2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe) | 978-1-6654-8032-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISGT-Europe54678.2022.9960589

Application of Machine Learning to Oscillation Detection using PMU Data based on Prony Analysis

Taif Mohamed, Mladen Kezunovic
Dept of ECEN
Texas A&M University
College Station, TX, USA
taif205@tamu.edu, kezunov@ tamu.edu

Zoran Obradovic
Computer and Information Sciences
Temple University
Philadelphia, PA, USA
zoran.obradovic@temple.edu

Yi Hu, Zheyuan Cheng
Quanta Technology
Raleigh, NC, USA
{yhu, zcheng}@quanta-technology.com

***Abstract*—Various types of oscillations could occur in the power grid from time to time. Most of them are harmless, while some could significantly impact the reliable power system operations. With increased penetration of renewable energy sources and the general transition to more complex power system operation comes the need for automated and accurate oscillation detection and classification methods. Such methods have been extensively studied in the past. Still, most of the earlier work was done for situational awareness purposes based primarily on simulated waveforms from synthetic power system models. This paper presents the results of an oscillation event detection method using machine learning algorithms trained on features extracted by Prony analysis from field-recorded PMU data. The unique experience of working with field-recorded historical synchrophasor data obtained from 38 PMUs located in the Western Interconnection of the US is shared. Four machine learning oscillation detection and classification models are trained using the results of Prony analysis as input features. The CatBoost classifier outperforms alternatives achieving 76.86% accuracy. An analysis of the data and related labels reveals several aspects of the event labeling that may have hindered the performance of the investigated detection and classification techniques. In the end, we suggest future event labeling approaches that might help avoid the challenges and limitations of current PMU recording practices.**

Keywords*—*oscillation, PMU, event detection, Prony, modal analysis

I. INTRODUCTION

Oscillations occur in the power grid for numerous reasons, including equipment failure, inadequate generator control tuning, transient response to faults, and power imbalances. Such oscillations can be categorized into two main categories: natural and forced, and within these categories, they are further split by the range of frequency. Natural oscillations occur in the power system locally and on the inter-area scale. As long as they are well-damped, these oscillations are manageable. When oscillations become poorly damped, they threaten the system stability and can cause damage to equipment, and might escalate to blackouts eventually. Even more dangerous are forced oscillations driven by abnormal conditions and may be close in frequency to natural system oscillation modes [1]. A major blackout in the US Western Interconnection occurred on August 10, 1996, due to forced oscillations interacting with a weakly damped system mode [2]. Consequently, detection and mitigation of oscillation events remain one of the power system operation and control challenges.

Oscillations can be defined by frequency mode, amplitude, and damping ratio. Oscillations of frequencies between 5 Hz and 50 Hz are generally categorized as subsynchronous, while oscillations under 5 Hz are categorized as low-frequency oscillations (LFO) [1]. Renewable resources can create low frequencies in the range of 10-15Hz. Mitigation and control devices exist for these different types, such as power system stabilizers for LFOs and subsynchronous damping controllers for subsynchronous oscillations [3]. Over the years, oscillation events have been extensively studied using ambient and ringdown analysis methods. Ambient methods, such as those proposed in [4]-[7], assume a quasi-steady state of the power system. Ringdown analysis methods, such as those implemented in [8]-[11], use the recorded data after an operation disturbance. The North American Electric Reliability Corporation (NERC) adopted ambient methods, including Yule-Walker, Least Squares methods, Frequency Domain Decomposition, and Stochastic Subspace Identification. Ringdown approaches mentioned by NERC include the Prony method, Eigensystem Realization Algorithm, Matrix Pencil, Variable Projection, Hankel Total Least Squares, and several frequency-domain methods [2].

The utility of these methods has been proven using either simulations or field recordings available for past events. These methods, especially ringdown methods such as Prony, perform best in the time window following a system event. With large amounts of data being collected today by phasor measurement units (PMUs), the detection of oscillations events and their locations and types is still a compelling area of research. Specifically, the introduction of machine learning to this field is most intriguing. Several variations of neural networks have been studied for such an application, including a novel neural network mentioned in [12] and a convolution neural network-based algorithm discussed in [13]. A significant challenge is the scarceness of oscillation events resulting in a lack of instances of such events to train these machine learning models. A transfer learning-based solution addressing this challenge was proposed in [14].

The PMU data used in our study are recorded in the Western Interconnection of the US. They have large amounts of missing data points, duplicated data, and outliers [15], which creates data quality challenges. Because it is also anonymized by eliminating PMU location and system topology, the option of using any power system model-based detection algorithm is infeasible. Furthermore, the event log provided with the data lacks some valuable details about the oscillation events, such as type or frequency.

Our contribution is the design of an oscillation data model that can detect and classify oscillation events by utilizing the basic PMU data features identified by Prony analysis. We augment the classic Prony method with a machine learning algorithm called CatBoost, making the detection more adaptive than tunning detection thresholds based on power system model simulations. In addition, based on our first-hand experience working with field-recorded historical PMU data, we offer various data labeling and feature extraction recommendations that could benefit researchers and engineers applying machine learning algorithms to improve wide-area monitoring and real-time operations based on PMU data.

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

The rest of the paper is organized as follows. Section II discusses the problem background, including the challenges presented by the data used in this study. Section III describes the methodology involved in developing and testing the classifier models. Section IV presents the results. Section V is a discussion of the observed algorithm performance and some possible explanations of the unexpected outcomes, followed by Conclusions and References.

II. Background

A. The US Western Interconnection

This study utilizes PMU data that has been recorded in the US Western Interconnection. The Western Interconnection has been extensively studied in the literature and by NERC and is known to have unique types of oscillations [15]. As previously mentioned, the interaction between forced and natural oscillations may lead to a resonance caused by the initial localized forced oscillation, where the amplitude of oscillations grows and can be seen across the system. The source of forced oscillations due to resonance is hazardous because it can cause generator protection in different parts of the system to trip in response [1]. Being aware of the frequencies of these modes by automatically detecting and classifying them is essential for the proper operator's response in case of a forced oscillation near these modes.

According to the Western Electricity Coordinating Council (WECC), the five inter-area modes are listed along with their corresponding nominal frequency and damping ratios in Table I [16]. While the two North-South modes are the most observed and well-studied, due to the availability of PMU field recordings, the lack of related field recordings for the other three modes has resulted in some unknown properties like their damping ratios during normal operation.

TABLE I. Western Interconnection Inter-Area Modes

Mode Name	Mode Properties	
	Nominal Frequency	***Damping Ratio***
North-South Mode A	0.25 Hz	10% to 15%
North-South Mode B	0.34 Hz to 0.4 Hz	5% to 10%
East-West Mode A	0.45 Hz	Unknown
British Columbia "BC" (North-West)	0.6 Hz	Unknown
Montana (North-West)	0.8 Hz	Unknown

B. PMU Data

Voltage, current, and frequency data are collected by 43 PMUs in the Western Interconnection of the US over two years (2016-2017). These PMUs report phasor measurements of positive sequence and three-phase voltage and current as well as frequency and rate of change of frequency (ROCOF) at the rate of 30 or 60 frames/second (fps). Accompanying the data is an event log containing the start and end time of events, event type, cause, and descriptor (label). Out of 4,854 events, only 100 events are labeled as oscillation events over two years. These events have a wide range of durations lasting from 2 minutes to 9 hours. Further descriptors such as cause, oscillation type (LFO or subsynchronous; local or inter-area), oscillation mode, amplitude, or damping ratio are not provided. We also have no information about the locations of the PMUs or the network topology, which added to the ambiguity of the problem. Therefore, the data must be visually inspected by a domain expert to determine or speculate the nature of these oscillations and decide on the best course of action for developing an automated event detection tool.

Statistical analysis of the data reveals several data quality issues that could hinder any event detection algorithm, as noted in the earlier work [15],[17].

- *Missing data*: PMUs are missing on average 0.69% to 30.01% of their positive sequence voltage magnitude and frequency measurements.
- *Data duplicates*: 19 PMUs report duplicate data points that reached up to 10^6 points on some days.
- *Outliers*: all PMUs are found to have outliers, but 6 PMUs had an excessive number of these outliers.
- *Flat 60 Hz*: four PMUs report flat 60 Hz frequency for extended periods of time.
- *Inconsistent voltage level*: four PMUs report inconsistent voltage levels.

Due to the aforementioned data quality issues, data from five PMUs are excluded from further studies. This decision is critical to avoiding the complications that might arise from inconsistent voltage and frequency measurements, which are pivotal to analyzing oscillation events. Accordingly, the final dataset comprises the measurements from the remaining 38 PMUs.

C. Prony Analysis for Feature Selection

The Prony method is chosen to build the set of features used by a machine learning algorithm in this study. Although we have not contributed to the computation method of Prony, which is widely studied, our approach of combining it with machine learning approaches is rarely found in the literature. The main reason behind our decision is to create a feature set that would hold information about the oscillation mode frequency, amplitude, and damping factor. The Prony method starts with an input, a data series that can be fitted into an exponential model as in (1). For an input vector $\mathbf{x}$ containing N samples, $\hat{\mathbf{x}}_k$ can be estimated as follows:

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{m} A_i e^{(\alpha_i + j2\pi f_i)(k-1)T + j\phi_i} = \sum_{i=1}^{m} h_i z_i^{k-1} \qquad (1)$$

where A_i, α_i, f_i, ϕ_i are amplitude, damping ratio, frequency, and initial phase of each mode i, respectively. T is the sampling time window in seconds. h_i and z_i are the residue and polynomial root of the i-th mode, respectively.

The solution involves finding the least-squares approximation of these components. The problem can be rewritten in matrix form as in (2).

$$\begin{bmatrix} x_m \\ x_{m+1} \\ \vdots \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} x_{m-1} & x_{m-1} & \cdots & x_0 \\ x_m & x_{m-1} & \cdots & x_1 \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-2} & x_{N-3} & \cdots & x_{N-m-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \qquad (2)$$

When (2) is rewritten as $\mathbf{x} = \mathbf{X}$ a, a least-squares estimate of a can be obtained. Next, the roots of the characteristic polynomial z_i can be found using the values of a computed in (2). The values of h_i can also be found using another least-squares approximation [8]. Finally, the four main components of the Prony approximation can be found from z and h, as shown in (3).

$$A_i = |h_i|, \alpha_i = \frac{\ln|z_i|}{T}, f_i = \frac{1}{2\pi T}\tan^{-1}\frac{\Im\{z_i\}}{\Re\{z_i\}}, \phi_i = \tan^{-1}\frac{\Im\{h_i\}}{\Re\{h_i\}} \quad (3)$$

III. Methodology

A. Feature Extraction

Twelve features are extracted for every PMU in the dataset after removing the PMUs deemed "bad PMUs" due to the data quality issues mentioned in Section II.B. Decisions such as reporting rate and order of Prony are chosen according to recommendations found in the literature in similar studies. The following steps are followed to extract these features:

1) Data retrieval: Data is retrieved from its initial storage form, the Apache Parquet format [18]. All data processing and event detection code is developed using Apache Spark [19] and Python [20].

2) Data sampling: Data is sampled using a sliding window of 10 seconds (T = 10s). Because the PMUs have two reporting rates, 30 and 60 fps, and computations are slowed down by data quantity. The data are downsampled so that every PMU would be reporting ten fps. This downsampling results in a total of 100 samples for each time window.

3) Data window: Every oscillation event is divided into time windows of 10 seconds. For normal operation time windows, 10-second time windows are obtained from every hour over two years, excluding the time ranges that correspond to the labeled events.

4) Computation of Prony parameters: With the order of Prony set to 20 modes, the four Prony parameters described in (6) are computed for each PMU at every time window. This setup results in 80 features per PMU per time window.

5) Feature selection: The modes computed by Prony are reordered by descending amplitude. Ignoring the first mode, which is the nominal frequency (60 Hz), the three modes of highest amplitude are selected. The four Prony parameters of these three modes are then saved as the feature set for the corresponding PMU. Twelve features are extracted for every PMU of the 38 PMUs. In total, the feature vector for every time window is composed of 456 features. An attempt is made to aggregate these features for all PMUs using a data fusion method. However, the resulting event detection accuracy is very low, suggesting that such an approach might be causing a loss of important information within the individual PMUs' Prony components.

B. Training the Classifier Models

Several machine learning models are used for the experiment. They are all designed using the scikit-learn library in Python [21], except for CatBoost, which has its own library [22]. Even though attempts have been made using other classifiers, our best-performing classifiers are Multi-Layer Perceptron (MLP), AdaBoost (AB), Random Forest (RF), and CatBoost (CB).

As mentioned in Section II.B, a hundred events labeled as oscillation events varied in terms of time length from 2 minutes to 9 hours. Therefore, the concept of a sliding window is used to get equal time windows of the events and to account for the nonlinearity that arises during more extended events. It has been reported that an oscillation event might evolve from one mode to another over time [23]. Such time windows are all labeled "1", indicating an event.

A critical step in model design is to split the training and test data temporally. The reason for the temporal split, as opposed to a random split, is that system modes overlap for the time windows of the same event and normal operation windows that are close in time. This similarity means that a classifier might have the features vectors overlap for time windows that correspond to an event in the training and test datasets, which could overestimate the performance. Consequently, the data was split into temporally disjoint subsets so that the data from the first year (2016) is used for training, and the data from the second year (2017) is used for testing. In another experiment, the roles are reversed, where 2017 is used to train the models, and 2016 is used to test them.

IV. Results

The results obtained by the four classifiers are summarized in Table II (training in 2016 and testing in 2017) and Table III (training in 2017 and testing in 2016). The classifiers' performance is evaluated using well-established metrics, including Area Under the Receiver Operating Curve (AUC), Area Under the Precision Recall Curve (AUPRC), Precision, Recall, and F-1 score [24]. In the tables, the classifiers are abbreviated as follows: multi-layer perceptron (MLP), AdaBoost (AB), Random Forest (RF), and CatBoost (CB).

The CatBoost classifier performs best by all the metrics. A significant improvement in performance from a 64.46% to a 76.86% accuracy is achieved when the data from 2017 is used to train the classifier models instead of the data from 2016. This result is interesting considering that the number of oscillations recorded in 2017 (36) is less than the number of oscillation events recorded in 2016 (64). So, the nature of the features pertaining to the events rather than the amount of training data in these separate years seems to matter.

TABLE II. Performance of different classifiers when trained in 2016 and tested in 2017

Classifier	Performance Metrics				
	AUC	**AUPRC**	**Precision**	**Recall**	**F1-score**
MLP	0.5561	0.4866	0.5451	0.5333	0.5391
AB	0.4932	0.4926	0.4862	0.4949	0.4905
RF	0.4753	0.4973	0.4139	0.4763	0.4429
CB	**0.6446**	**0.5891**	**0.5825**	**0.5589**	**0.5705**

TABLE III. Performance of different classifiers when trained in 2017 and tested in 2016

Classifier	Performance Metrics				
	AUC	**AUPRC**	**Precision**	**Recall**	**F1-score**
MLP	0.5632	0.8668	0.5260	0.5442	0.5349
AB	0.7343	0.9418	0.5903	0.6500	0.6187
RF	0.7040	0.9235	0.6247	0.7170	0.6677
CB	**0.7686**	**0.9229**	**0.5844**	**0.6532**	**0.6169**

V. Discussion

The results from Section IV inspire several questions:

1) *Why might using data from the year 2017 to train the models be better than using data from the year 2016?*

2) *Why is the maximum accuracy obtained by any of the classifiers in any experimental setup using field data*

limited to lower than 80%, which is generally not a satisfactory level of performance?

To address these questions, this study is first expanded to compare the performance of the same classifiers when each labeled oscillation event is isolated as a separate instance for testing. In addition, oscillation events are also studied in depth by other team members using different tools. As a result, more information about these events than what the labels conveyed is revealed, and other oscillations not included in the event log are discovered.

A. *Performance Comparison of Individual Oscillation Events*

All four classifiers perform better when the data used to train them is obtained from the year 2017 rather than 2016, and the testing is done in 2016. The most intuitive reason that comes to mind first is that the year 2017 might contain more data than the models can learn. However, this is not true since 2017 only has 36 oscillation events, compared to the 64 events found in 2016. To obtain more details about the behavior of these classifiers when encountering individual oscillation events, each event is isolated temporally and tested separately. One hundred small experiments are performed to achieve this analysis:

- For events 1 to 64: training data is made up of all 36 events in 2017 and a proportional number of normal operation time windows. Each event is then used as a separate test dataset, along with a balanced number of normal-operation time windows.
- For events 65 to 100: training data is made up of all 64 events in 2016 and a proportional number of normal-operation time windows. Each event is then used as a separate test dataset, along with a balanced number of normal-operation time windows.

The results of these experiments are illustrated in Fig. 1. The performance of CatBoost is chosen for this experiment since it is the classifier that performs best overall. It is evident in Fig. 1 that CatBoost has high accuracy (AUC) when tested on events in 2016 (1 to 64). The left-hand side of the graph exhibits values of AUC that are close to 1.0 except for five outliers. Wider dispersion of AUC values is seen on the right-hand side of the graph; however, the model's accuracy when testing on each event drops for much of the remaining 36 events. This behavior might suggest that the data contained in the events captured in 2017 is not as well-correlated or well-defined as the data captured in 2016.

B. *Study of Feature Randomness*

To test this theory, a randomness study is performed on the features extracted for events in 2016 versus those extracted from 2017. In this study, the "Runs test for randomness" is used to evaluate the degree of randomness in each feature over each year separately [25]. The runs test is a statistical approach for determining whether a dataset was created through a random process. The test statistic computed by the runs test is the z-value [26]. The test is a typical two-sided statistical test in which the z-value obtained for each feature can be compared against a critical value. The critical value selected for a 99% level of confidence is 2.576. Based on this test:

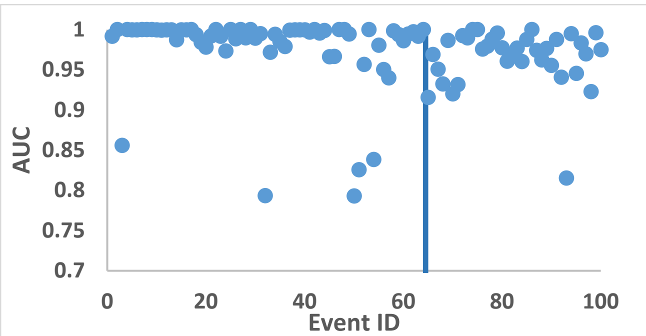

Fig. 1. Performance of CatBoost on individual events.

- The year 2016 has 99 random features out of 456 (21.7%).
- The year 2017 has 295 random features out of 456 (64.7%).

This considerable difference in the randomness content of the extracted features in these two different years might explain the disproportionate comparison in performance of the models in the two years. Because no pattern is found in the randomness of the features, none of the features are excluded from the experiment. Removing only those features identified by the runs test did not lead to improved accuracy, either.

C. *Other analysis methods*

Separate studies are explicitly performed on the same oscillation events using other analysis methods. The oscillation events included in the event log are visually inspected by observing the positive sequence voltage angle difference. Fast Fourier Transform (FFT) and Matrix Pencil are cross-validated as event detection methods. In this experiment, which is performed early on in the study, it is assumed that these oscillations are all of low frequency. Therefore, the captured frequency modes are limited to the range of 0 – 2 Hz. These three combined methods can identify 86 out of the 100 events in the event log. The damping ratios of the modes of these events range between 20% and 40%, which would be considered sufficient damping in the normal operation scenario. This finding drives the decision regarding feature selection described in Section III.A.4). The modes are kept as features based on their magnitudes rather than their frequencies. This decision allows the algorithm to identify both subsynchronous and low-frequency oscillation events.

Even more interesting is that these methods can identify oscillation events that are not included in the event log. Using real-power data, relative angle difference data, and frequency data, 79, 224, and 755 oscillations are discovered even though they are not labeled. These events are only within the days that already have a labeled oscillation event, suggesting that a much larger number of events could be found over the entire dataset. The main hindrance in this kind of study is the lack of knowledge of the criteria used to label oscillation events in the original event log. The 100 events identified by the event label are not accompanied by any further information about the cause or description of the events. We do not know whether only low-frequency events are labeled "oscillation" or if subsynchronous oscillations also qualify for this label. These ambiguities in the provided data labels can further explain the limited performance of our classifiers.

VI. Conclusion

The following may be inferred from our study:

- Using Prony method parameters extracted from the PMU data as features to train and test machine learning classifiers might lead to high randomness in the feature content, as seen in the data extracted from the second year of the data we used (2017).

- It is crucial to include further descriptors for oscillation events in event labels like type (low frequency or subsynchronous), dominant frequency mode, and damping ratio. It will also be helpful to facilitate the event mitigation if events are labeled as local or inter-area to help find the source of oscillations.
- Criteria for what qualifies as an "oscillation" event should be more precise. In the inspected low-frequency oscillations (0 – 2 Hz), the damping ratios of the events in the log ranged mainly between 20% to 40%, even though the literature suggests that oscillations may be of concern when their damping ratios drop below 10%.

DISCLAIMER

REFERENCES

[1] "Forced Oscillation Monitoring & Mitigation September 2017," North American Electric Reliability Corporation (NERC), 2017. [Online]. Available: https://www.nerc.com/comm/RSTC_Reliability_Guidelines/Reliabilit R_Guideline_-_Forced_Oscillations_-_2017-07-31_-_FINAL.pdf.

[2] "Interconnection oscillation analysis," North American Electric Reliability Corporation (NERC), 2019. [Online]. Available: https://www.nerc.com/comm/PC/SMSResourcesDocuments/InterconI nterco_Oscillation_Analysis.pdf.

[3] H. Hosseini, B. Tusi, N. Razmjooy, and M. Khalilpoor, "Optimum Design of PSS and SVC Controller for Damping Low Frequency Oscillation (LFO)," The 2nd International Conference on Control, Instrumentation and Automation, 2011.

[4] G. Liu and V. M. Venkatasubramanian, "Oscillation Monitoring from Ambient PMU Measurements by Frequency Domain Decomposition," in Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on, 2008, pp. 2821-2824.

[5] J. W. Pierre, D. J. Trudnowski, and M. K. Donnelly, "Initial Results in Electromechanical Mode Identification from Ambient Data," Power Systems, IEEE Transactions on, vol. 12, pp. 1245-1251, 1997.

[6] S. A. N. Sarmadi and V. Venkatasubramanian, "Electromechanical Mode Estimation Using Recursive Adaptive Stochastic Subspace Identification," IEEE Transactions on Power Systems, vol. 29, pp. 349-358, 2014.

[7] M. Farrokhifard, M. Hatami, and M. Parniani, "Different Approaches for Estimation of Dampings and Frequencies of Electromechanical Modes from PMU Ambient Data," in 2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC), 2015, pp. 1748-1753.

[8] J. F. Hauer, C. J. Demeure, and L. L. Scharf, "Initial Results in Prony Analysis of Power System Response Signals," Power Systems, IEEE Transactions on, vol. 5, pp. 80-89, 1990.

[9] Z. Tashman, H. Khalilinia, and V. Venkatasubramanian, "Multi-Dimensional Fourier Ringdown Analysis for Power Systems Using Synchrophasors," IEEE Transactions on Power Systems, vol. 29, pp. 731-741, 2014.

[10] A. Zamora, V. M. Venkatasubramanian, J. A. de la O Serna, J. M. Ramirez, and M. R. A. Paternina, "Multi-dimensional Ringdown Modal Analysis by Filtering," Electric Power Systems Research, vol. 143, pp. 748-759, 2017/02/01/ 2017.

[11] M. Hatami, M. Farrokhifard, and M. Parniani, "A Non-Stationary Analysis of Low-Frequency Electromechanical Oscillations Based on a Refined Margenau-Hill Distribution," Power Systems, IEEE Transactions on, vol. PP, pp. 1-12, 2015.

[12] Z. Shen and R. Ding, "A Novel Neural Network Approach for Power System Low Frequency Oscillation Mode Identification," 2019 IEEE International Symposium on Circuits and Systems (ISCAS), 2019.

[13] Q. Li, Z. Xu, X. Qin, K. Zhang, C. Zhang, and S. Ye, "A Convolution Neural Network Method for Power System Oscillation Type Identification," 2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2), 2020.

[14] Z. Yu, D. Shi, J. Li, Y. Wang, X. Zhao, Z. Wang, and J. Li, "Using Transfer Learning to Distinguish Between Natural and Forced Oscillations," 2020 IEEE Power & Energy Society General Meeting (PESGM), 2020.

[15] Z. Cheng, Y. Hu, Z. Obradovic, M. Kezunovic "Using Synchrophasor Status Word as Data Quality Indicator: What to Expect in the Field?," Proc. IEEE Int'l Conf. on Smart Grid Synchronized Measurements and Analytics – SGSMA, Split, Croatia, May 2022.

[16] "Document name modes of Inter-Area Power Oscillations in ...," Western Electricity Coordinating Council: WECC, 30-Nov-2013. [Online]. Available: https://www.wecc.org/Reliability/WECC%20JSIS%20Modes%20of %20Inter-Area%20Oscillations-2013-12-REV1.1.pdf.

[17] "PARTF: PMU Data Quality: A Framework for the Attributes of PMU Data Quality and a Methodology for Examining Data Quality Impacts to Synchrophasor Applications," North American SynchroPhasor Initiative, Mar-2017. [Online]. Available: https://www.naspi.org/reference-documents?f%5B0%5D=reference_documents_authored_on%3A2016-08.

[18] Databricks, "What is Parquet," [Online]. Available: https://databricks. com/glossary/what-is-parquet.

[19] Apache Spark, [Online] Available: https://spark.apache.org/

[20] Python, [Online] Available: https://www.python.org/

[21] Scikit-learn – Machine Learning in Python, [Online] Available: https://scikit-learn.org/stable/

[22] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," in Advances in Neural Information Processing Systems 31 (NIPS 2018), 2018.

[23] M. V. Venkatasubramanian, M. M. Farrokhifard, and Y. Zhi, "Oscillation Monitoring of the RTE Power System using Synchrophasors," Power Systems Engineering Research Center (PSERC) Report, Sep-2019. [Online]. Available: https://documents.pserc.wisc.edu/documents/publications/reports/2012_reports/S_81G_Final_Report__1_.pdf.

[24] J. V. Bradley, "Distribution-free Statistical Tests," Englewood Cliffs, New Jersey: Prentice-Hall; 1968.

[25] M. A. Bujang and F. E. Sapri, "An Application of the Runs Test to Test for Randomness of Observations Obtained from a Clinical Survey in an Ordered Population," Malaysian Journal of Medical Sciences, vol. 25, no. 4, pp. 146–151, Aug. 2018.

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