

# Fast Extraction and Characterization of Fundamental Frequency Events from a Large PMU Dataset using Big Data Analytics

Rashid Baembitov, Tatjana Dokic, Mladen Kezunovic  
Texas A&M University  
[bae\\_rashid@tamu.edu](mailto:bae_rashid@tamu.edu), [tatjana.djokic@tamu.edu](mailto:tatjana.djokic@tamu.edu),  
[kezunov@ece.tamu.edu](mailto:kezunov@ece.tamu.edu)

Yi Hu  
Quanta Technology  
[yhu@quanta-  
technology.com](mailto:yhu@quanta-technology.com)

Zoran Obradovic  
Temple University  
[zoran.obradovic@temple.edu](mailto:zoran.obradovic@temple.edu)

## Abstract

*A novel method for fast extraction of fundamental frequency events (FFE) based on measurements of frequency and rate of change of frequency by Phasor Measurement Units (PMU) is introduced. The method is designed to work with exceptionally large historical PMU datasets. Statistical analysis was used to extract the features and train Random Forest and Catboost classifiers. The method is capable of fast extraction of FFE from a historical dataset containing measurements from hundreds of PMUs captured over multiple years. The reported accuracy of the best algorithm for classification expressed as Area Under the receiver operating Characteristic curve reaches 0.98, which was obtained in out-of-sample evaluations on 109 system-wide events over 2 years observed at 43 PMUs. Then Minimum Volume Enclosing Ellipsoid Algorithm was used to further analyze the events. 93.72% events were correctly characterized, where average duration of the event as seen by the PMU was 9.93 sec.*

## 1. Introduction

Instantaneous fundamental frequency and the Rate Of Change Of Frequency (ROCOF) are the main indicators of overall balance between the supply and demand and the changes in such balance [1]. There are multiple types of frequency related events, including presence of harmonics and sub-harmonics, low frequency oscillations, and fundamental frequency deviations [2]. This paper focuses on detection and duration of the fundamental frequency deviation events using PMU measurements.

Utilities have different practices for the fundamental frequency monitoring. Every utility sets the thresholds that when exceeded indicate a frequency event. One example can be found in [3]

where the statistical analysis is performed to evaluate the occurrences of frequency events in Great Britain over different periods of the year with predetermined fundamental frequency threshold set by the utility. Various types of frequency events were analyzed, and the disturbance classification was implemented based on wavelet multiresolution analysis and pattern recognition techniques in [4]. Several studies used Principal Component Analysis to reduce the dimensionality of the PMU dataset [5, 6]. The reduction in dimensionality helps handling large PMU datasets by removing the need to execute the detection on PMUs that are not affected by the event, which is useful for faults that may only be visible by a small portion of PMUs. Fundamental frequency events are system wide events and, in most cases, the changes in the signal can be detected by every PMU. Exceptions are some smaller intensity events where the change in the signal cannot be differentiated by some PMUs due to the algorithm inability to differentiate very small signal changes. . Hence, in the analysis of system-wide events it is beneficial to consider the complete set of measurements from all PMUs. Minimum Volume Enclosing Ellipsoid method was proposed to classify the types of events based on PMU measurements [7-9]. This method demonstrates promising capabilities in event detection and classification, but it was only tested on data from a few PMUs and for a limited number of events.

Our contribution in this paper is in enabling a fast extraction of frequency events from an extremely large dataset. To enable fast execution and accurate characterization of frequency events, we implement a two-step method. First, we execute a fast frequency event detection extracting 20 min windows of signals that contain frequency events. In this step statistical analysis is performed to collect a set of features that are then used to train two different classifiers: Random Forest, and CatBoost classifier. In the second step we implement a slower but more precise event duration characterization method based on Minimum Volume

Enclosing Ellipsoid that can specify the exact start and end time of the event recorded by the PMU data.

## 2. Background

In the AC electrical power systems, the voltages and currents can be represented as periodic functions:

$$x(t) = X_m(t) \cos(2\pi ft + \varphi) + D(t) \quad (1)$$

where  $t$  is time,  $X_m(t)$  is peak magnitude,  $f$  is fundamental frequency,  $\varphi$  is phase angle, and  $D(t)$  are disturbance signals, such as noise, harmonics, DC offset, inter-harmonic interference, etc.

The frequency is normally maintained within a small deviation from a nominal frequency (50 or 60 Hz). The instantaneous frequency of the power system is constantly changing with slight deviations from the nominal frequency.

The ROCOF is defined as the first derivative of instantaneous frequency, or a second derivative of the phase [10].

### 2.1. Fundamental Frequency Events

Small deviation of fundamental frequency occurs regularly in AC electric power systems, due to the mismatch between electricity supply and demand. To maintain normal operation, thresholds to detect small frequency deviation are set to alarm unacceptable operating conditions. The major deviation of system frequency from its nominal value can occur due to different contingencies in the system such as faults, sudden load increase or decrease, sudden loss of a generator, etc. Not all the faults will lead to frequency events (FE). The FE may occur when clearing the fault leads to system unbalance between supply and demand, such as during generator tripping, system islanding, loss of major load, etc.

The fundamental FE refers to situations when the frequency of the system exceeds the preset operational limits for normal operation [1]. To separate the normal fluctuations of frequency from contingency situations, electric power system operators set the limits that should not be crossed in normal operation. For example, in [11] unacceptable frequency conditions are defined as any situation in which steady state frequency falls outside of the statutory limits of +/-0.5

Hz relative to nominal frequency. In [11], the deviations outside of +/-0.5 Hz limits are rare, and only occur in case of severe events. Another example is ERCOT that has a larger deviation of fundamental frequency in the normal operation range. According to [12] any fundamental frequency between 59.97 Hz and 60.03 Hz is considered normal operation. We can see that in this case the smaller deviation (up to 0.03 Hz) is considered normal, as opposed to a previous example where it is set to 0.5 Hz.

The fundamental FE are system-wide events, that should be seen from all the PMUs connected to the affected power network. Thus, it is beneficial to observe these events from the recording available from multiple PMUs in the network because it makes the method robust in case of missing and bad data.

### 2.2. PMU Data

PMU is a measuring device that calculates estimated phasors of the sinusoidal voltage and current signal as described in Eq. (1). The voltage and current phasors are sent to the Phasor Data Concentrator (PDC) at the sampling rate of 30-120 samples per second (sps) depending on the device. Some PMUs record only positive sequence voltage and current, while some report the phasors for all three phases depending on the device setup.

PMUs also calculate power system fundamental frequency and ROCOF in addition to synchrophasors. The fundamental frequency and ROCOF are calculated and reported at the same sampling rate as synchrophasors. The fundamental frequency and the ROCOF measurements produced by PMUs should meet the accuracy and dynamic performance requirements specified by IEEE/IEC 60255-118-1 standard (the latest synchrophasor standard which superseded the IEEE C37.118.1-2011) [10].

### 2.3 Big Data and Apache Spark

In this paper we describe a method for frequency event detection and characterization that can be applied on an exceptionally large set of PMU measurements of dozens of Terabytes (TB). When dealing with such a large dataset it is necessary to use the tools that can store, retrieve, and process large sets of data in a distributed database. For that purpose, the data was stored as an Apache Parquet database, and Apache Spark is used to access and analyze the data.

The benefits of using compressed Apache Parquet files are presented in Table I [13]. We can see from the Table I that the parquet format provides a large compression rate and ensures a much smaller query run time, compared to the conventional tables stored as CSV files. Apache Spark provides multiple functionalities necessary for processing of the large

**Table I. Comparison of Parquet and CSV files [13]**

| Dataset                       | Size on Amazon S3 | Query Run Time |
|-------------------------------|-------------------|----------------|
| Data stored as CSV files      | 1 TB              | 236 seconds    |
| Data stored in Apache Parquet | 130 GB            | 6.78 seconds   |
| Savings                       | 87% less          | 34x faster     |

distributed datasets, including Spark SQL, Spark Streaming, Spark MLlib for machine learning, and GraphX for graph analysis.

### 3. Methodology

We will first describe the dataset used for the study, as well as the steps taken to preprocess it. Then, we will describe the steps taken to implement a Fast Frequency Event Detection method based on statistical analysis of 20-min windows of FFE using different machine learning algorithms. The 20 min window was selected for the reasons of computational efficiency. After that we will describe the frequency event characterization method based on Minimum Volume Enclosing Ellipsoid used to further analyze properties of fundamental FFE.

The reasons for using two methods to fully characterize the fundamental FFE are:

- *Fast Frequency Event Detection* is created to enable a fast execution on a large dataset. This method is very time efficient, but only provides limited differentiation of the types of frequency event. The method indicates a presence of fundamental frequency event inside a 20-min window, but it does not provide any additional event information.
- *Frequency Event Characterization* is a slower method that provides more precise characterization of FFE as seen by the PMUs. Because it is not time efficient, this method is only executed on a subset of data extracted by the Fast Frequency Event Detection. Executing this method on the entire dataset is not feasible, since the execution time would take multiple months even if using high-performance computing resources. Executing this method on a smaller PMU dataset gives more precise event characterization in reasonable time.

#### 3.1. Data Description and Preprocessing

Datasets used for the study reported in this paper include:

- Synchrophasor measurements collected from 43 PMUs for the period of 2 years. Total size of the dataset is 5.35 TB. Data is stored as an Apache Parquet database
- Historical event logs for the same period of two years stored as a CSV file. The event logs have imprecise timestamps for the event start/stop times

**3.1.1. Data Ingestion.** Data ingestion is the process of extracting data to the computational platform from original data source. It can be thought of as

“importing” of the data into the cluster. The data in the format of Apache Parquet files was copied to the High-Performance Cluster, and then was ingested by Apache Spark for further processing. The structure of the data was embedded into folders based on the time the data was collected. In such a way, one folder would include data for one day of measurements for all PMUs.

**3.1.2. Data Cleansing.** Data cleansing identifies “bad” and corrupted data, which would have little or none value for the further analyses and thus is being removed from dataset. The recordings from PMUs that contained less than 50% of the useful signals were removed from the dataset.

**3.1.3. Data Curation.** Data curation is associated with managing, transforming, and organizing the data in a way that is useful for the discovery of points of interest, i.e. finding events in the given dataset. The timestamps for the recordings from PMUs were all changed to the same timestamp format and were transferred to the same time zone. Before proceeding to the algorithmic part of the analyses one needs to determine if data needs to be sorted by time. For the first proposed step data does not need sorting, which allows for faster processing. For the second step of the method, the data needs to be sorted. It is also worth mentioning that one needs to sort data by the occurrence time for the purpose of plotting.

**3.1.4. Data Quality.** The PMU dataset has multiple data quality issues, including violations of data accuracy, data availability, and data timeliness [14]. More about data quality issues of PMU data can be found in [15]. In terms of data accuracy, bad data outliers are present in the data. It is important to separate these from the actual events in the network. When it comes to data availability, the dataset has multiple types of missing values, including missing timesteps, and missing measurements for the existing timesteps. The issues with the reported sample time were also detected, which indicates an existing violations of data timeliness. The frequency event characterization method developed in this paper can detect several types of data quality issues including missing measurements and outliers. The method is trained to skip the sections of data that are of unacceptable quality. If majority of the data points in a time window was missing or outliers were detected, the data window gets discarded. The algorithm raises a flag and proceed with further processing of the dataset. In our example, if at least 4 points of valid data are present in the data window, the algorithm keeps calculating the MVEE.

### 3.2. Fast Frequency Event Detection

The purpose of Fast Frequency Event Detection Module is to quickly locate FFE inside the large PMU dataset by sliding a 20-min window and collecting a set of statistical characteristics. A training set was selected, and different classifiers were used to separate between the FFE and normal operation. The historical event logs are used to extract two sets of time windows:

- 297 20-minutes long time windows (+/-10 min from the reported start of the event) around the frequency event instances reported in the event log as time stamps.
- 244 randomly selected 20-minutes long time windows during normal operation, taken from the periods when no event was reported in the event logs.

Then, these time windows were used to extract the data from the PMU database. Two measurements were selected from the PMU database: frequency and ROCOF.

**3.2.1. Threshold selection.** The detection method uses a set of predefined thresholds to collect the statistical characteristics of selected time windows of PMU data. The selection of different thresholds as statistical characteristics is demonstrated in Fig. 1 for frequency measurements, and Fig. 2. for ROCOF measurements.

**3.2.2. Feature Extraction.** Next step is the extraction of features based on statistical analysis of collected threshold tests on the selected dataset. For each 20 min time window, the following steps are taken for each of the 43 PMUs:

- Check the threshold violation for each timestep.
- Count the number of points beyond each threshold (total of 14 features, one for every frequency and ROCOF threshold).
- Extract minimum value of frequency and ROCOF (2 features minF, minDF)
- Extract maximum values of frequency and ROCOF (2 features maxF, maxDF)
- Generate a table with all the extracted features. Dimension of the table is 43 x 18, because there is 43 PMUs and total of 18 extracted features.

It is worth mentioning some physical aspect of the measurement points that lay beyond the threshold selection. They are proportional to the area of the analyzed signal above a specific threshold. For example, one can think about the area between the threshold and a frequency signal as an excessive energy, generated by the sources and not consumed by

loads nor attributed to the losses in the system. Instead, this excessive energy was spent on generators' acceleration that led to frequency increase. Similarly, if the measured signal was below the lower thresholds, the collected measurement points are proportional to the area of the signal under the specified threshold. In this case the generators are not producing enough energy to cover the increase in load that is causing the frequency to fall under the acceptable value.

### 3.2.3. Construction of the inputs for classification.

After the extraction all the features, they were combined into the input table used for the classification algorithm. The diagram of the process is shown in Fig. 3.

First, each table generated for a single time window is flattened into a vector containing 756 elements:

$$43 \text{ PMUs} \times 18 \text{ features} = \text{event vector [756]}$$

The features for a single PMU for a given 20-minutes window are as follows:

- # of frequency measurements above 60.5 Hz
- # of frequency measurements above 60.2 Hz
- # of frequency measurements above 60.1 Hz
- # of frequency measurements above 60.05 Hz
- # of frequency measurements below 59.95 Hz
- # of frequency measurements below 59.9 Hz
- # of frequency measurements below 59.8 Hz
- # of frequency measurements below 59.5 Hz
- # of ROCOF measurements above 1.5 Hz/sec
- # of ROCOF measurements above 1.0 Hz/sec
- # of ROCOF measurements above 0.5 Hz/sec
- # of ROCOF measurements below - 0.5 Hz/sec
- # of ROCOF measurements below - 1.0 Hz/sec
- # of ROCOF measurements below - 1.5 Hz/sec
- Minimum frequency measurement value
- Maximum frequency measurement value
- Minimum ROCOF measurement value
- Maximum ROCOF measurement value

Then the labels are created for each time window as:

- 1 – in case of a reported frequency event
- 0 – otherwise

The final training dataset contains following tables:

- X: [297+244] x [756]
- Y: [297+244]

The resulting table was analyzed by two different algorithms. The split of the data into training and testing set was performed randomly with Stratified K-Folds cross-validation with 5 folds.



Figure 1. Frequency thresholds

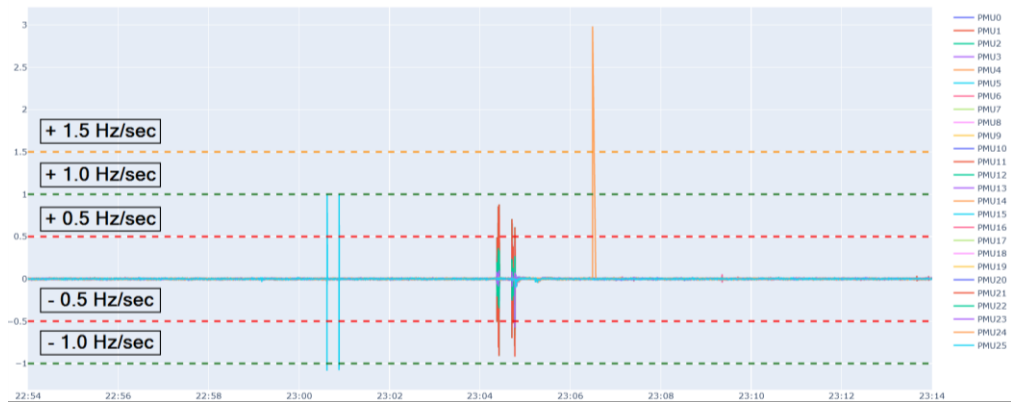


Figure 2. ROCOF thresholds

**3.2.4. Random Forest Classifier.** The first tested algorithm is Random Forest Classifier implemented in sklearn library (version 0.22.1) for Python where default parameters were utilized.

Random Forest is an ensemble algorithm [16]. It consists of multiple classification decision trees. So, each new event is classified separately by each of the decision trees where each tree puts the event in one of the two classes: NO or FFE. In other words, each tree “votes” on the event. The final verdict on the class is selected based on maximum votes [17]. The number of trees in the forest and how deep the trees are reflecting hyperparameters should be chosen for each specific problem.

**3.2.5. Catboost Algorithm.** The second tested algorithm is Catboost, which is an algorithm for gradient boosting based on decision trees [18, 19]. Gradient boosting algorithms belong to the family of ensemble algorithms and use weak classifiers in sequential manner to create strong classifier. A gradient boosting algorithm aims at achieving minimum error on the training dataset in a functional space where each function is a model. Every model in this composition assesses a gradient of the error for

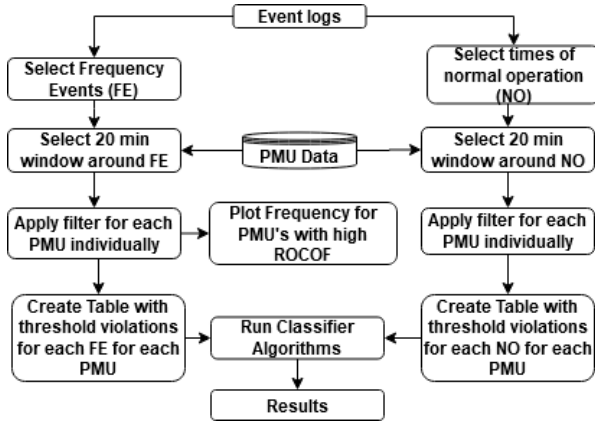
elements in a feature space. Predictions are added together using some weights to arrive to the final classification [20].

The following parameters were used for training:

- max iterations: 1000,
- learning rate: 0.01,
- early stopping rounds: 300,
- evaluation metric: AUC,
- tree depth: 8

### 3.3. Frequency Event Characterization

Fast frequency event detection method described in the previous section 3.2 is used to select 20 min windows of PMU data that contain a frequency event. Using this method, 446 20-min windows with FFE were selected. The next step is to further characterize these FFE by determining the exact time window in which the event is visible by the PMU and offset of that window from the reported event time (provided in the event log as time stamp). For that purpose, the event detection method was implemented based on the Minimum Volume Enclosing Ellipsoid (MVVE) method [7-9, 21-23]. MVVE incorporates all available PMUs, thus capturing the exact duration of the event across the system. The threshold method only detects



**Figure 3. Diagram of the Fast Frequency Event Detection Module**

measurement points when signal is over the threshold. The MVEE method can capture the entire length of the signal that deviates from the normal operation. That is useful for further characterization of such events, such as creating a distinct “portrait” of the event.

**3.3.1. Minimum Volume Enclosing Ellipsoid.** For a set of points in  $n$ -dimensional space, the MVEE is defined as the smallest possible  $n$ -dimensional ellipsoid that completely encloses all the points [7-9, 21-23]. as demonstrated in Fig. 4. The solution to the MVEE problem was implemented in Python based on the solution described in [21, 22].

The ellipsoid can be expressed in the center form as:

$$\varepsilon = \{x \in \mathbb{R}^n | (x - c)^T E (x - c) \leq 1\} \quad (2)$$

where  $c$  is the center of the ellipsoid, and  $E$  is a  $d \times d$  matrix for an ellipsoid in  $d$ -dimensional space. For example, if we are creating an ellipsoid in a 2-dimensional space, the matrix  $E$  would be  $2 \times 2$ . The point  $x_i$  is inside of the ellipsoid if:

$$(x_i - c)^T E (x_i - c) \leq 1 \quad (3)$$

Set of points that we are trying to enclose can be expressed as a matrix  $P$  of the size  $(d \times n)$ , where  $n$  is a number of points.

The MVEE can be found by solving the optimization problem:

$$\text{minimize } \log(\det(E)) \quad (4)$$

subject to:

$$(P_i - c)' * E * (P_i - c) \leq 1 \quad (5)$$

where  $P_i$  is the  $i$ -th column of the matrix  $P$  [11]. The solver is based on Khachiyan Algorithm [23].

After  $E$  and  $c$  are calculated, the volume of the MVEE can be calculated using Eq. (6):

$$\text{Vol} = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d+2}{2})} \det(E^{-1})^{\frac{1}{2}} \quad (6)$$

where  $\Gamma$  is the standard Gamma function of calculus.

### 3.3.2. Frequency Event Detection based on MVEE.

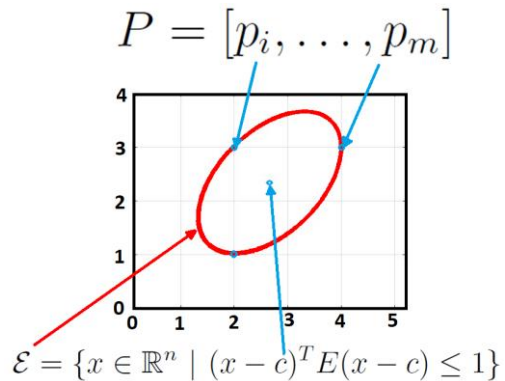
In this step the set of PMU measurements is enclosed in MVEE, and difference in MVEE Volume between normal operation and frequency event is used to detect the frequency event changes visible by the PMU. The change in PMU data samples have impact on the volume of MVEE [7-9]. During the normal operation, the volume is small, while every deviation from normal operation present in any of the input parameters will increase the volume of the MVEE, making it easy to detect.

The overview of the method is presented in Fig. 5. Each PMU measurement data stream is analyzed separately. The dimension  $d$  of MVEE is 4, as it takes four measurements, positive sequence voltage magnitude  $V_{pm}$ , positive sequence current magnitude  $I_{pm}$ , frequency  $f$ , and ROCOF  $df/dt$ . Number of points in each MVEE window varies based on the sampling rate and selected MVEE window size. Number of points must be larger than the selected MVEE dimension, otherwise the MVEE matrix is singular.

Two levels of processing using MVEE were implemented:

- Level 1. 10-sec MVEE window
- Level 2. 1-sec MVEE window

The reason for having two levels of window sizes is that the 20-min event window contains a large percent of normal operation cases. It is very time consuming executing a small 1-sec window over the 20-min event, and not necessary. We first use a larger 10-sec MVEE window to get rid of large chunks of data with normal operation. Then when we select a smaller subset of data that covers the frequency event. We use a smaller 1-sec MVEE window to get a more precise information about the frequency event. This way we only use a small 1-sec MVEE window over the subset



**Figure 4. Minimum Volume Enclosing Ellipsoid**

of data that has the most information about the frequency event.

The goal of the first level is to find the 30-sec window that completely encloses the event, using a 10-sec sliding MVEE window with a step of 10-sec. The 10-sec sliding MVEE window is executed over a 20-min event window determined using Fast Frequency Event Detection method from Section 3.2. In this case one sliding MVEE window is a 4 x 300 matrix for PMUs with 30 sps, and 4 x 600 matrix for the ones with 60 sps.

During the Level 1, the 10-sec window with maximum calculated MVEE Volume, out of all 20-min of data, is selected as the center of the event. In some cases, we discovered multiple events during the 20 min window. These events were separated manually into 30 sec windows. The development of automated way to separate them is left for future work. If multiple events occurred during one 30 sec window they are marked as overlapping events. Two 10-sec windows around this center (one before and one after) are also taken to form a 30-sec window of the event. This 30-sec window of the event is then sent to the Level 2 of MVEE processing.

The purpose of the Level 2 is to find the precise start and end time of the frequency event as seen by the PMU, using a smaller 1-sec MVEE window (with 0.5-sec steps between two consecutive MVEE windows). The 1-sec MVEE window slides over the 30-sec event window determined during the Level 1. In this case one sliding MVEE window is a 4 x 30 matrix for 30 sps, and 4 x 60 matrix for 60 sps PMUs.

During level 2, it is necessary to determine the MVEE volume threshold between a normal operation and frequency event. The subset of 30-sec events was selected and visually inspected to determine the visible extent of the frequency event from the PMU measurements. This was used to create labels for each 1 second of data (label = 1 for frequency event, label = 0 for normal operation). The MVEE volume was calculated for this subset of data. Then the histogram was used to determine the MVEE volume threshold between the frequency event and normal operation for a 1-sec MVEE window for each individual PMU.

After we have obtained all the thresholds, we can use them to select the 1-sec MVEE windows that are over the threshold. The FFE as visible by the PMU typically span over a few seconds. All the 1-sec MVEE windows that are over a threshold for a certain event on a certain PMU are grouped together to form a final event window.

The event log contains the information about the start of the FFE. We use this information to calculate the offset between the start of the frequency event that

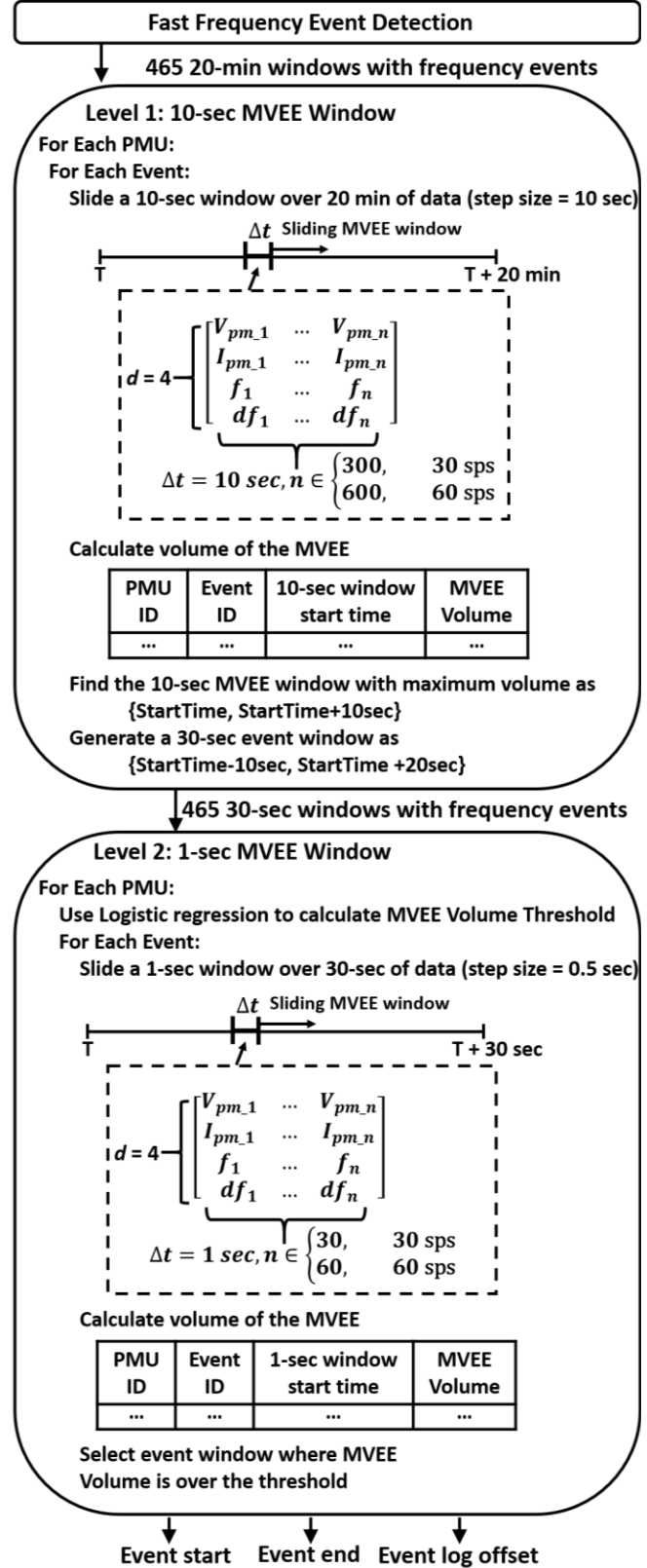


Figure 5. Two step processing in the Frequency Event Characterization Module

was provided in the event log, and our calculated start of the event based on PMU measurements.

The output of the FFE characterization module is: a) start of the frequency event as seen by the PMU, b) end of the frequency event as seen by the PMU, and c) offset between the start time from the event log and our calculated Event Start.

## 4. Evaluation

The model was implemented on the High-Performance Computing cluster at Texas A&M University with 28 cores, and 64 G of memory. The extraction of features took the longest time compared to other tasks and was completed in about 3 hours.

### 4.1. Evaluation metrics

Two classifiers were used to characterize datasets, and then a better performing classifier was chosen. The Confusion Matrices (CM), Receiver Operating Curves (ROC) and Precision Recall Curves (PRC) [24] were applied for results evaluation for each algorithm. ROC shows performance of a classifier as its threshold are changed. PRC depicts what combination of precision and recall can be achieved with different thresholds. F-1 score is the harmonic mean of the precision and recall. Table II is composed of the three metrics for each algorithm: Area Under the ROC (ROC AUC), Area Under the PRC (PRC AUC) and F-1 score.

### 4.2. Fast Frequency Event Detection Performance

Random Forest performed well as can be seen from Table II. CM for one of the K-folds for this algorithm is depicted in Table III One may see that algorithm is placing majority of Normal Operation events and FFE in the right classes. ROC and PRC are shown in blue color on the Fig. 8 and Fig. 9 respectively.

Catboost produced best results among tested classifiers. Table II contains metric outcomes for this algorithm. CM for Catboost is presented on Table IV The improvement from the previous algorithm is seen in lower number of false negatives for the Catboost. Yellow lines on Fig. 6 and Fig. 7 represent AUC and PRC for the algorithm.

Catboost allows out-of-the box feature importance evaluation. The most important features are:

- Minimum value of frequency and ROCOF
- Maximum values of frequency and ROCOF

If one is to construct a dataset consisting only of above-mentioned 4 features for each PMU, resulting in total of  $43 \times 4 = 172$  features for each 20-minute window, then up to 93% of the final scores could be obtained. That leads to the conclusion that if the computation time for the big dataset is an essential point, then the fast results can be obtained with usage

of only 4 features per PMU. This is what makes a proposed algorithm so fast in analyzing big datasets.

### 4.3. Frequency Event Characterization Results

Based on the Level 1 of the MVEE using 10-sec sliding window, majority of FFE were precisely characterized. Table V presents the main categories of events encountered during the Level 1 processing stage. Regular event is any frequency event that does not overlap with another frequency event. Overlapping events are the ones that share 30 sec time window. As we can see from the Table V:

- 382 FFE exactly coincide with the event log, and are shorter than 30 sec.
- 29 FFE have a starting time that exactly coincides with the event log, and are longer than 30 sec. The end time was not available in the event logs for any of the 446 events analyzed.
- 22 FFE were wrongly detected at the times when they did not occur. The reason for this is the MVEE detected another event (not the frequency event) inside the same 20-min window. The detected event had a stronger signature (MVEE volume) than the frequency event inside the same window, which made the algorithm to pick it up. One possible solution for this problem is removing the other types of the events from the dataset before the analysis of the FFE. This is left for the future work.
- 13 FFE were overlapping with each other. We analyzed the performance of the algorithm on them separately. Out of the 13 overlapping events, 7 (53.85%) were successfully detected and coincided with the event log (4 shorter than

**Table II. Metrics**

|                      | ROC AUC | PRC AUC | F1 score |
|----------------------|---------|---------|----------|
| <b>Random Forest</b> | 0.973   | 0.981   | 0.932    |
| <b>Catboost</b>      | 0.979   | 0.985   | 0.942    |

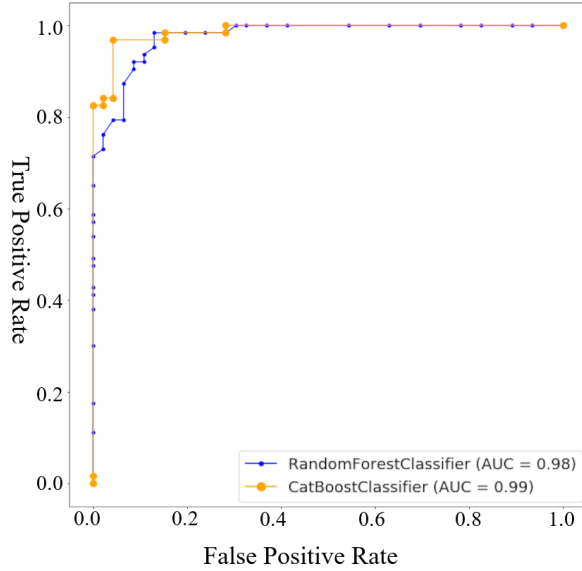
**Table III. Confusion matrix for Random Forest**

| Random Forest CM |     | Predicted label |    |
|------------------|-----|-----------------|----|
|                  |     | FFE             | NO |
| True label       | FFE | 41              | 5  |
|                  | NO  | 4               | 59 |

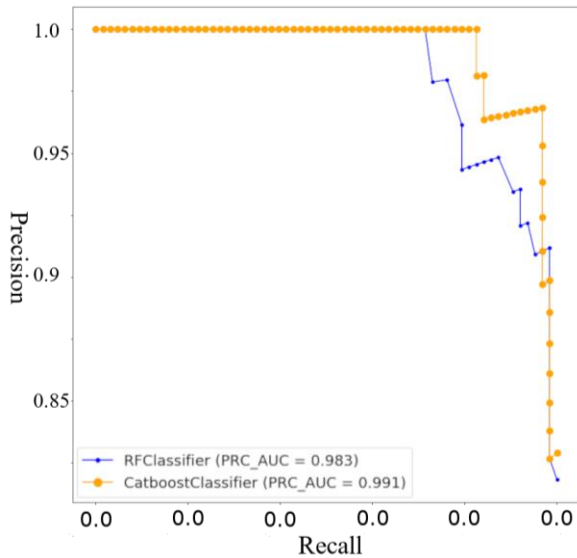
**Table IV. Confusion matrix for Catboost**

| Catboost CM |     | Predicted label |    |
|-------------|-----|-----------------|----|
|             |     | FFE             | NO |
| True label  | FFE | 44              | 2  |
|             | NO  | 4               | 59 |





**Figure 6. Receiver-Operating Curves**



**Figure 7. Precision-Recall Curves**

30 sec, and 3 longer than 30 sec), 6 events (46.15%) were not detected correctly due to their MVEE volume being smaller than for the other frequency event within the same data window.

For the events that are longer than 30 sec the start and end time was obtained directly from the level 1 results. All the events that were detected as being shorter than 30 sec are sent to the second level of MVEE analysis that uses a 1-sec MVEE window to analyze the selected 30-sec window of data.

Table VI presents the calculated minimum, maximum, and average duration of FFE that are sub 30-sec long obtained using Level 2 MVEE.

## 5. Conclusions

This paper presents a multi-level procedure for fast and accurate detection and characterization of FFE using PMU measurements. Following are the contributions of this paper:

- Fast Frequency Event Detection based on statistical analysis has been implemented. The method is capable of fast extraction of 20-min event windows containing FFE selected out of the large dataset of PMU measurements. The method can be used online to quickly detect FFE. Two machine learning techniques have been used to separate FFE from NO: Random Forest Classifier and CatBoost Classifier. CatBoost demonstrated the best performance with ROC AUC of 0.979.
- Frequency Event Duration based on Minimum Volume Enclosing Ellipsoid was implemented for precise characterization of fundamental FFE as seen from the PMU measurement data. The method determines the start and end of a frequency event as seen by the PMU. Using this method, a set of frequency event signals can be extracted to be used for further classification of the FFE recorded by PMUs.

### Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy,

**Table V. Level 1 of Event Characterization using 10-sec MVEE window – distribution of events based on detection outcome**

| Category                     | Duration | Number of Events (regular + overlapping) | % of Total Events |
|------------------------------|----------|--|-------------------|
| Event detected correctly     | < 30 sec | 382 + 4                                  | 93.72%            |
|                              | > 30 sec | 29 + 3                                   |                   |
| Event not detected correctly | -        | 22 + 6                                   | 6.28%             |

**Table VI. Level 2 of Event Characterization using 1-sec MVEE window – frequency event duration statistics for sub 30-sec events**

| Min     | Max    | Average  |
|---------|--------|----------|
| 0.5 sec | 22 sec | 9.93 sec |

completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

## References

- [1] P. S. Wright et al., "Field Measurement of Frequency and ROCOF in the Presence of Phase Steps," *IEEE Trans. on Instrumentation and Measurement*, vol. 68, no. 6, June 2019.
- [2] R.M. Gardner et al., "Power system event location analysis using wide-area measurements," in *2006 IEEE Power Engineering Society General Meeting*, Montreal, Que., Canada, 2006.
- [3] H. Wyman-Pain et al., "Changes in Frequency Events and the Frequency Response Markets in Great Britain," in *2018 IEEE Power & Energy Society General Meeting (PESGM)*, Portland, OR, 2018.
- [4] A. M. Gaouda et al., "Pattern Recognition Applications for Power System Disturbance Classification," *IEEE Trans. on Power Delivery*, vol. 17, no. 3, pp. 667-683, July 2002.
- [5] L. Xie et al., "Dimensionality Reduction of Synchrophasor Data for Early Event Detection: Linearized Analysis," *IEEE Trans. on Power Systems*, vol. 29, no. 6, pp. 2784-2794, Nov. 2014.
- [6] M. Rafferty et al., "Real-Time Multiple Event Detection and Classification Using Moving Window PCA," *IEEE Trans. on Smart Grid*, vol. 7, no. 5, Sep. 2016.
- [7] O. P. Dahal and S. M. Brahma, "Preliminary Work to Classify the Disturbance Events Recorded by Phasor Measurement Units," in *2012 IEEE Power and Energy Society General Meeting*, San Diego, CA, USA, July 2012.
- [8] O. P. Dahal et al., "Comprehensive Clustering of Disturbance Events Recorded by Phasor Measurement Units," *IEEE Trans. on Power Delivery*, vol. 29, no. 3, June 2014.
- [9] J. Ma et al., "Building Decision Trees for Characteristic Ellipsoid Method to Monitor Power System Transient Behaviors," in *IEEE Power and Energy Society General Meeting*, Providence, RI, USA, July 2010.
- [10] *Measuring relays and protection equipment - Part 118-1: Synchrophasor for power systems - Measurements, IEC/IEEE 60255-118-1:2018*, Dec. 2018.
- [11] National Grid, "National Electricity Transmission System Security and Quality of Supply Standard – Version 2.4," April 2019. [Online]. Available: <https://www.nationalgrid.com/document/141056/download>.
- [12] Electric Reliability Council of Texas, "ERCOT Fundamentals Training Manual," May, 2016. [Online]. Available: [http://www.ercot.com/content/wcm/lists/161158/ERCOT\\_Fundamentals\\_Manual.pdf](http://www.ercot.com/content/wcm/lists/161158/ERCOT_Fundamentals_Manual.pdf). [Accessed Oct 2020].
- [13] Databricks, "What is Parquet," [Online]. Available: <https://databricks.com/glossary/what-is-parquet>.
- [14] C. Huang et al., "Data quality issues for synchrophasor applications Part I: a review," *J. Mod. Power Syst. Clean Energy* 4, p. 342–352, 2016.
- [15] NASPI PMU Applications Requirements Task Force, "PMU Data Quality: A Framework for the Attributes of PMU Data Quality and a Methodology for Examining Data Quality Impacts to Synchrophasor Applications," NASPI-2017-TR-002, March 2017.
- [16] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, 2001.
- [17] T. Yiu, "Understanding Random Forest," August 2019. [Online]. Available: [towardsdatascience.com/understanding-random-forest-58381e0602d2](https://towardsdatascience.com/understanding-random-forest-58381e0602d2).
- [18] "Open-Source Gradient Boosting Library with Categorical Features Support," Yandex. [Online]. Available: <https://catboost.ai/>.
- [19] L. Prokhorenkova et al., "CatBoost: Unbiased Boosting with Categorical Features," in *Advances in Neural Information Processing Systems 31 (NIPS 2018)*, 2018.
- [20] The Apache Software Foundation, "Gradient Boosting," [Online]. Available: <https://apacheignite.readme.io/docs/gradient-boosting>.
- [21] N. Moshtagh, "Minimum Volume Enclosing Ellipsoids," [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.116.7691&rep=rep1&type=pdf>. [Accessed 3 June 2020].
- [22] N. Moshtagh, "Minimum Volume Enclosing Ellipsoid," MATLAB Central File Exchange, [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/9542-minimum-volume-enclosing-ellipsoid>. [Accessed 3 June 2020].
- [23] L. G. Khachiyan, "Rounding of polytopes in the real number model of computation," *Mathematics of Operations Research*, vol. 21, p. 307–320, 1996.
- [24] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation," *Journal of Machine Learning Technologies*, vol. 2, 2008.