GLM) (Hastie, Tibshirani, and Friedman). Among these algorithms, DT, RF, AdaBoost, and GBM are decision tree-based. During the model training period, 10-fold cross-validation repeated 5 times is used for parameter tuning and model evaluation.

A practical question before training the algorithms is whether the extracted features should be normalized:

$$f^* = \frac{f - mean(f)}{sd(f)}. \qquad [4.2]$$

Normalization might affect the performance of some of the machine learning algorithms, but not all of them. Tree-based algorithms are usually not affected while ANN often needs the data to be normalized.

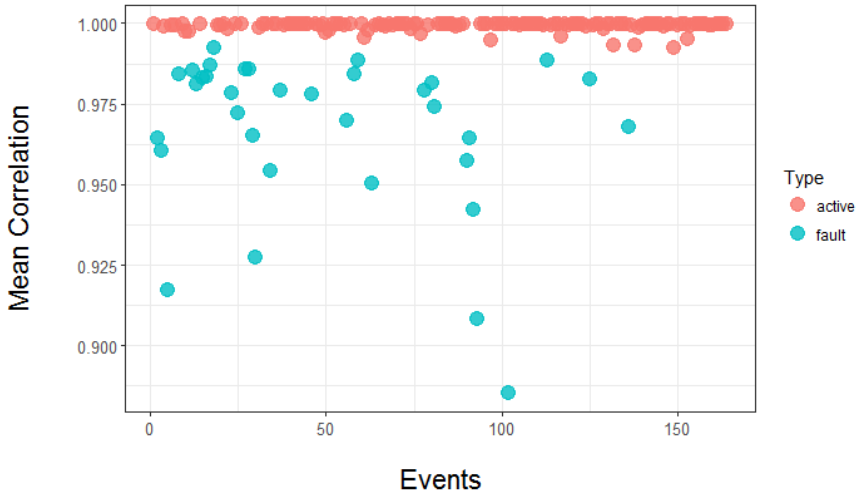


Fig. 4-6. Mean of correlations between paired frequencies for all the events.

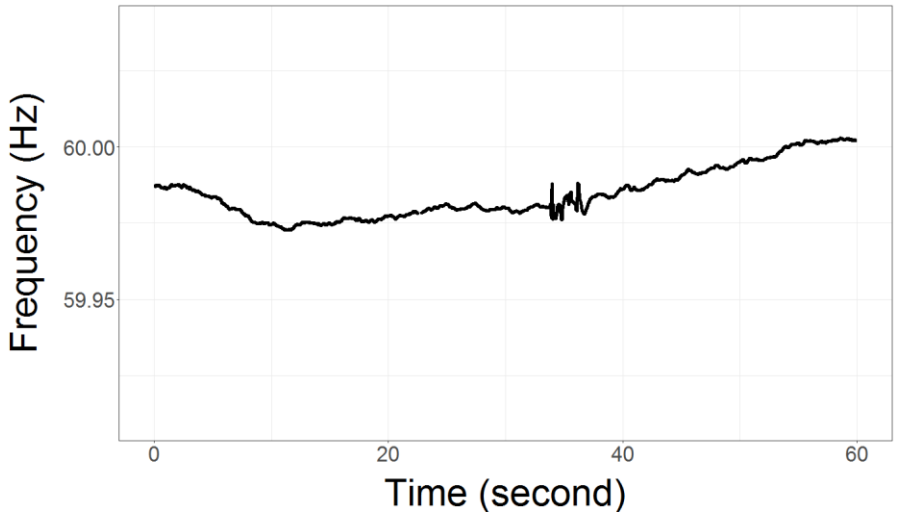


Fig. 4-7. Frequency over a minute for an example with a small fault.

Another question pertains to whether the extracted features are all useful or whether the number of predictors can be reduced using dimensionality reduction techniques like Principle Component Analysis (PCA) (Abdi and Williams) or Variable Importance provided by the machine learning algorithms. Variable importance is usually calculated by a permutation test based on the idea that if a feature is important, then rearranging the values of that feature randomly will decrease the performance of the algorithm (Strobl and Zeileis). The impact of variable screening is discussed in Section 4.3.

When training the algorithm, it is important to pay attention to both sensitivity and specificity. Sensitivity refers to the probability of correctly identifying an event given there is actually an event. Specificity is the probability of recognizing there is no event given there is actually no event. The objective is to have high detection rate, or high sensitivity, and few false alarms, or high specificity.

After detection of an event, an important aspect of situational awareness is to know the type of event that just happened. The second stage of the proposed approach is to classify events. Among the training data, around 20% of events are faults and the other 80% of events are active power events. Therefore, all the algorithms will likely lean towards classifying the events to be active power events based upon the original features. Exploratory data analysis shows that correlations between frequency signals are generally lower when there is a fault than when there is an active power event. This can be seen in Figure 4-6. Therefore, the mean of all the correlations between paired frequencies is added as a new feature in the second stage. Sensitivity in the second stage refers to the probability of correctly identifying an active power event given the event is actually an active power event. Specificity here is the probability of identifying a fault given that the event is actually a fault.

Although the resulting machine learning algorithm is very good at detecting and classifying most events (shown in Section 4.3), it is not good at detecting small faults. Figure 4-7 shows an example of a small fault event. As can be seen, most parts of the minute look like a normal minute. Therefore, a new

algorithm is developed to be complementary to the main algorithm as it is specialized at detecting small faults. This algorithm has three steps:

1. Calculate the first order differences of the data within the window;

2. Conduct a *k*-means clustering on the first order differences with k (the number of clusters) set to be 2;

3. Calculate the mean silhouette value (always between 0 and 1) of the resulting clustering.

The silhouette value is calculated as (Rousseeuw):

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),\ b(i)\}}, \qquad [4.3]$$

where $a(i)$ stands for the mean distance of datum i with all other data within the same cluster and $b(i)$ is the lowest mean distance of i to all points in any other cluster. A high value of the mean silhouette of all points, usually greater than 0.9, indicates the appearance of a small fault event. This approach helps to identify small faults, because faults usually cause the frequency to oscillate, which can be well captured by high values in the first order differences. Such high values form an obvious outlier group that is different from the near-zero values of normal frequency first order differences. Therefore, if there is proof that there are two groups well separated within the first order differences, there is a high probability of the appearance of a fault. A mean silhouette value greater than 0.9 shows evidence that there are two different groups in the clustering result.

4.3 Results

In this section, the best feature extraction method is identified. The results from model training and testing are examined to select the best machine learning algorithm for each stage. Then, separate testing on a whole month of unseen data is evaluated. Lastly, further discussions regarding variable importance and event precursor are provided.

For the three methods of feature extraction, the 16 signature elements approach turns out to be the best regarding detection and classification accuracy. The 5-statistics summary approach is simple and computationally fast, but it generally leads to around 5% lower correctness in sensitivity except for SVM and GLM. For GLM, the 5-statistics summary obtains a decent result of 97.55% sensitivity and 99.11% specificity. The SVM result is also quite good, having a 96.20% sensitivity and 99.05% specificity. However, these results are still not as good as the ones obtained based on the 16 signature elements. The 28 signature elements approach turns out to be the worst regarding event detection, with generally a 10% decrease in sensitivity compared to the 16 signature elements. This is an interesting finding because the original 16 signature elements are still included in the 28 signature elements. This indicates that the additional three statistics are adding more noise rather than information regarding the state of the grid. This also implies that not every feature is helpful for our task, so dimensionality reduction is useful for both speeding up the algorithm and increasing accuracy.

Table 4-1. Event Detection Results and Computational Time (in Seconds).

Algorithm	Sens	Spec	Accu	Training	Testing
AdaB	98.67	99.40	99.16	473.0s	0.50s
ANN	81.66	99.41	93.59	13.3s	0.14s
DT	94.90	97.92	96.93	3.3s	0.18s
GBM	98.18	99.53	99.08	9.0s	0.14s
GLM	97.91	99.35	98.88	12.6s	0.14s
RF	96.96	99.05	98.37	21.3s	0.12s
SVM	97.29	98.51	98.11	9.2s	0.14s

Incorporating five different signals other than frequency does not help the machine learning algorithms perform better. In fact, all the machine learning algorithms perform worse than when just using one frequency signal. On average, there is an over 2% decrease in sensitivity in the first stage. Again, this shows that adding more features does not necessarily correspond to an increase in accuracy. The need to normalize the features is also investigated. Most algorithms are not affected by normalization, except for ANN. ANN results have been greatly improved after normalization, even appearing to be the best algorithm, achieving 99.25% sensitivity and 99.07% accuracy. Further testing on unseen days reveals that this final model is over-trained, which is a common problem with ANN.

The optimal values of the hyper-parameters were determined by a grid search. Event detection results are shown in Table 4-1. Computational time is obtained using a single desktop. It is worth mentioning that the training time is not very important since it is performed off-line on the historical data. Testing time is very important if this approach is to be used in a real-time on-line setting. The priority is sensitivity since the main objective is to identify system events. AdaBoost achieved the best sensitivity, but it takes a long time to train. GBM is the second best regarding sensitivity, and it only takes 9 seconds to train. On the other hand, specificity is also important because too many false alarms are also not ideal. Even a 1% decrease in specificity matters because that corresponds to about $1440\times0.01=14.4$ more false alarms on average per day. Therefore, over 99% specificity is desirable. The best specificity is 99.53% obtained by GBM, which corresponds to 6.8 false alarms per day. Regarding computational time, AdaBoost takes much longer than the other algorithms to train. The testing time includes feature extraction time, so theoretically it is the time needed to detect events from real-time PMU data. The testing of one incoming minute can be done within 0.2 seconds for most algorithms, which is good compared to the present technique using Supervisory Control and Data Acquisition (SCADA). SCADA measurements come in every 4 seconds and it takes minutes (sometimes more than 10 minutes) for the state estimation algorithm to run and to inform the system operator that an event might have happened. If such time can be reduced to seconds by machine learning applications on PMU data, situational awareness will be improved for operators and a near real-time wide area view of the power grid can be achieved.

Table 4-2. Event Classification Results and Computational Time (in Seconds).

Algorithm	Sens	Spec	Accu	Training	Testing
AdaB	99.22	97.17	98.79	6.3s	0.24s
ANN	100.0	48.00	89.09	9.3s	0.26s
DT	99.22	97.17	98.79	2.5s	0.31s
GBM	100.0	97.17	99.41	7.0s	0.29s
GLM	100.0	99.50	99.90	5.9s	0.26s
RF	99.85	98.00	99.46	6.9s	0.25s
SVM	97.33	98.50	97.58	5.4s	0.25s

Event classification results are shown in Table 4-2 for the second stage of model training. Since there are more active power events in the training data, it is easier for the algorithms to identify them. Some of the algorithms achieve 100% at classifying active power events into the correct type. The harder part is specificity, which represents the probability of correctly classifying a fault. GLM achieved the best specificity, which is 99.50%. It also has the best overall accuracy. Regarding computational time, it is interesting to note that by adding the "mean correlation" feature, AdaBoost returns to normal computational time. The testing time includes the time to calculate the mean correlation for twelve frequency signals. Most training can be done within a few seconds, and most testing of one event can be done within 0.3 seconds. Considering sensitivity, specificity, and computational time, GBM is chosen for the first stage and GLM is chosen for the second stage.

Separate testing is done on the May 2017 data. GBM and GLM models are trained for event detection and classification without any of the May 2017 data. Such models are used to detect and classify events for all the minutes of May 2017. There are 41 recorded events in May 2017, and GBM flags 37 of them. The 4 missed events are all small faults that affect only 3 or 4 sites rather than the over 40 sites for most events. The small fault event detection algorithm is used to screen the whole month of data and it detects all the 4 missed events and some other small potential faults that are not listed. Among the 37 flagged events, 36 of them are classified as the right type. There is one fault misclassified as an active power event. To conclude, the algorithm flags all the recorded events in May and classifies all of them, but one, into the correct types. For each event, the time needed to detect and classify it is less than one second. The same testing using other algorithms is also investigated. AdaBoost and GLM are chosen to conduct event detection for May 2017, because they are the best two algorithms, besides GBM, regarding performance in the training period. AdaBoost misses one listed event, and GLM misses four listed events, both with the help of the small event detection algorithm in the first stage. For event classification, RF and GBM are chosen to test the May 2017 data. They both correctly classify 34 out of 37 events, which is worse than GLM.

There are on average 10 false alarm minutes every day for this testing. One concern about this approach is whether it can flag events that happened after 50 second within the one minute window. There are 6 such events in May 2017. The algorithm flags all 6 of them. For 4 of these 6 events, the algorithm also flags the next minutes since the major parts of the events are in the next minutes. Regarding dimensionality reduction, PCA does not do well at reducing the number of predictors. Even after accounting for 97% of the variation, the principle components can only produce around 90% accuracy in the first stage. On the other side, variable importance is a good option for dimensionality reduction. When removing all variables other than the 4 most important variables, most algorithms produce very similar results compared to using all 16 features.

Most algorithms agree on which variables are most important for event detection. Variables 4 (standard deviation of a), 5 (minimum of b), 8 (standard deviation of b), 9 (minimum of c) appear in 5 algorithms to be the top 4 variables. For event classification, most algorithms agree that "Mean Correlation" is the most important variable regarding algorithm performance. SVM turns out to be the best in overall accuracy for event detection after variable selection, even better than GBM with 16 variables. However, SVM is worse than GBM regarding specificity. According to the results, SVM (with the 4 most important variables) is chosen to test the May 2017 data. It detected 39 out of 41 events, including 2 small faults. But it has on average 23 false alarms per day, which is twice the number of false alarms of GBM with 16 variables. For the classification, three algorithms are tried with the top 4 variables. None of them perform as well as GLM with 16 variables. Training time is generally shorter when using 4 variables while testing time is not affected much.

Testing on a whole month of data with the small event detection algorithm also reveals possible event precursors. It could be important to be able to detect small events that precede big events. These small

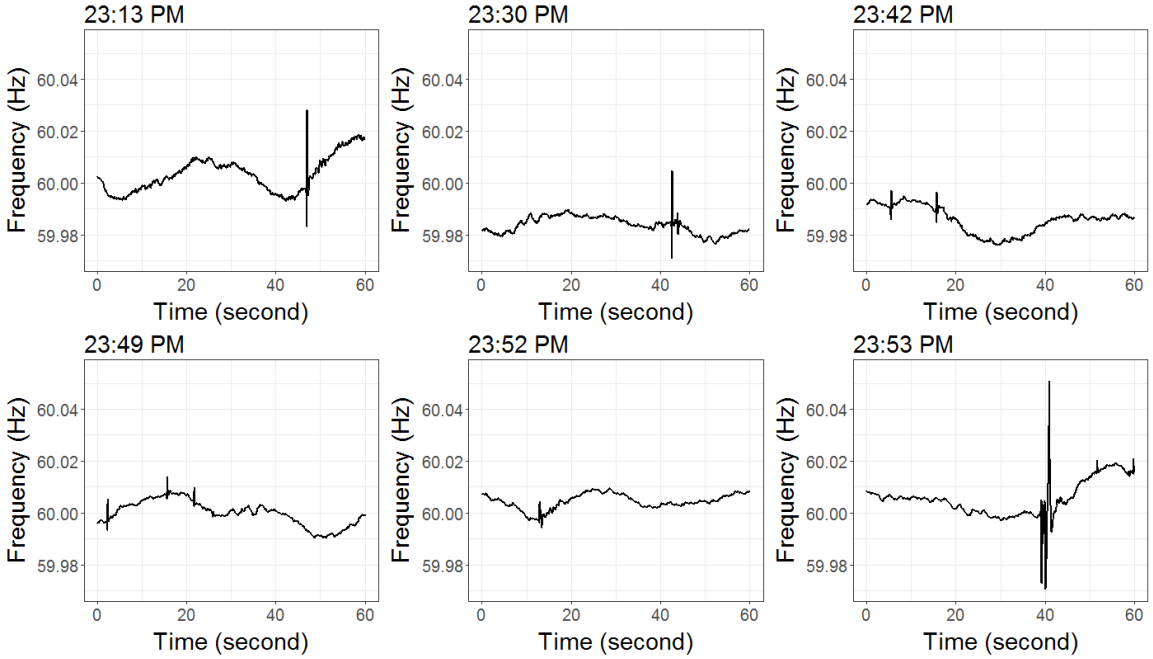


Fig. 4-8. An example of several small faults before a large fault.

events might be good indicators of larger events and detecting them might help in avoiding big events. Figure 4-8 is a good example of small faults happening before a big fault. All the small faults are missed by the given list. The last big fault is recorded by the list. Our algorithm detected all of the faults shown in this figure.

5.0 References

Abdi H. and Williams L., "Principal component analysis", *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433-459, 2010.

Amidan B. and Ferryman T., "Atypical Event and Typical Pattern Detection within Complex Systems", *IEEE Aerospace Conference Proceedings,* 2005.

Eastern Interconnection Data Sharing Network, Inc. *EIDSN*. 2017. <www.eidsn.org>.

Follum, J. and F. Tuffner. "A multi-channel method for detecting periodic forced oscillations in power systems." *2016 IEEE Power and Energy Society General Meeting (PESGM)*. 2016. 1-5.

Follum, J., F. Tuffner and U. Agrawal. "Applications of a New Nonparametric Estimator of Ambient Power System Spectra for Measurements Containing Forced Oscillations." *2017 IEEE Power Energy Society General Meeting*. 2017. 1-5.

Eastern Interconnection Data Sharing Network, Inc. *EIDSN*. 2017. <www.eidsn.org>.

Follum, J. and F. Tuffner. "A multi-channel method for detecting periodic forced oscillations in power systems." *2016 IEEE Power and Energy Society General Meeting (PESGM)*. 2016. 1-5.

Follum, J., F. Tuffner and U. Agrawal. "Applications of a New Nonparametric Estimator of Ambient Power System Spectra for Measurements Containing Forced Oscillations." *2017 IEEE Power Energy Society General Meeting*. 2017. 1-5.

Hastie T., Friedman J. and Tibshirani R., The Elements of Statistical Learning. New York: Springer, 2013.

Kosterev, D. N., C. W. Taylor and W. A. Mittelstadt. "Model validation for the August 10, 1996 WSCC system outage." *IEEE Transactions on Power Systems* 14 (1999): 967-979.

Kosterev, D., et al. "Implementation and Operating Experiences with Oscillation Detection Application at Bonneville Power Administration." *Proceedings of CIGRE 2016 Grid of the Future Conference*. 2016.

Lloyd, S. "Least squares quantization in PCM." *IEEE Transactions on Information Theory* 28 (1982): 129-137.

MacQueen, J. "Some methods for classification and analysis of multivariate observations." 1967.

MathWorks. "kmeans." 2017. <http://www.mathworks.com/help/stats/kmeans.html>.

Peak Reliability. *Universal Data Sharing Agreement*. 2017. <https://www.peakrc.com/aboutus/DataSharing/Pages/default.aspx>.

Rousseeuw, P. J. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis." *Journal of Computational and Applied Mathematics* 20 (1987): 53-65. <http://www.sciencedirect.com/science/article/pii/0377042787901257>.

Strobl C. and Zeileis A., "Why and how to use random forest variable importance measures (and how you shouldn't)", *useR! 2008, Dortmund, Germany*, 2008.

Trudnowski, D., D. Kosterev and J. Undrill. "PDCI damping control analysis for the western North American power system." *Power and Energy Society General Meeting (PES), 2013 IEEE*. 2013. 1-5.

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by **Battelle** *Since 1965*

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

www.pnnl.gov