Detection, Characterization, and Mitigation of Disruptive Events (DECODE) by Combining Machine Learning/Artificial Intelligence on Synchrophasors and Physics-based Analysis

Final Project Report

S-100

Power Systems Engineering Research Center

Empowering Minds to Engineer
the Future Electric Energy System

Detection, Characterization, and Mitigation of Disruptive Events (DECODE) by Combining Machine Learning/Artificial Intelligence on Synchrophasors and Physics-based Analysis

Final Project Report

Project Team
Mladen Kezunovic, Project Leader
Texas A&M University

Vijay Vittal
Anamitra Pal
Arizona State University

Vaithianathan (Mani) Venkatasubramanian
Washington State University

Graduate Students
Taif Mohamed
Texas A&M University

Yadunandan Paudel
Arizona State University

Deepak Joshi
Washington State University

PSERC Publication 24-06

December 2024

For information about this project, contact:

Mladen Kezunovic
Texas A&M University
Department of Electrical and Computer Engineering
Wisenbaker Engineering Building 323C
College Station, TX 77843
Phone: 979-845-7509
Email: kezunov@ece.tamu.edu

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: http://www.pserc.org.

For additional information, contact:

Power Systems Engineering Research Center
Arizona State University
527 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-727-2052

Acknowledgments

The data used in this project was provided by Pacific Northwest National Laboratory (PNNL) and Réseau de Transport d'Électricité (RTE). Portions of this research were conducted with the advanced computing resources provided by Texas A&M High Performance Research Computing and the Sol supercomputer at Arizona State University.

Executive Summary

The advances in machine learning/artificial intelligence (ML/AI) over the last 10 years have enabled development and implementation of effective algorithms for the detection and classification of events using phasor measurement unit (PMU)/synchrophasor data. While many solutions have been proposed, there is a critical need for automated tools that are accurate, cost-effective, and computationally efficient when applied to large historical and streaming datasets captured by multiple PMUs. By leveraging over 20 years of research that used physics-based analysis to correlate synchrophasor data to disruptive events, this project has developed automated tools that combine various data-based and physics-based solutions in real-time to help system operators detect and classify three types of events, namely, faults, frequency, and oscillation events, that may lead to system emergencies.

The objectives of this project were attained by making progress on four fronts: 1) Development of data handling and sorting techniques for collecting large historical synchrophasor datasets to improve bad data detection, feature engineering, and event label assignment for automated ML/AI solutions; 2) Identification of the most effective ML/AI techniques for unsupervised, semi-supervised, and supervised learning associated with faults, frequency, and oscillation events, 3) Seamless integration of ML/AI-based and physics-based solutions for cost-effective and computationally efficient automated event characterization, and 4) Development and selection of appropriate transfer learning (TL) methodologies associated with detection and characterization of faults, frequency, and oscillation events that provided high accuracy when applied to datasets for which the ML/AI algorithm was not trained on.

The diversity of system topologies and changing PMU data distributions with each new dataset, which may even be from different grids, are challenging problems for traditional power system event detection and characterization techniques. To ensure validity of the detection and characterization methodologies developed in this project, they were applied to a wide range of datasets with diverse operating conditions. The proposed methodologies were found to significantly reduce the training effort while maintaining high accuracy of prediction. A unique challenge that was identified and subsequently addressed in this project was the analysis of new, small datasets that exhibited a scarcity or lack of event labels. These datasets were provided by industry partners but were often found to be insufficient for training robust ML/AI models. To overcome this challenge, the ML/AI models were developed for each event type and then TL techniques were utilized to transfer the models to the new datasets.

In the following paragraphs, a brief description of the research methodology starting with the datasets that were used in this analysis, is provided. This is followed by a summary of the main results as well as a discussion of their implications, including recommendations based on the project's findings.

Field-recorded synchrophasor measurements from various regions of the US and France were the primary sources of "real" data used in this study. However, the recorded measurements suffered from data-quality issues; these issues were tackled in the data preprocessing stage. The real-world measurements also lacked the occurrence of all three events that were the focus of this study. Training ML/AI models with an incomplete dataset could introduce bias and failure in accounting

for major, infrequent events that could significantly impact the system. Therefore, a wide range of “simulated” datasets were created for all three types of events to create a balanced training dataset for the ML/AI algorithms.

Feature engineering, which is an important step in the overall task of decoding power system events of interest using ML/AI models, was focused on next. It was observed that although synchrophasor data could be used directly or replaced by their simple statistical descriptors to be used as features, the extraction of relevant contrasting features for various events usually gave better results. As such, careful consideration was given in this study to select features that appropriately represent the underlying application (line fault events, frequency events, and oscillation events). To that end, features for fault, frequency, and oscillation events were extracted using principles of protection studies, modal decomposition, and image recognition, respectively.

The focus then shifted towards the development of the ML/AI models, particularly their training and testing. The developed ML/AI-based algorithms exploited the steps of data preprocessing and feature engineering to create useful training and testing datasets. These datasets were used as input to several supervised learning algorithms to form the base classifiers for each event type. TL methods were then explored to transfer the knowledge gained by models trained on simulated datasets or large, diverse field-recorded datasets to smaller, less diverse field-recorded datasets.

Next, the results obtained using the different ML/AI models were explored, and conclusions were drawn about suitability of the different models and approaches. The team found that correlation alignment (CORAL)-based TL with support vector machine (SVM) classifiers was particularly effective for line fault detection, while extreme gradient boosting (XGBoost) classifiers with fine-tuning based on Prony features demonstrated good performance for frequency event detection. Image-based ML/AI approaches showed promise for oscillation detection and classification as they were matched with neural network (NN)-based TL. The direct application of the base ML/AI classifiers trained using simulated datasets to field measurements during testing resulted in poor performance. However, exploring the TL capability of ML/AI models by utilizing simulated datasets along with limited field-recorded labeled events significantly improved the performance in classifying unlabeled field-recorded events.

Finally, the major study findings from this project along with reflections on some of the challenges that the team experienced with synchrophasor data quality in the datasets that were made available to them, was presented. The report concludes with some recommendations that stand out based on the team’s learning experience over the duration of this project and past experiences in a similar space and outlines the expected outcomes if these recommendations are considered.

Project Publications:

[1] T. Mohamed et al., "Automated Detection and Classification of Critical Power System Events using ML on PMU Data," 2024 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA), Washington, DC, USA, 2024, pp. 1-6, doi: 10.1109/SGSMA58694.2024.10571526.

[2] T. Mohamed et al., “Transfer Learning on Synchrophasor Data for Automated Detection and Classification of Critical Power System Events,” Submitted to IEEE Transactions on Power Systems.

Student Theses:

[1] Taif Mohamed, “Machine Learning Applications to the Detection and Classifications of Power System Faults,” PhD, August 2025.

[2] Yadunandan Paudel, "Decoding Frequency Events in Power Systems using Machine Learning on Synchrophasor Data," PhD, May 2026.

[3] Deepak Joshi, " Data-Driven Machine Learning Approaches for detecting and Classifying Oscillations in Power Systems," PhD, July 2025.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background

Synchronized phasor measurements enable efficient detection and classification of critical power system events. With the widespread deployment of phasor measurement units (PMUs), there is now an abundance of high-resolution data streams that significantly improve event analysis [1]. However, the vast volume of real-time PMU data streaming presents a challenge for operators and engineers, as manually interpreting critical grid events through visual inspection is time-consuming. Furthermore, variations in the topologies and data distributions of PMU data sources in different grids necessitate reliable methods for transferring knowledge from one dataset to another [2]. The critical events targeted in this study are *line fault events*, *frequency events*, and *oscillation events*. Each of the three event types has been explored in previous work and can be analyzed using empirical methods [3]-[7]. This project presented the unique opportunity to study, identify, and analyze these event types using machine learning/artificial intelligence (ML/AI) techniques based on field-recorded PMU measurements provided by industry partners at different geographical locations with varied power system characteristics.

ML/AI methods are being applied to various areas of power systems operation and control due to their ability to recognize patterns and perform big data analysis automatically and efficiently in the cases where humans may have difficulties dealing with such a complex analysis in a short timeframe. In the case of critical event detection, using ML/AI can replace the manual interpretation of such events typically done through visual inspection and empirical tools. However, the effective use of machine learning (ML/AI) tools to automatically analyze field-recorded PMU data encounters constraints due to data quality issues such as significant missing data, unreasonable measurement values, and inaccurate labels and timestamps. Additionally, event labels accompanying raw PMU measurements are often scarce, inaccurate, or imprecise, hindering the direct application of simple, open-source supervised ML/AI algorithms [8]. Previous work has addressed the issue of label quality and scarcity through the use of simulated data, as is also done in this project. However, merely simulating more data for training is insufficient for creating a robust ML/AI model applicable to any new dataset. ML/AI models have to be designed meticulously to fit the task of analyzing such events through data preprocessing, feature engineering, and ML/AI algorithm choice and tuning. Finally, the knowledge transfer gained by these ML/AI models from one dataset to another from different power systems must be done through the appropriate use of transfer learning techniques that best fit the type of data, features, and ML/AI algorithms used for the original dataset. The need to apply transfer learning to new datasets is due to the difference in data distributions as well as the scarcity or lack of labels for these datasets. The reason TL was chosen as a means to overcome this challenge is also due to the small size of these datasets, which prevents the direct application of methods such as unsupervised learning or reinforcement learning. Chapter 4 of this report provides the reasoning behind the decisions made in terms of the use ML/AI base models and TL models. Subsections 1.1.1, 1.1.2, and 1.1.3 present a review of the available literature pertaining to the detection and classification of line faults, frequency events, and oscillation events, respectively.

1.1.1 Line Fault Detection and Classification

Extensive research has focused on utilizing ML/AI techniques for line fault detection. For example, decision trees (DTs) have been employed to distinguish between various fault types, including single-line to ground (SLG), line-to-line (LL), three-phase (TP) faults, and normal system conditions as discussed in [9]. Additionally, the effectiveness of several ML/AI algorithms, including Bayesian networks, support vector machine (SVM), multi-layer perceptron (MLP), and DTs, in identifying and predicting SLG and TP faults using simulated PMU data was examined in [10]. Another study that compared the performance of several ML/AI algorithms included linear discriminant analysis (LDA), artificial neural networks (ANN), SVM, *k*-nearest neighbors (*k*NN), and DTs applied to PMU data, did so by simulating the IEEE 123-bus system [11]. The main goal of the study in [11] was to differentiate between load loss, generation dip, and line faults, for which an online event-classifier based on quadrant discriminant analysis (QDA) was developed. To avoid overfitting the models during training, the features in [12] were selected using sequential forward selection (SFS). One common aspect among most of the studies is the use of simulated data to train, validate, and test the developed ML/AI models. One field-recorded PMU dataset has appeared in several studies and was made available by the Bonneville Power Administration (BPA). This dataset made the testing of these ML/AI models more practical. The ML/AI algorithms developed using BPA data include DTs [9], hierarchical clustering and *k*-means clustering [13], and DBSCAN [14]. While there is ample work done in the use of ML/AI to detect and classify line faults, this project presented a unique situation in which new field-recorded datasets with varied data distributions and conditions pertaining to data quality and labels exist. This setting inspired the team to search for a different approach to this problem.

The performance of ML/AI models can degrade when applied to datasets with differing attributes and distributions from those that they were trained on. Transfer learning (TL), a subset of ML/AI, addresses these issues. Reference [15] demonstrated an extensive application of TL to classify power system events using simulated data to create a comprehensive training dataset. However, transferring knowledge from one dataset to another introduces challenges such as event label scarcity, as noted in [16]. In this project, the new datasets are smaller and less varied in line fault types compared to those used in [16], necessitating a different TL approach to leverage the new test dataset effectively while maintaining the success of previously developed classifiers. TL was chosen among other ML categories in order to take advantage of the successful line fault classification and detection models created by the team in previous work [6]-[7], which could not be directly applied to the new datasets due to the difference in data distributions. To enhance prior efforts and adapt to the limited datasets from various sources, this report integrates TL with SVM and random forest (RF) techniques. Several methods and choices will appear throughout the report regarding line fault detection and classification that are built on the team’s previous experience during previous project [6]-[7]. The previous project focused on the use of the field-recorded dataset, which will be introduced later in this report and referred to as PNNL1. This large dataset was provided by PNNL and included anonymized PMU measurements recorded over the span of two years across the US Western Interconnection.

1.1.2 Frequency Event Detection and Classification

Fundamental frequency deviations occur in power systems due to the mismatch between power generation and demand. When the total generation is lower than the total demand, the system frequency drops and vice versa. Figure 1 shows the frequency balance phenomenon at a high-level by analogy to a beam balance - with supply (generation) and demand (load) at opposite ends [17].

In the bulk power system, real-time power demand and generation power exhibit stochastic behavior. Subsequently, the frequency of the power system is never constant at the designated nominal value but varies closely around it. Guidelines for operational and statutory frequency limits are usually established by state, regional, or federal authorities to keep the power system operating in a secure and reliable manner. An instance of such a guideline is given in [18] by the Western Electricity Coordinating Council (WECC) in the US, which is a recommendation to its members who agree to adopt these operating procedures during contingent frequency behavior to keep the Western Interconnection efficient and reliable. Additionally, the modern trend of increase in the generation mix of renewable sources by displacing the use of conventional synchronous generators, cause the rotational inertia of the entire system to decrease resulting in a decline of the primary frequency response [19]. This can cause an increase in the probability of occurrences of frequency events which is evident by some recently reported events in Great Britain [20]. Figure 2 shows an example of a sudden load increase or a generation trip (both of which are frequency events) and depicts a power system's frequency response after the disturbance [21].

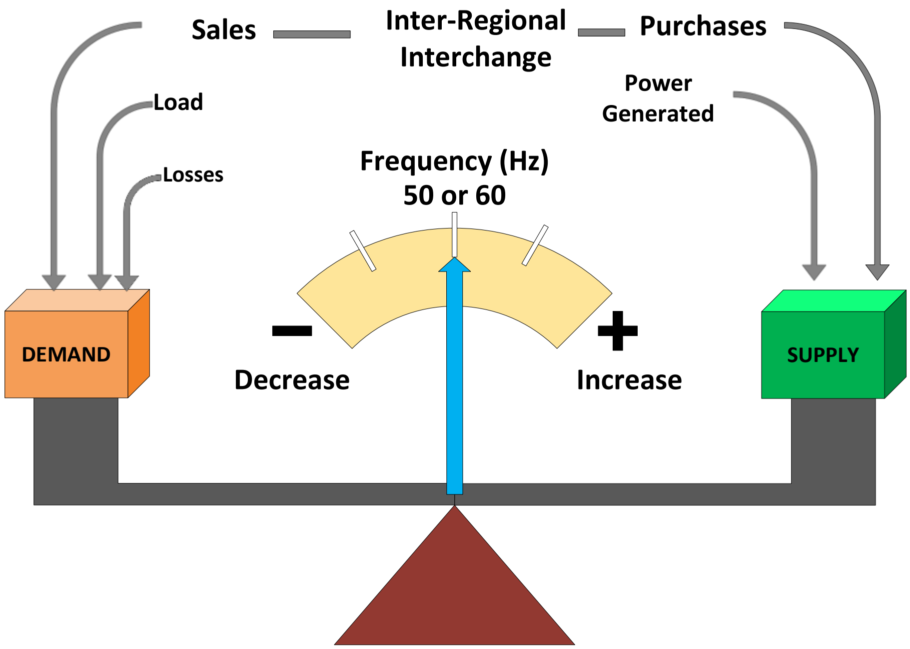


Figure 1: Power Balance and Nominal Frequency

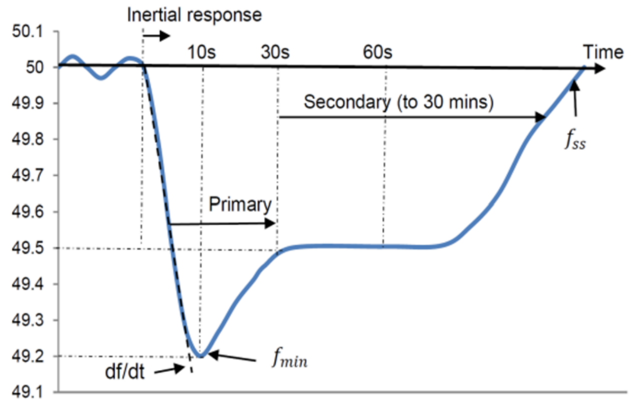


Figure 2: Generation trip/Load addition in a 50 Hz nominal frequency system

Although small perturbations in frequency are expected to occur during normal operation, also called the quasi-equilibrium state, sudden or significant frequency deviations due to generation rejection, load loss, or line trips/faults can cause the system to shift from a normal state to alert or emergency states quickly. These deviations are detected by one or more PMUs through frequency and rate of change of frequency (ROCOF) measurements. In this work, we leverage PMU data that contains captured frequency behavior in the recorded frequency and ROCOF information and use it for studies encompassing detailed analysis of frequency events.

Previous research utilizing PMU data to identify abnormal frequency events has employed dimensionality reduction techniques such as principal component analysis (PCA) [22] and independent component analysis (ICA) [23], while modal features from PMU data were used in ML/AI models such as logistic regression (LR) and SVM to distinguish abnormal frequency scenarios [24]. Other data-driven approaches for frequency event detection can be supervised or unsupervised approaches based on whether they require labels for event identification or classification. Even though unsupervised algorithms can distinguish event clusters [25]-[27], they cannot map real-world interpretation to those clusters [24] which is a considerable disadvantage. Supervised approaches, on the other hand, need carefully designed accurate labels to achieve any meaningful inferences from their use. Supervised approaches can be further divided into physics-based and model-free methods. A combination of several of these approaches can be well-suited depending on the type of application at hand, hence users need to test various approaches for their specific task. Despite these efforts, a comprehensive study to identify and characterize frequency events is lacking. The challenge of insufficient event labeling also affects frequency events [27]. The study facilitated by this project addresses this by decoding frequency events in field data both with and without descriptive event labels from more than one system using TL approach called fine-tuning (FT).

TL approach is specifically suitable for this task among other techniques in ML as alternatives because of availability of synchrophasor data from various utilities that have distinct system design and operational practices (60 Hz US system and 50 Hz European system) and may have inherent differences in data distribution of electrical quantities of interest. Additionally, without comparable data sizes of significant quality which is the case in this project with RTE datasets not being as comprehensive as PNNL, an ML/AI model cannot be trained satisfactorily as an independent entity across systems. Furthermore, it is seldom true that an ML/AI model trained on one system is directly applicable to identify and discriminate similar events in an entirely different system without first learning some subtle unquantifiable differences that system variation has to offer. FT, which is a type of parameter transfer approach under inductive transfer learning, overcomes the aforementioned challenges by updating pre-trained models to decode frequency events on a different dataset than what it was originally trained for. In this work, we use FT by updating pre-trained ML/AI classifiers on simulated data with event instances from field recorded datasets viz. PNNL2 and RTE1+RTE2 to improve the classifiers' generalization ability.

1.1.3 Oscillation Event Detection and Classification

Analyzing oscillatory behavior in power systems is crucial for maintaining the stability and reliability of the grid. These oscillations are inherent to system dynamics and can significantly affect the entire power system [28]. The widespread deployment of PMUs has made real-time grid data readily available. This wide-area data can be used to detect and classify oscillation events based on their characteristics including their frequency, damping, and mode shape. To date, various identification methods including spectral and correlation methods, ringdown methods, and parametric mode estimation from ambient data, have been employed to study oscillation events [29]-[33]. In recent years, ML/AI has made significant progress in classifying problems using extracted features from large historical datasets. A recent paper [34] proposed a framework that transformed multivariate time series data into two-dimensional colored images, which were concatenated into a larger image and input into a convolutional neural network (NN) to classify oscillation events. TL with convolutional neural network was implemented on the medical image classification to overcome the data scarcity problem as well as save hardware resources [35]. A deep neural network (DNN) with a sparse autoencoder was used in [36] to extract features from the measured power system signal for detecting a power quality disturbance. A practical approach for power oscillation classification using real-time identification was presented in [37], where the Hilbert transform was utilized to derive envelope curves as features for training and testing an SVM model. The sub-synchronous oscillation classification was studied using an SVM-based data-driven method [36]. In this report, a TL-based NN that enabled the transfer of extensive oscillation knowledge from simulated datasets to the model is presented. This approach aims to improve the model's performance in analyzing real-world, field-captured events, which, although rare, can have significant impacts and cause considerable economic losses. TL-based NNs extract the features from pixelated oscillation images.

1.2 Report Outline

This report aims to describe the process followed during the course of this project.

- Chapter 2 details the data that was utilized in this study. This includes field-recorded and simulated PMU data.
- Chapter 3 delves into the methods of feature extraction explored for each event type. Feature engineering became the first focus of the project considering the potential benefits of designing features based on domain knowledge.
- Chapter 4 discusses the choice of ML/AI models, and later, TL models. First, ML/AI models were developed using simulated data and event-rich field-recorded data. Then, TL methods were investigated for transferring the knowledge learned by these ML/AI models to smaller, less diverse datasets.
- Chapter 5 evaluates the results, performs subsequent analysis, and outlines the lessons learned both generally and specifically for each event type.
- Chapter 6 presents the conclusions and insights gained from this project.

2. Data Availability, Preprocessing, and Applications

2.1 Introduction

Data availability and quality are the most integral factors for successful ML/AI model development. An ML/AI algorithm, whether supervised or unsupervised, benefits from the availability of large quantities of data that is diverse enough to give the (ML/AI) model numerous instances of each class type. Event detection and classification models have long depended on simulated PMU data for the possibility of creating balanced datasets that contain similar instances of each scenario that the models need to identify that were not captured by the field-recorded data. Also, simulated data cannot fully replicate the dynamics and disturbances that may exist in field-recorded data, which may lead to high levels of error when applied directly to field-recorded data. Conversely, field-recorded data cannot solely solve this dilemma due to the imbalance that exists in the different event classes.

To elaborate, most of the field-recorded PMU data represent normal operation, and even when event-related data is isolated, a clear imbalance is found between different event classes. One solution is to integrate simulated and field-recorded PMU data to derive a balanced dataset. Data quality is another important consideration second to data availability. Section 2.2 discusses the steps taken to ensure that the PMU data is ready to use with the ML/AI algorithms. That process entails performing statistical analysis to reveal any bad data as well as checking the quality of the event labels, if any are provided. Section 2.3 presents the qualities of the available field-recorded PMU data. Section 2.4 then explains the choices made to simulate event type-specific data and use it together with field-recorded data to train and test task-specific ML/AI models.

2.2 Data Preprocessing

Data preprocessing is essential for ensuring that accurate and reliable data is available for training ML/AI models. The process involves several steps to clean, transform, and prepare the data for further analysis. The cleaning process involves the handling of inaccurate time-stamping and missing PMU measurements. Inaccuracies in event timestamping were resolved by visually inspecting and manually adjusting the timestamps. The PMU signals with 60% or more missing values were dropped from the analysis. The missing measurements or the not-a-number (NaN) values were replaced by their estimates using the linear interpolation method [38]. The preprocessing method also addresses data integrity, mislabeling, and incompleteness due to sparse PMU locations by detecting and correcting inconsistencies, relabeling data accurately, and compensating for gaps in data coverage. The cleaned synchronized measurements were normalized and standardized before feature extraction.

2.3 Field-recorded PMU Data

The field-recorded PMU measurements were collected from different regions of the United States and the Réseau de Transport d'Électricité (RTE), the France-based transmission system operator.

The measurements were used to detect and classify frequency events, line faults, and oscillation events. Table 1 summarizes the details of the available datasets. PNNL1 and PNN2 datasets were obtained from the Pacific Northwest National Laboratory (PNNL). The datasets consist of two years (2016 and 2017) of PMU readings from two U.S. interconnections. The events in the PNN1 datasets were labeled after various data quality issues were resolved by the data preprocessing methods discussed in subsection 2.2. The PNNL2 datasets were not accompanied by event log files. These datasets were used for the analysis of frequency and oscillation events. The possible frequency events were labeled using the steps outlined in subsection 2.3.1. Probable ringdown oscillation events were found using the standard deviation approach applied to sliding data analysis windows [39]. The third dataset consists of 118 PMU measurements from the WECC. These WECC datasets were used to study the oscillation events, and they included two generator brake test events, each leading to system-wide oscillations. These oscillation events were classified as well-damped using physics-based approaches such as Prony, matrix pencil, and eigen-value realization.

Table 1: Field Recorded Datasets

Source	PNNL1	PNNL2	WECC	RTE1	RTE2
No. of PMUs	43	40	118	10	12
Length	2 years	2 years	30 mins	7 days, 40 mins each	6 days
System Frequency	60 Hz	60 Hz	60 Hz	50 Hz	50 Hz
Data Types	3P V, I Vp, Ip, f, df/dt	Vp, Ip, f, df/dt	3P V, I Vp, Ip, f, df/dt	3P V, I "+, -, 0" seq V, I, f, df/dt	3P V, I "+, -, 0" seq V, I, f, df/dt
Reporting Rate	30, 60 fps	30 fps	30 fps	50 fps	50 fps
Availability of Labels	Yes	No	No	Yes	Yes
Application	F	Fr, O	Fr, O	F, Fr, O	F, Fr, O

RTE provided the fourth and fifth datasets, designated as RTE1 and RTE2. The RTE1 dataset consists of 10 PMU measurements over seven non-consecutive days. There were 16 events recorded in the RTE1 dataset. Despite being smaller in scale compared to the other datasets, the RTE dataset furnished critical insights into PMU locations, facilitating in-depth event location studies. RTE2 dataset was unlabeled and was primarily utilized for the testing of the trained ML/AI models for all three types of events discussed in this report.

2.3.1 Handling absence of labels for frequency events

We define a 'unique frequency event' as an under-frequency or over-frequency event detected by one or more PMUs. However, such events were not labeled in any field PMU dataset. Since the task of labeling events (frequency-related or otherwise) in a huge volume of granular PMU data is arduous, the following steps were taken to obtain the unique frequency events in a dataset that contains captured frequency behavior in the form of frequency and rate of change of frequency (ROCOF) measurements.

Step 1: From all hourly files for a PMU, the minimum and maximum values of frequencies and their indices are obtained. The maximum absolute value of ROCOF and its index is also obtained.

Step 2: The results are sorted in descending order of ROCOF, and a threshold is picked to filter potential frequency events. We use a ROCOF threshold of 0.49 Hz/s to be in alignment with IEEE Std 1547-2018 [40], which states that distributed energy resources (DERs) shall ride through and not trip for a minimum ROCOF of 0.5 Hz/s.

Step 3: The maximum ROCOF index is checked against both the maximum and minimum frequency indices to determine if the minimum or maximum frequency reported occurs at the same time as the maximum value of ROCOF.

Step 4: As frequency in contingencies excluding blackouts, does not deviate by ±5 Hz, events violating this criterion are excluded.

Step 5: For events with maximum and minimum frequencies within close range, another filter is applied where maximum and minimum frequencies occur in around 1-minute intervals to consider all possible power system responses originating from the same initiating event.

Step 6: Lastly, multiple PMUs are grouped together if they show similar responses within 5 timestamps.

Figure 3 shows the steps outlined as a flowchart. These events were labeled, and these labels were used to obtain a prediction from the pre-trained classifiers, and the discussion is presented in detail in Chapter 4.

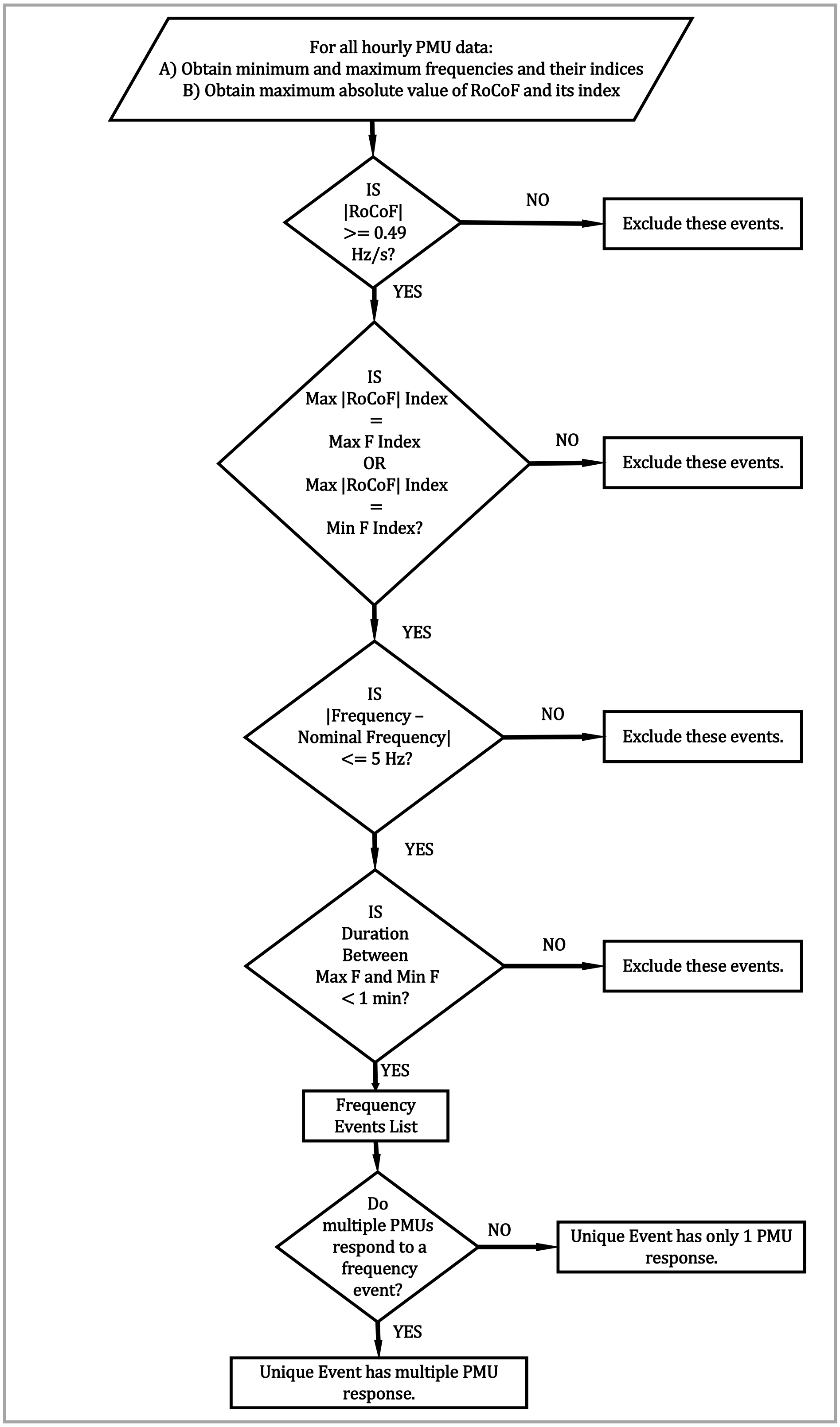


Figure 3: Flowchart to find unique frequency events in a PMU dataset

2.3.2 Duration of Unique Frequency Events

A sliding window over the frequency data around the occurrence of unique frequency events is used to compute the approximate starting point, ending point, and duration of unique frequency events thus obtained. A simple flowchart of the process to compute the duration of the unique frequency events is outlined below in Figure 4.

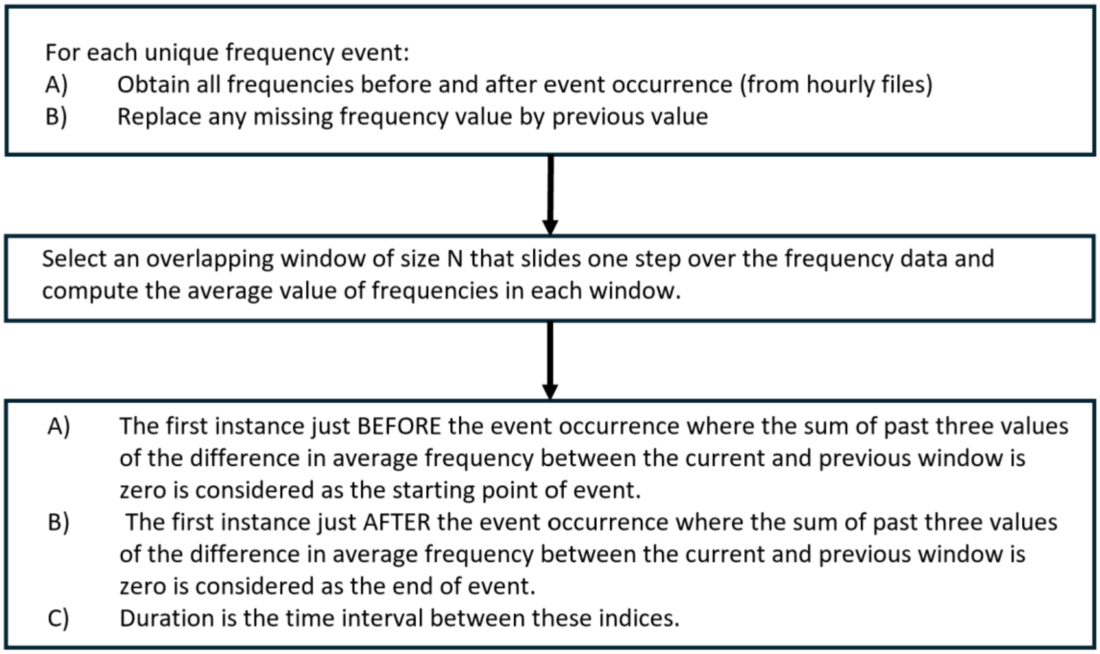


Figure 4: Flowchart to compute start point, end point, and duration of frequency events

2.4 Simulated Data

Simulated datasets were created to facilitate the training process of the ML/AI models since the field-recorded datasets may not contain a sufficient number of classes of events. The field-recorded datasets are also known to often miss event labels. Simulated event datasets were generated with precise and accurate labels, which significantly enhanced the training process of supervised ML/AI models. Unlike reinforcement learning and unsupervised learning, supervised learning utilizes the labels for the training. These datasets also provide clear and well-defined examples of events, allowing the models to learn from accurately labeled data, thereby improving their ability to classify and predict similar events in real-world scenarios. However, these datasets may not capture the same noise and dynamics of field recorded data. Thus, simulated data was used in conjunction with field data (and not as a replacement of field data), for each critical event type as described below.

2.4.1 Simulated Data for Line Faults

An obvious data imbalance exists in the types of line faults that occur in field-recorded PMU data. This data imbalance is attributed to the fact that three-phase (TP) faults are the rarest fault type, followed by line-to-line (LL) and line-to-line-to-ground faults (LLG). The most commonly occurring fault type is single-line-to-ground (SLG) faults. Such a data imbalance may negatively affect the performance of any supervised ML/AI algorithm, as it is not provided with enough samples of every class that it is meant to detect and classify. Hence, simulated datasets were generated to improve the training data by compensating for the classes that are more rarely seen. A PSCAD [41] simulation based on a model that was designed by an IEEE Power System Relaying and Control Committee (PSRC) working group titled "EMTP Reference Models for Transmission Line Relay Testing" [42] is chosen for this simulation. This model perfectly fits the requirements of the simulation as it was originally developed to test relays under realistic power system conditions. Utilizing the model shown in Figure 5, a series of fault simulations were conducted, adjusting PMU locations and systematically varying fault types along the lines to produce a diverse set of fault scenarios.

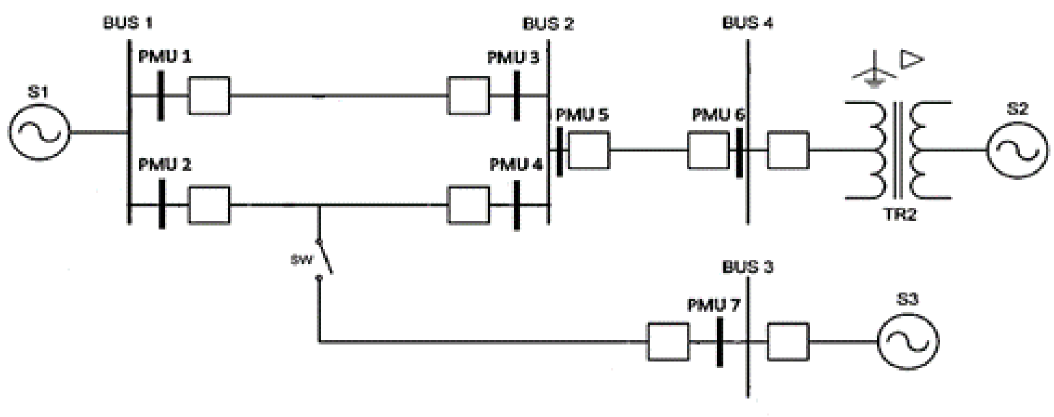


Figure 5: PSCAD simulation model

The ten fault types listed in Table 2 were simulated on each of the four lines repeatedly at 10% increments of the line length to create a comprehensive training dataset. Based on this team's experience in previous line fault studies [6]-[7], the LL and LLG faults were combined and represented using a single label when preparing the training dataset. This decision came after extended testing of several classifiers and the conclusion that these fault types are not easily distinguishable in the field-recorded PMU data [6]. For the same reason, three-phase and three-phase-to-ground faults were represented using only three-phase-to-ground faults. The result was a set of 400 line fault events whose PMU data was collected at 7 different PMU locations. As expected, the signature left by the line fault varied from PMU to PMU due to their locations relative to the location of the fault. These differences, which are illustrated in Figure 6 inspired a sensitivity study to reveal the effects of PMU location on the performance of an ML/AI model trained using the simulated data and tested using field-recorded PMU data.

Table 2: Simulated Line Fault Types

Fault Number	Fault Type	Final Label
1	A-G	1
2	B-G	2
3	C-G	3
4	AB	4
5	BC	5
6	CA	6
7	AB-G	4
8	BC-G	5
9	CA-G	6
10	ABC-G	10

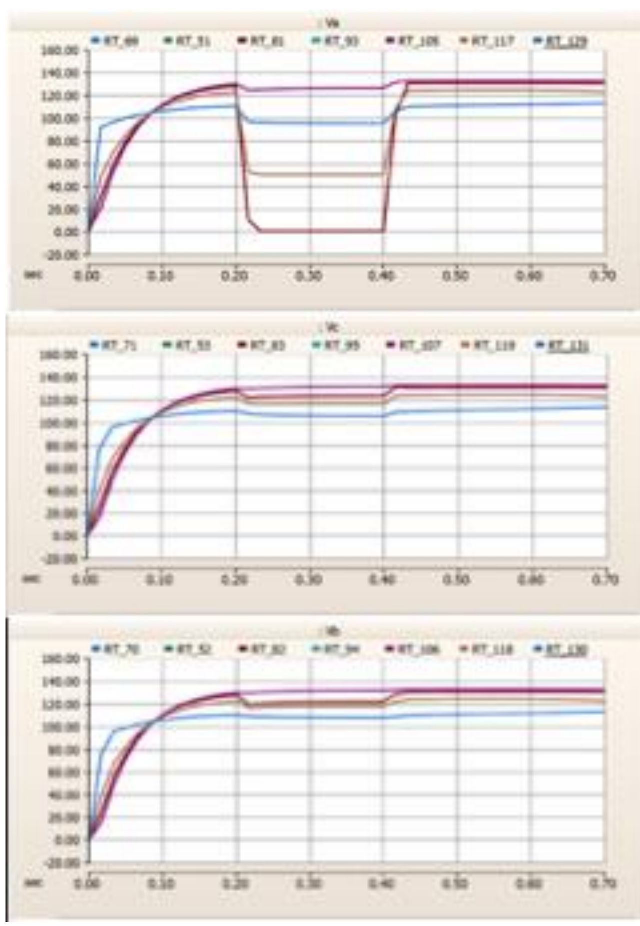


Figure 6: Example of A-G fault as seen by 7 PMUs at different locations

To test the sensitivity of ML/AI models to the PMU location from which simulated data is collected, seven training datasets were created, one for each PMU location shown in Figure 5. Such a study is important for line fault detection since line faults are local events. If a PMU is located further away from a fault, the impact of the fault on the PMU measurements may not be significant enough to detect and classify the fault accurately. PNNL1 data was used for testing during this sensitivity study considering that it contained a more diverse set of line faults compared to the other field-recorded datasets. Because the goal of this study is to test the sensitivity only with respect to PMU location, the ML/AI model was chosen to be a SVM model, which has proven to be the most successful ML/AI algorithm during previous line fault detection studies [4]. The lines on which the faults were simulated were also limited to lines 1 and 2 between Bus 1 and Bus 2 to create more distance between the simulated faults and certain PMUs. The results of this sensitivity demonstrated in Table 3 indicates that PMU 7 is the optimal location for collecting data to train a ML/AI model intended for use on a different dataset. This is likely due to PMU 7's strategic positioning at the midpoint between Bus 1 and Bus 2, where the line faults for this part of the experiment were created. PMU 7's position may have allowed it to observe faults in a more balanced and uniform manner. In contrast, other PMUs might perceive some faults more intensely due to their closer proximity to these events. The main takeaway from this sensitivity study is that if only one PMU was available for data collection, the best location for such a PMU would be at the location of PMU 7.

Table 3: Results of PMU Location Sensitivity Study

Training Data (PMU)	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Weighted F1-Score	0.90	0.91	0.91	0.90	0.90	0.91	**0.92**

2.4.2 Simulated Data for Frequency Events

To simulate frequency events and gather corresponding events' dynamic data, GE's (General Electric) PSLF software was used [43]. The test system is a large WECC test case containing 18457 buses, 3360 generators, and 9023 loads (see Table 4). Contingencies such as generator loss, load loss, and line faults along with normal scenarios were simulated to obtain valuable data required to complement the field-recorded data and train suitable ML/AI models for the detection and classification of frequency events.

Table 4: Summary of the large WECC test case

Swing Bus Summary			
Bus	**ID**	**MW**	**MVar**
30000 PTSB 7 20.00	1	547.098	219.461

Equipment Totals					
Type	**Count**	**Max. Count**	**Type**	**Count**	**Max. Count**
Buses	18457	125000	DC buses	12	200
AC line Sections	15760	150000	DC lines	9	100
Lines w/cond geometry	0	1500	DC Converters	8	200
Transormers	7170	50000	Areas	21	10000
Generators	3360	40000	Zones	454	10000
Motors	0	200	Owners	284	2000
Loads	9023	75000	Interfaces	69	1000
Shunts	1439	80000	Interface Branches	292	5000
Static VAR devices	989	20000	Transactions	44	1000
Power Electronic Devices	0	50	Impedance Tables	20	125
VSC DC lines	0	50	Generator Q Tables	31	3000
Breakers	0	250000	Substations	0	25000

All the event dynamics were simulated for a total of 10 seconds (with 2 seconds of flat start followed by 8 seconds of event dynamics). To capture the event spread, electrical quantities such as positive sequence voltage magnitudes, positive sequence voltage angles, frequency, and RoCoF were monitored (recorded) from all the neighboring buses up to three hops from the point of contingency. The reporting rate used for simulation was 30 frames/s similar to US field PMU recordings available for the project. Table 5 shows the number of simulated events for each contingent category of frequency events.

Table 5: Number of simulated events for frequency event categories

Type	Number	Label
Normal Scenario	5430	1
Generator Loss	2478	2
Load Loss	8749	3
Line Fault	4705	4
Total number of events = 21362		

For the brevity of presentation in this document, plots of four electrical quantities (from PSLF) from neighboring buses are shown in Figure 7 for an instance of generator loss. The generator loss event is the loss of a large generator in WECC which had a power generation of 1410 MW.

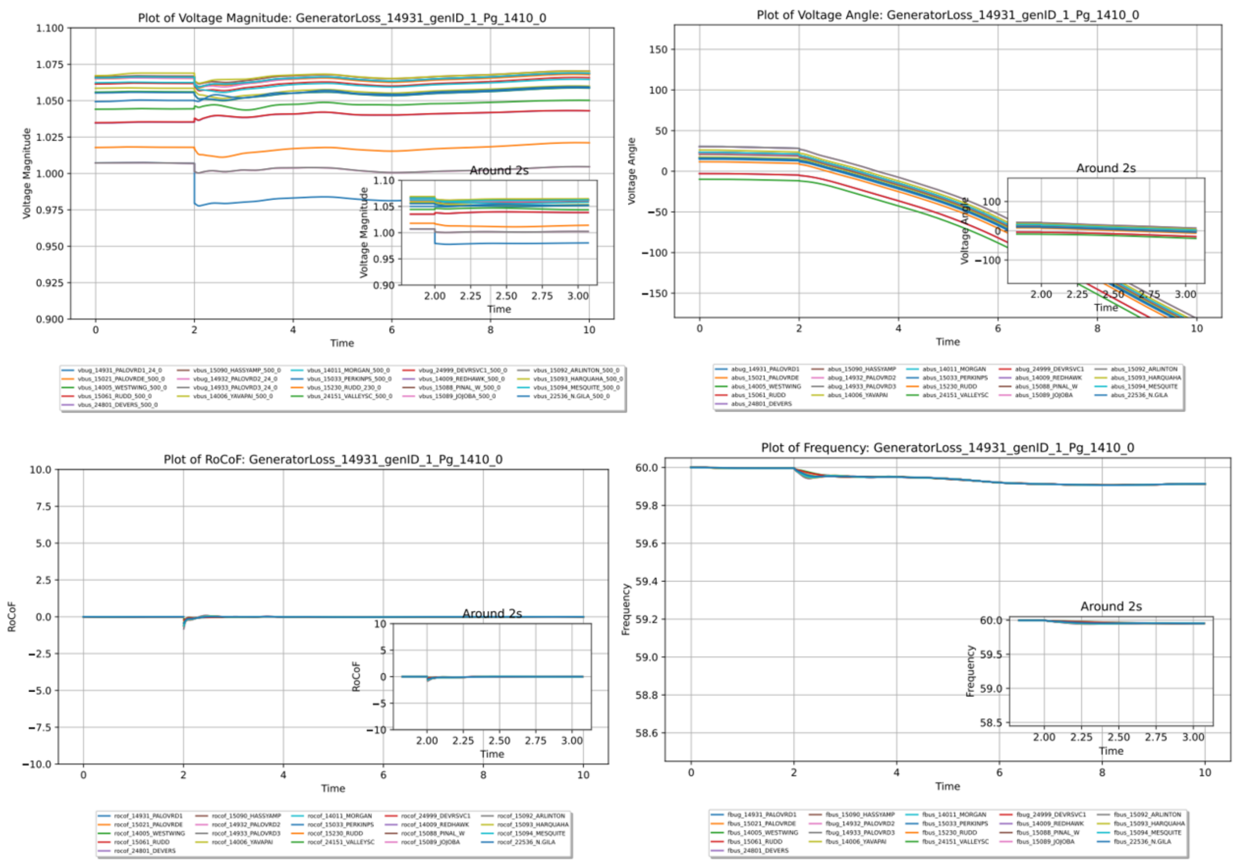


Figure 7: Positive sequence voltage magnitude and angle, frequency, and RoCoF of neighboring buses for a generator loss scenario at Paloverde in a large WECC test case

2.4.3 Simulated Data for Oscillation Events

Field-recorded measurements often do not encompass a sufficient variety of oscillation events, leading to datasets that may not fully represent the range of scenarios required for robust training. Simulated datasets supplement field data by providing comprehensive, controlled, and diverse scenarios required for effective training. In the oscillation study, simulated datasets were generated by providing a step input to a second-order system as shown in the equations below:

$$Y(s) = \frac{\omega^2}{s^2+2\zeta\omega s+\omega^2} X(s) \quad (1)$$

$$x(t) = L^{-1}[X(s)] = 1(t) + \phi(\mathrm{t}) \quad (2)$$

$$\tilde{y} = L^{-1}[Y(s)] + \Psi(t) \quad (3)$$

In the equations above, $X(s)$ shows the input signal in the Laplace domain, L^{-1} represents the Laplace inverse function. Similarly, $\phi(\mathrm{t})$ represents white Gaussian noise, $\Psi(t)$ is a measurement noise, and ω stands for the natural frequency of the system. For a closer resemblance to field-recorded conditions, a 10% noise label was incorporated into the simulated training datasets. Similarly, to enhance the diversity of oscillatory scenarios, the frequency band was randomly varied between 0.1 Hz and 1.5 Hz, and the damping ratio was randomly adjusted from 0 to 100 to create a vast library of oscillation events with known labels. These training datasets were evaluated as a part of the initial research [44].

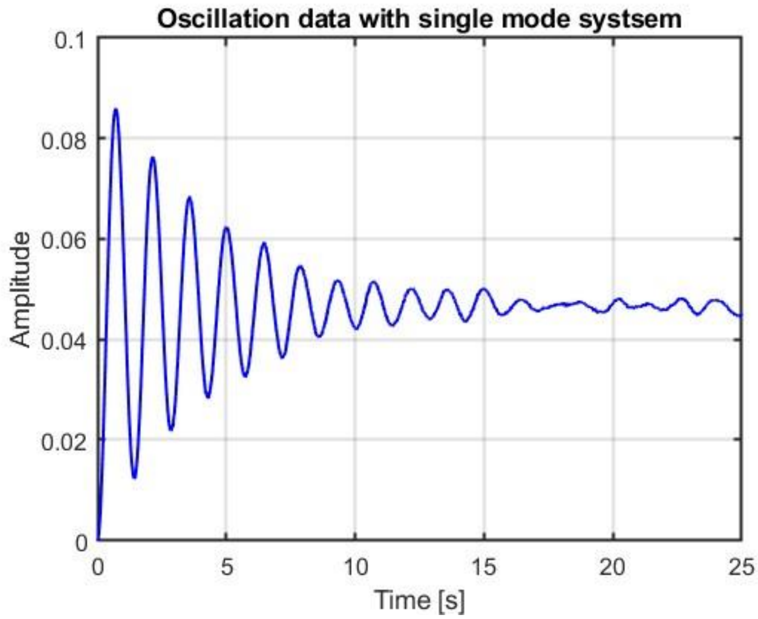


Figure 8: Training oscillation dataset with single-mode

In subsequent studies, improvements were made to these datasets by incorporating multiple modes in the system. A step input to a fourth-order system was used to generate data associated with two oscillatory modes with varying frequency and damping ratios as shown below:

$$Y(s) = H_1(s)H_2(s)X(s) \tag{4}$$

where, $H_1 = \frac{\omega_i^2}{s^2+2\zeta_i\omega_i s+\omega^2}$ and $H_2 = \frac{\omega_j^2}{s^2+2\zeta_j\omega_j s+\omega^2}$

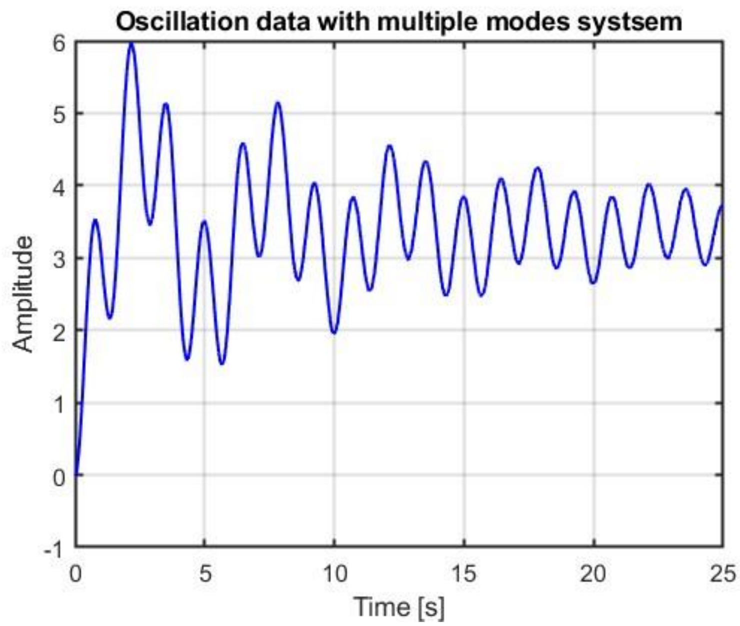


Figure 9: Training oscillation dataset with multiple modes

2.5 Summary

PMU measurements from different regions of the US power system and the RTE system were used in this study. The measurements suffered from data quality issues that were handled using different data preprocessing methods. The field-recorded datasets do not sufficiently contain all classes of events and often have missing event labels. A wide range of simulated event datasets were generated for line fault events, frequency events, and oscillation events with precise and accurate labels, in order to facilitate the training process of the ML/AI models described in subsequent chapters.

3. Feature Engineering

3.1 Introduction

Various types of features may be extracted from PMU data, but this project focused on utilizing the domain knowledge and experience of the team to extract features that carry the most useful information for each application. Line faults, frequency events, and oscillation events create different signatures in PMU data with regard to the intensity, type of response, and duration. For instance, oscillation events generally happen over longer periods of time than line faults and frequency events. Similarly, distinguishing different line faults requires the use of three-phase voltage measurements. Finally, features extracted from frequency measurements alone may not be sufficient for frequency event analysis. This chapter demonstrates the feature engineering process for each critical event type.

Distinct features were extracted from the PMU data for each event type depending on the expected characteristics of the PMU measurements during these events. Features for line faults were extracted from three-phase voltage measurements, while oscillation event features were extracted from frequency measurements, for instance. Frequency event features were extracted using positive sequence voltage magnitude and phase angle in addition to frequency and ROCOF measurements. Having such diverse feature extraction methods required the team to consider several ML/AI algorithms to investigate the one that worked best for the features at hand. Subsections 3.2, 3.3, and 3.4 describe the feature extraction process for line faults, frequency events, and oscillation events, respectively.

3.2 Feature Engineering for Line Faults

The success of any ML/AI model largely depends on the quality of the features fed into it. For line fault events, physics-based features derived from power system protection principles were utilized [6]-[7]. These features were calculated as the normalized sum of the differences between the maximum and minimum voltage magnitudes recorded within each 2-second time window, as illustrated in (5) and (6).

$$\Delta(V\emptyset) = max(V(\emptyset)_{MAG}) - min\,(V(\emptyset)_{MAG}) \qquad (5)$$

$$\mathrm{SUM}(\mathrm{V}\emptyset) = \sum_{\mathrm{i=1}}^{\#\mathrm{PMUs}} \frac{\Delta(\mathrm{V}(\emptyset))}{\text{Number of PMUs}} \qquad (6)$$

This feature computation was chosen knowing that the most significant aspect of the PMU measurements during a line fault is the voltage drop seen as a result and visualized in Figure 10. This voltage drop, when averaged across all the available PMUs, can become more significant, slightly mitigating the issue of PMU location relative to the line fault. Averaging also ensures that the features are normalized, which is especially important since the field-recorded datasets include PMUs at different nominal voltage levels. Line faults occurring at lower voltage levels may have otherwise been misclassified due to having lower impact on the data than those at high voltage levels. The time window for feature extraction was chosen to be a 2-second sliding time window.

This decision results from extensive previous studies in which 5-minute, 1-minute, 10-second, 2-second, and 1-second time windows were explored [6]. The 2-second time window is where the performance of the ML/AI model plateaued and was consequently set as the most efficient choice. Figure 10 highlights a 2-second window in which the maximum and minimum voltage values are obtained for each PMU in the PSCAD model as shown. The final features are then computed across all PMUs and for each phase separately.

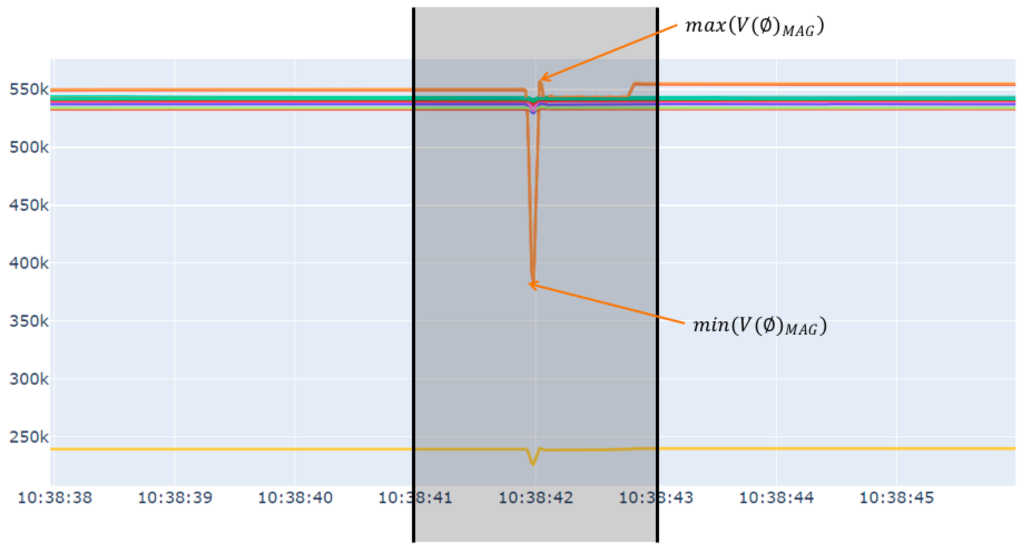
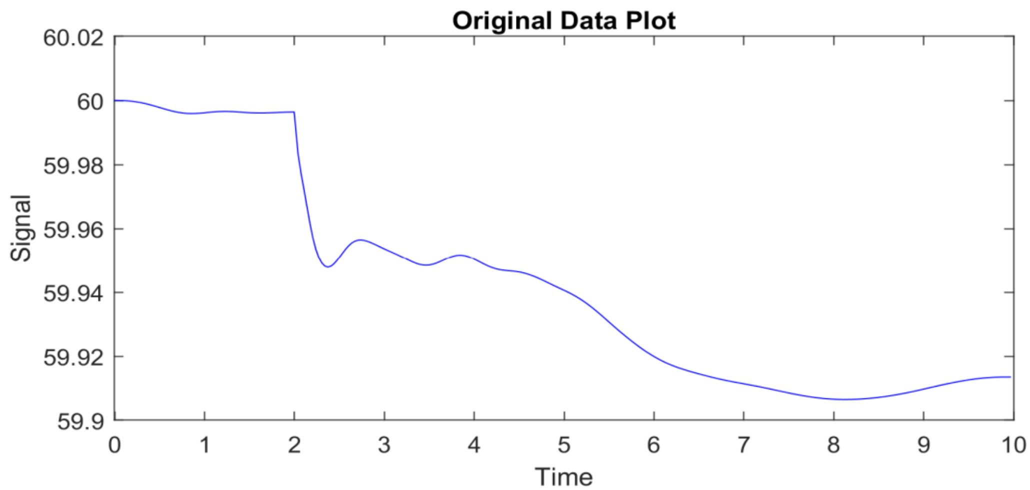


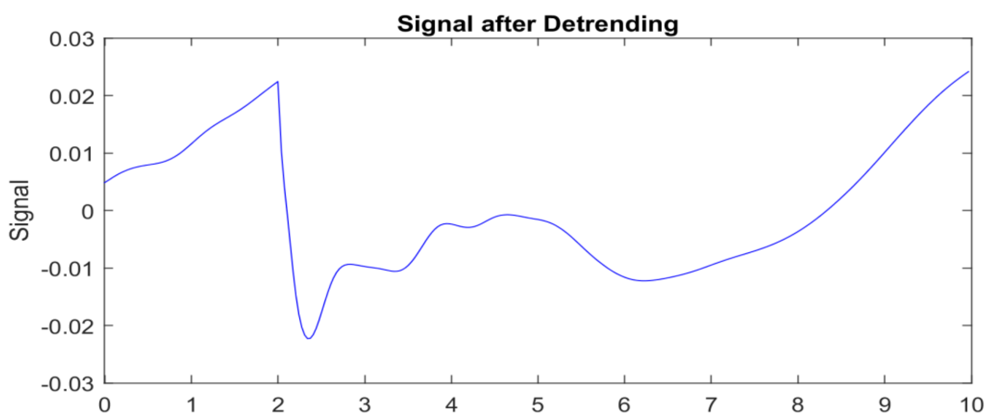
Figure 10: Visual representation of line fault feature computation

3.3 Feature Engineering for Frequency Events

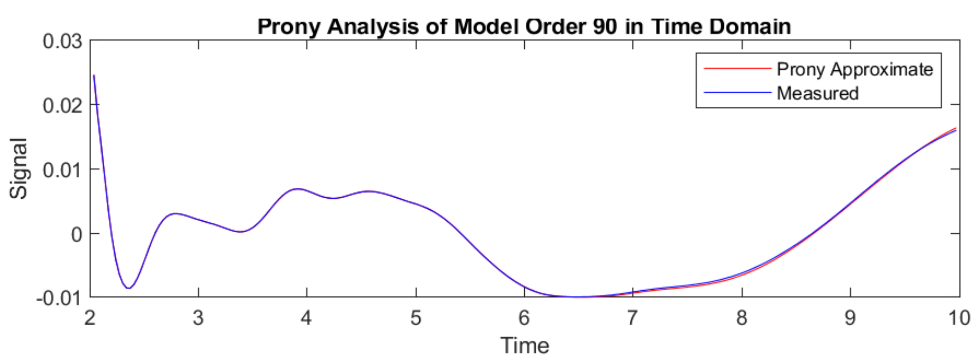
As the temporal effects in a power system are driven by the interacting dynamics of system components [24], features for this application are obtained from modal decomposition of the relevant electrical quantities such as positive sequence voltage magnitudes and phase angles, frequency, and RoCoF, all recorded by PMUs. Modal decomposition explains the contents of a PMU data stream after an event by identifying the underlying dominant modal features. We use the Prony method [45] to obtain the most dominant modes from each signal of interest. An instance of frequency signal reconstruction and its intermediate steps during generator loss at a bus in the WECC system using the Prony method is shown in Figure 11.


Original Data Plot
60.02
60
59.98
59.96
59.94
59.92
59.9
Signal
0
1
2
3
4
5
6
7
8
9
10
Time

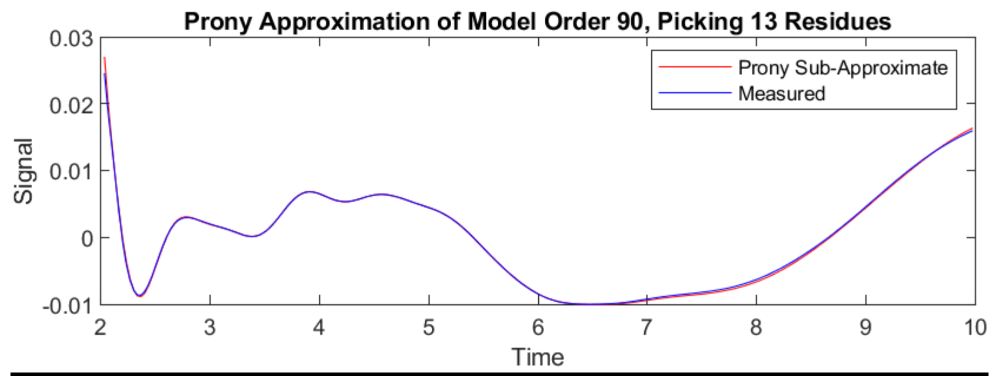
11(a)


Signal after Detrending
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
Signal
0
1
2
3
4
5
6
7
8
9
10

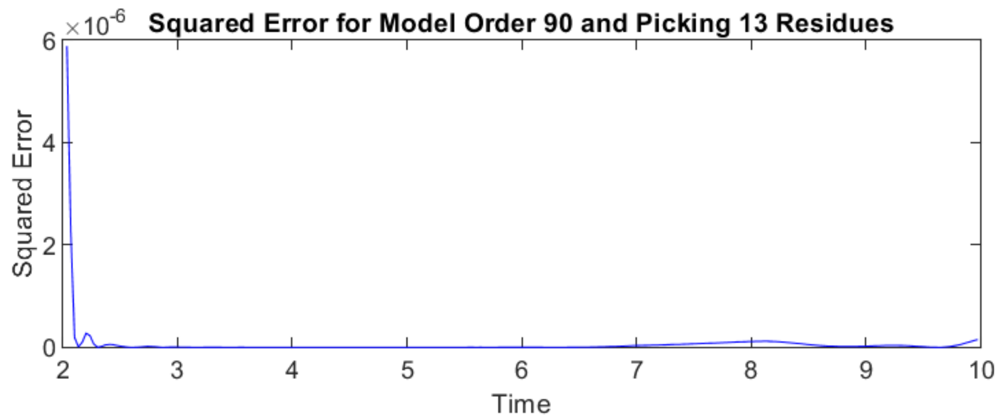
11(b)


Prony Analysis of Model Order 90 in Time Domain
Prony Approximate
Measured
0.03
0.02
0.01
0
-0.01
Signal
2
3
4
5
6
7
8
9
10
Time

11(c)



11(d)



11(e)

Figure 11: Frequency signal reconstruction using modal decomposition for a generator loss event (Figures 11(a) – 11(e))

Table *6* shows the mode numbers along with their amplitude, damping, frequency, and energy. These modal parameters were used to reconstruct the post-disturbance frequency signal recorded at the bus considered as shown in Figure 11.

Table 6: Dominant Modes Used for Signal Reconstruction in Figure 11 and their details

Mode #	**Amplitude**	**Damping**	**Frequency**	**Energy**
1	9.2e-03	-3.7e-01	3.6e-01	8.7e-04
2	9.2e-03	-3.7e-01	3.6e-01	8.7e-04
3	8.7e-03	-2.1e+01	0.0e+00	2.5e-05
4	7.7e-03	-1.0e+00	8.7e-01	2.3e-04

5	7.7e-03	-1.0e+00	8.7e-01	2.3e-04
6	6.9e-03	-1.1e+00	1.1e+00	1.7e-04
7	6.9e-03	-1.1e+00	1.1e+00	1.7e-04
8	5.6e-03	-5.1e-01	5.5e-01	2.3e-04
9	5.6e-03	-5.1e-01	5.5e-01	2.3e-04
10	5.0e-03	-1.1e+00	1.5e+00	8.6e-05
11	5.0 e-03	-1.1e+00	1.5e+00	8.6e-05
12	4.6e-03	1.6e-01	1.5e-01	6.1e-03
13	4.6e-03	1.6e-01	1.5e-01	6.1e-03

The features used for this work are the frequency and damping ratio of the dominant modes followed by the amplitudes and energies of the residual coefficients of those modes. Mathematically, this can be expressed as:

$$F_{\text{param}} = [\text{Frequency}, \text{Damping}, \text{Amplitudes}(1 \ldots \#\text{PMUs}), \text{Energies}(1 \ldots \#\text{PMUs})] \tag{7}$$

where, param is a variable that represents one of the four signals used to generate modal features: positive sequence voltage magnitude and voltage angle, frequency, and RoCoF, i.e., param ∈ {V_{mag}, V_{ang}, Frequency, RoCoF}. Note that F_{param} will be a row vector of size 1-by-DIM where DIM = N + N + N*#PMUs + N*#PMUs, and N represents the number of dominant modes selected. Finally, the feature row vector (*FRV*) for an event will have a size of 1-by-(4*DIM), which is mathematically written as:

$$FRV = \left[F_{\text{Vmag}}, F_{\text{Vang}}, F_{\text{Frequency}}, F_{\text{RoCoF}}\right] \tag{8}$$

3.4 Feature Engineering for Oscillation Events

For oscillation events, the feature engineering was formulated as an image recognition problem for classifying oscillation events. An oscillatory pixelated image was created by using the positive voltage magnitude signal, but flexibility also exists to utilize signals such as active power flows, reactive power flows, and bus voltage phase angle. The simulated datasets were used to create pixelated oscillation images for training the ML/AI models. The pixelated image captures the damping and frequency characteristics of the oscillation event. In initial work, single-mode oscillation training datasets were used to create pixelated images as shown in Figure 12.

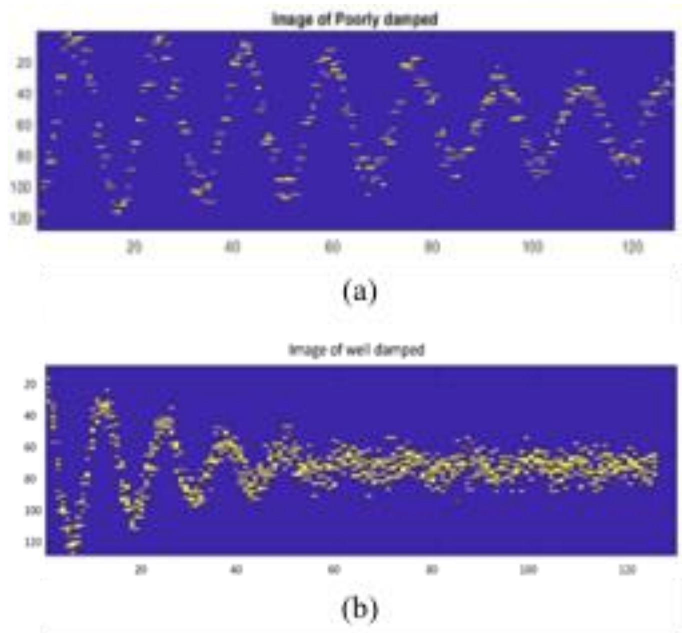


Figure 12: (a) Poorly damped pixelated oscillation image with single mode (b) Poorly damped pixelated oscillation image with single mode

Figure 12 shows the pixelated images for the poorly damped (a) and well-damped oscillation (b). The image size of each pixelated oscillation picture is 120x120 pixels. In further study, the simulated training data was made more comprehensive by designing signals containing multiple modes. The pixelated oscillation for a multiple-mode signal is shown in Figure 13. The pixelated images are used as a feature for training the ML/AI approach for oscillation classification.

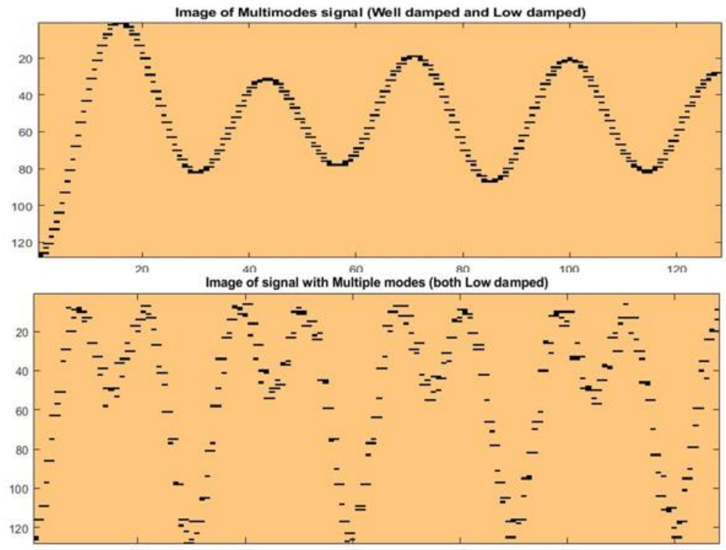


Figure 13: Pixelated oscillation image with multiple modes system

3.5 Summary

Although the PMU data can be fed directly to an ML/AI model, extracting relevant contrasting features for various event types is usually preferred. In this project, the designed features are chosen to provide appropriate representation of the underlying application. Features for line faults leveraged the principles of protection studies and represented event impact on voltages in the network. Next, features for frequency events leveraged the principles of modal decomposition signifying wide interacting dynamics in the system generally associated with frequency. Finally, features for oscillation events leveraged principles of the image recognition problem as it helped the qualitative analysis of oscillations into well-damped versus poorly damped types. This process of feature engineering across applications signifies the carefulness needed in selecting physics-based features that data-driven applications using ML/AI lack at times.

4. Data Model Development, Training, and Validation

4.1 Introduction

The choice of the ML/AI algorithm and training dataset affects the overall performance of the final model. Since different features were chosen for each event type in this study, the ML/AI algorithms that work best with the associated feature type were examined first. That is, each critical event classification problem was treated as a separate task initially. Then, the possibility of adapting one solution to tasks different from the one it was developed for was considered. Different supervised learning-based classifiers were considered after overcoming the scarcity of labels using simulated data. In addition to creating a balanced training dataset using simulated data and integration of simulated and field-recorded data, testing datasets for each event type were chosen based on the availability of instances of the respective event type. Section 4.2 discusses the process of developing the base classifiers for each event type.

The new datasets that were obtained from industry partners provided a unique opportunity for validating the new ML/AI models on different datasets. However, the number of instances of each event type and their subcategories was limited in these new datasets. In other words, it would not be possible to retrain a new ML/AI model for each new dataset. This situation is expected since PMU data might not be readily available for extended periods, and even large amounts of historical PMU data may not be accompanied by accurate labels. The most practical solution is to then train the base classifiers developed using appropriate datasets to be transferred to new datasets while maintaining their performance. Several transfer learning (TL) algorithms were considered for each event type, recognizing the differences in features and base classifiers. Section 4.3 provides the details of every TL algorithm, and the datasets used for training and testing. Finally, in Section 4.4, the adaptability of these TL algorithms is tested by applying them to the alternative event types.

4.2 ML/AI Model Development

4.2.1 Line Faults

The base ML/AI model created for the detection and classification of line faults largely builds upon the team's experience studying PNNL1 data in a previous project [6]-[7]. During that extensive study, the dataset was analyzed for daily statistics of each measurement type in order to reveal the PMUs that exhibit high amounts of missing data and unreasonable values. The waveforms corresponding to the events listed in the event log were visually inspected in order to correct any inaccuracies in timestamps and event labels. This extended process proved to significantly improve the performance of any ML/AI model, not only the ones targeting line faults. The effects of these two crucial preprocessing steps are detailed in [3], [6]-[7], [16], and [46]. The feature engineering and selection of ML/AI classifiers were guided by the top-performing methods identified in [6]-[7].

Preprocessing the RTE datasets involved eliminating continuous missing data, NaN values, and faulty PMU signals. A sliding data window technique was employed, and any windows that surpassed predefined missing data thresholds were excluded from further analysis. A detailed description of the final set of features extracted for every 2-second time window are explained in Section 3.2 of this report. Additionally, SVM-based models were found to be most successful when classifying line faults in previous studies [6].

The main goal of this stage of the project for the category of line faults analyzed, was to validate the utility of the methods that resulted from previous studies [6]-[7], and introduce the new simulation model for line fault data. The new model, developed in PSCAD, was expected to improve the performance of the model by providing a closer representation to real PMU data. However, this assumption required validation through testing. Therefore, the first test was performed by comparing the results of using the new simulated data from PSCAD to train the SVM model versus using the simulated dataset developed for the previous project [6]. The RTE1 dataset was utilized for this validation as a testing dataset. According to the weighted F1-score, the ML/AI model trained with the new simulated data achieved an accuracy of 0.91, outperforming the previous simulated dataset, which had an accuracy of 0.88.

The use of simulated data to support the training process here is approached as an enhancement to the regular supervised learning approach that may have only utilized the field-recorded data. Supervised learning was preferred over unsupervised or reinforcement learning due to the availability of some labels, despite the need of further processing due to possible inaccuracies that exist in the event logs, and the knowledge that the team retains on the targeted events. Given that the new PSCAD dataset was more useful as a training dataset, the next question addressed was whether it was sufficient. To answer this question, the PNNL1 dataset was brought in as a candidate training dataset. The experiment was repeated three times by varying the training dataset between PSCAD, PNNL1, and an integration of both. The details of this stage of the experiment are as follows:

- Time-window: 2 seconds
- Features: voltage difference-based features described in Section 3.2.
- ML Algorithm: SVM
- Testing dataset: RTE1
- Training dataset: PSCAD only / PNNL1 only / PSCAD+PNNL1

4.2.2 Frequency Events

After extracting the modal features obtained as explained in Section 3.3, five open-source ML algorithms were applied in a supervised setting using off-the-shelf packages in Python. Supervised machine learning algorithms were selected over unsupervised or reinforcement models for this task considering the availability of labeled frequency events in the simulated dataset from PSLF, and the advantages that supervised learning offers in terms of increased interpretability in mapping input features to output and their scalability with big data problems. Each of the algorithms used in this study is briefly explained next.

1) Support Vector Machine (SVM)

SVM is one of the most effective algorithms in supervised ML. Its core objective is to identify an optimal hyperplane that effectively segregates two distinct classes. Given the multiclass nature of line fault classification, the implementation of SVM necessitated a conversion into a series of binary classifications, employing the one-vs-one strategy. This approach involved deploying separate classifiers, each responsible for differentiating between a pair of classes. By combining the outcomes of these binary classifiers, a robust multiclass classification framework was established [47].

2) Random Forest (RF)

RF is a type of "ensemble" classifier. The architecture of RF consists of numerous decision trees (DTs), each evaluating incoming events independently. Every tree contributes a singular vote towards classifying each input under the category it deems most likely. The ultimate classification decision is derived from the aggregation of these votes, with the class receiving the majority of votes being chosen [48]. Critical to the optimization of the RF model are the hyperparameters, such as the number of trees in the forest and the maximum depth of these trees, which require careful selection tailored to the specifics of the problem at hand. The choice to use the RF classifier in our research is attributed to its quick processing capabilities, resilience against noisy data, and its proficiency in detecting nonlinear relationships within the dataset.

3) Extreme Gradient Boosting (XGBoost)

XGBoost is a gradient boosting algorithm that sequentially adds DTs to an ensemble to improve on previous errors. It can handle imbalanced data effectively, assign higher weights to misclassified samples, and adjust the class distribution during training [49]. This capability makes XGBoost particularly useful in scenarios where the classes are unevenly distributed, as it can adapt its learning process to give more attention to minority classes, leading to improved model performance and better generalization on imbalanced datasets.

4) K-Nearest Neighbors (kNN)

*k*NN labels an unknown instance by utilizing the distance of the point to the labeled data. The distances are usually defined by a distance metric such as the Euclidean distance or Manhattan distance. The choice of the value of k can define underfitting, overfitting, or optimal classification, and is tuned to solve a specific problem at hand. For a binary classification problem, it is usually advantageous to pick an odd number for the value of *k*, to overcome the ambiguity of information from the neighbors.

5) Gaussian Naïve Bayes

The Gaussian Naïve Bayes algorithm uses Bayes' theorem by assuming the likelihood of features to be Gaussian to classify different instances of data.

4.2.3 Oscillation Events

A possible oscillation event was found in the measured field recorded measurements using a sliding window approach. A ratio of the standard deviation of the ringdown event window to the standard deviation of the ambient window was calculated in a sliding window fashion as shown in Figure 14.

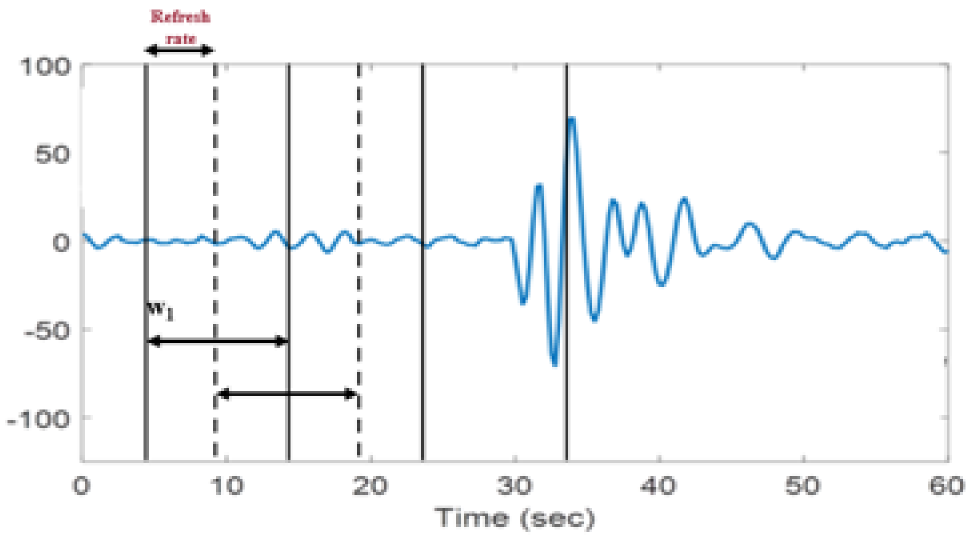


Figure 14: Sliding window analysis

Once the standard deviation ratio threshold exceeded the set threshold for five consecutive windows, the datasets were considered for the oscillation study. The recorded ringdown events were then converted into pixelated images as discussed in Chapter 3.4. The supervised neural network (NN) approach is considered best suited for image classification and has been explored in our study to classify oscillation events based on their frequency and damping characteristics. Unlike Unsupervised and RL, NN utilizes the labels of the oscillation for effective training. The simulated datasets were created with accurate labels which aided the learning process of NN. An image-based oscillation detection classification process was studied using the following supervised learning methods.

1) Multi-layer Neural Network (MLNN):

MLNN consists of a multilayer perceptron including an input layer, hidden layers, and an output layer as shown in Figure 15

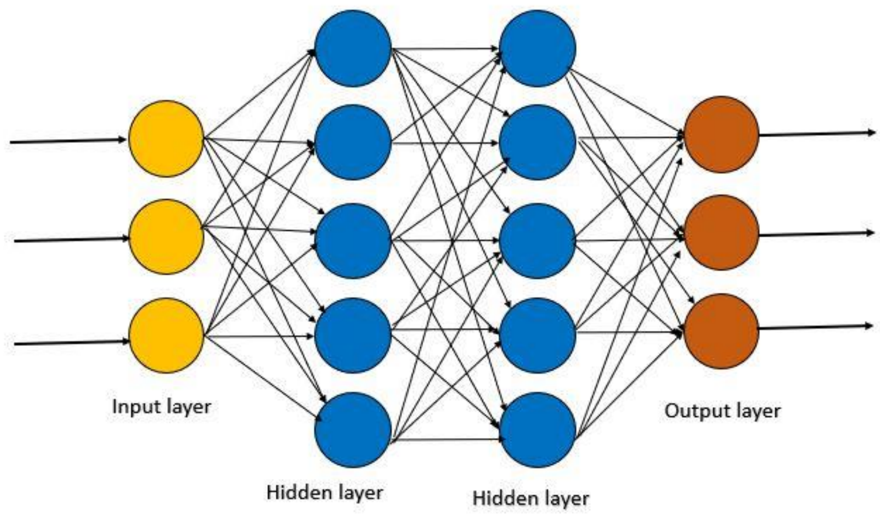


Figure 15: Multilayered -neural network model

The pixelated image data was fed into the MLNN feedforward layer. The backpropagation error minimization technique is used to compute the gradient of the loss function. The loss function in our case is Mean Squared Error (MSE),

$$L = \frac{1}{N} \sum_i (y_i - t_i)^2 \qquad (9)$$

where N is the number of data points, y_i is the predicted output from the MLNN and t_i is the targeted value of the classification. The goal is to minimize the MSE by adjusting the network's weight and biases. The general weight update is carried out by taking the gradient of the loss function with the weight and hence updating the weight using the following equation,

$$\Delta W_i = -\eta * \frac{\partial L}{\partial Wi} \qquad (10)$$

where, η is the hyperparameter controlling the step size of the update known as learning rate. The SoftMax function is utilized in the output layer of neural network models as the activation function to achieve feature classification. The output of the SoftMax function is the vector of probabilities of all possible outcomes. Mathematically, the SoftMax formula is expressed as follows:

$$S_i = \frac{\exp(y_i)}{\sum_{k=1}^{n} \exp(\mathrm{y_i})} \qquad (11)$$

2) Convolutional Neural Network:

Convolutional neural network (CNN) is a deep learning model that is well-known for image classification and pattern recognition. The main components of the CNN model are shown in Figure 16. CNN takes a pixelated oscillation image as an input signal and then goes to the convolution layer. A convolutional layer defines a filter that gets convoluted with each input matrix. The design of the convolutional filter for CNN is critical since it extracts the features from the input matrix. The result of the convoluted input passes into the pooling layer. The pooling layer performs the mean/median pooling and reduces the dimension of the feature. After several iterations of pooling and convolution, the result of the pooling layer is passed to the fully connected layer, which performs the classification similar to the MLNN as outlined above.

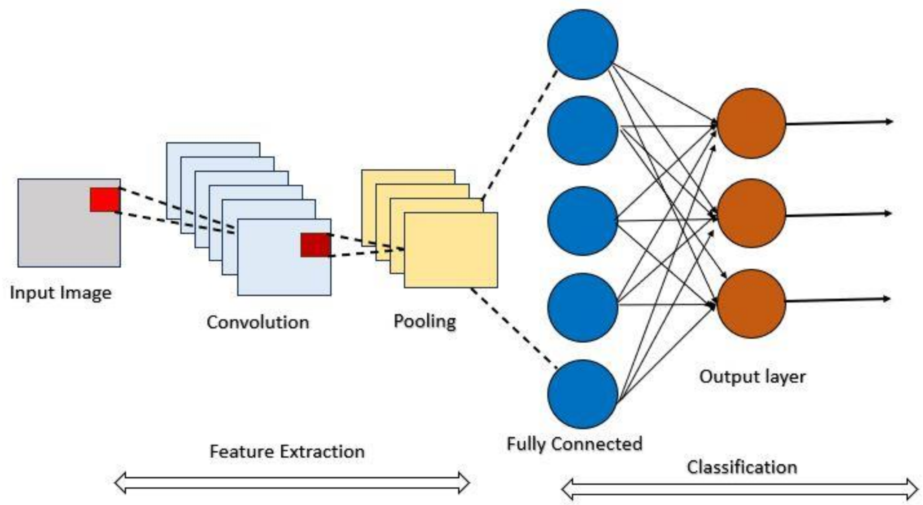


Figure 16: Convolutional neural network model

4.3 Transfer Learning (TL) Model Development

TL is a specialized area within ML that addresses the complexities associated with varying dataset distributions. Typically, when a new dataset (known as the target domain) is introduced for a particular task, relying solely on an ML model trained on an initial dataset (referred to as the source domain) may not be sufficient. This limitation is particularly pronounced when there are substantial differences in the data distributions between the source and target domains. These differences can lead to poor model performance if the model is directly applied to the new dataset without adjustments. One of the primary challenges in re-training an ML/AI model on a new dataset is the scarcity or complete absence of labeled data, which is crucial for supervised learning. Additionally, changes in feature extraction methods between the source and target domains can further complicate the transfer process. For instance, the features that were effective in the source domain may not be as informative or relevant in the target domain, necessitating modifications to the feature extraction process.

TL was selected at this stage because the small size of these datasets impedes the direct application of other methods, such as unsupervised learning or reinforcement learning. To elaborate, using reinforcement learning with these new datasets would require a new simulation to represent the environment (the power system) and create fault scenarios to drive the training process. This approach would negate the advantage of having multiple sets of historical PMU data, effectively reducing the problem back to a supervised learning scenario. Consequently, TL is the most suitable approach under these circumstances, as it allows for the effective adaptation of pre-trained models to new, unlabeled (or scarcely labeled) datasets without the need for extensive additional simulations or further data collection.

4.3.1 TL for Domain Adaptation for Line Faults

For addressing line faults, the chosen TL technique involved domain adaptation methods designed to align the feature spaces of both the source and target datasets. Domain adaptation focuses on minimizing the discrepancies between different domains, thus allowing the model to effectively apply the knowledge it gained from one dataset to another with different characteristics. Two TL algorithms were used in this study to achieve domain adaptation.

1) Correlation Alignment (CORAL)

CORAL is an unsupervised TL technique that relies on feature alignment. In other words, CORAL aligns the second order statistic (covariance) features of the data. Aligning the second order statistics of the source and target domains involves a linear transformation, A, that minimizes the distance between them using the Frobenius norm [50]. Using matrix transformations, the problem can be stated as expressed mathematically as shown in (12) below,

$$\min_{A} \| A^T C_S A - C_T \|_F^2 \qquad (12)$$

where C_S and C_T are the source and target domain covariance matrices, respectively. A detailed derivation for finding A can be found in [50]. Figure 17 illustrates the effect of applying CORAL to the features extracted from PSCAD+PNNL1 (source) and RTE1 (target), where the distance between the distributions decreases while maintaining the general shape. As observed on the right-side figure, the feature distributions seem to overlap more uniformly after applying CORAL.

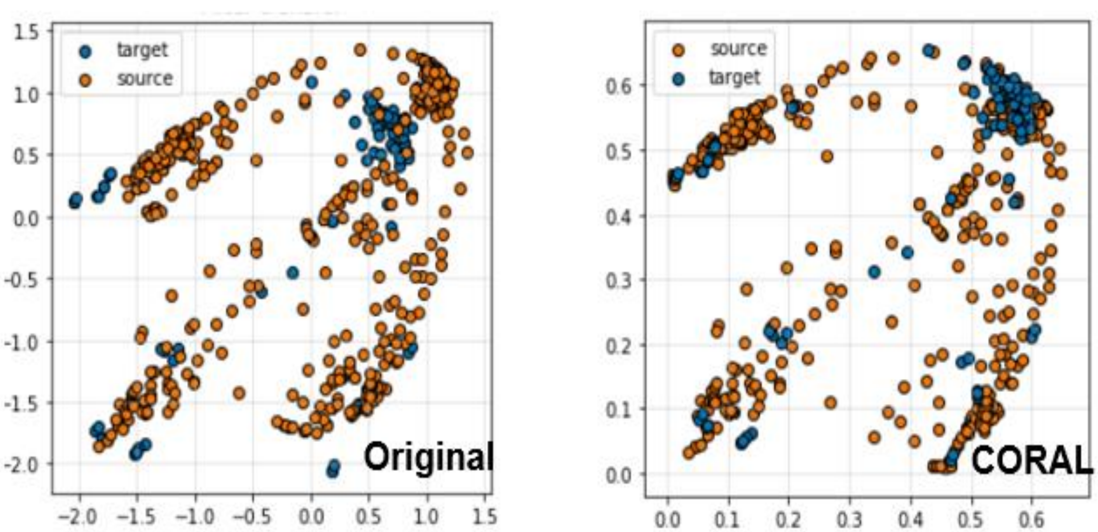


Figure 17: TL by CORAL on PNNL1 and RTE1 data

2) Transfer Component Analysis (TCA)

TCA is a feature mapping technique that learns a linear mapping that maps the features of the source and target domains onto a low-dimensional feature space. TCA is achieved by minimizing the distance between the marginal probability distributions of the source and target domains, $P(X_S)$

and $P(X_T)$. First, the algorithm uses a reduction method called maximum mean discrepancy embedding (MMDE) to embed the source and target domains on a low-dimensional latent space and learns its kernel matrix K:

$$K = \begin{bmatrix} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)} \tag{13}$$

where n_1 and n_2 are the sizes of the learning samples from the source and target domains. A transformation matrix W is then learnt that transforms the kernel matrix K that is guaranteed, as described by the derivations in [51], to be of a lower dimension than $(n_1 + n_2)$. Figure 18 illustrates the effect of applying TCA to the features extracted from PSCAD+PNNL1 (source) and RTE1 (target). TCA maps the features onto a common space, where the features are realigned. As observed on the right-side figure, the feature distributions are altered and centered in order to overlap but have lost their original shape.

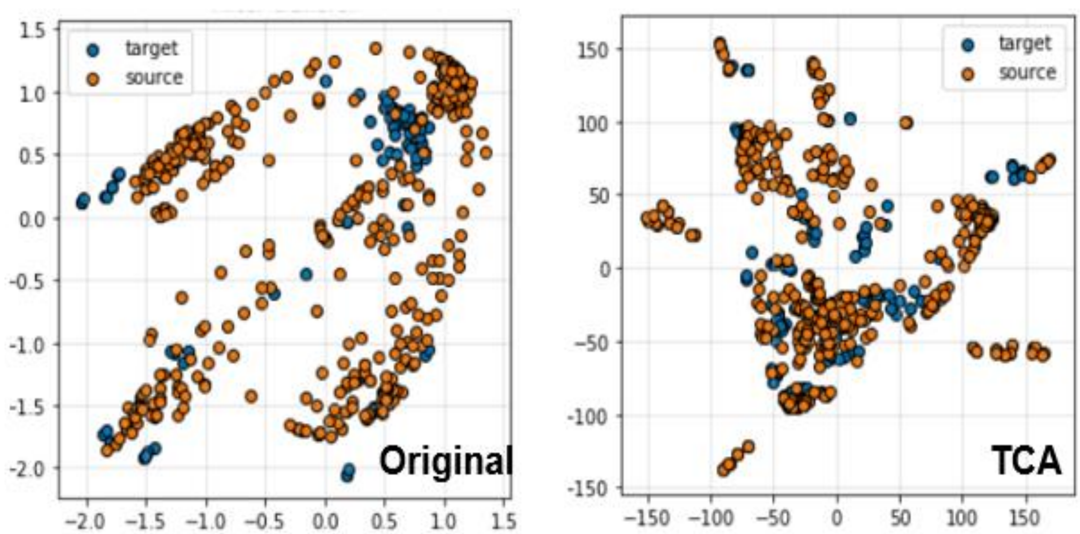


Figure 18: TL by TCA on PNNL1 and RTE1 data

4.3.2 TL via Fine-tuning (FT) for Frequency Events

FT is a specific form of TL that takes a pre-trained ML/AI model and continues the training process for a different, often smaller, target dataset [52]-[54]. FT falls under the parameter transfer branch of inductive transfer learning which assumes that although highly useful, model parameters obtained for the source dataset must be trained with a limited target data for better adaptability instead of directly using them [54]. The ML/AI model pre-trained on a source dataset usually has some optimal hyperparameters associated with it. At the same time, the target dataset often lacks sufficient data/labels in comparison to the source. FT uses a small number of known samples from the target dataset to fine-tune (by using a process such as grid search for this work) the attributes of the ML/AI model trained exhaustively on the source dataset to ensure that the resulting new ML/AI model has good performance on unseen samples of the target dataset. This technique helps in providing a warm start to train the new ML/AI model for the target dataset and eventually saves considerable time which would otherwise have been required if the training was done from scratch. In the context of the frequency event detection and classification problem, the source would be the simulated data created in PSLF, while the target would be the smaller field-recorded PMU datasets such as PNNL2, RTE1, and RTE2.

4.3.3 TL with Neural Networks (NNs) for Oscillation Events

TL for NNs has been explored in this study to leverage existing knowledge from the simulated oscillation datasets to enhance the performance of the NN methods discussed in Chapter 4.2.3. TL with CNN on medical image classification showed that the limited data and resources could be utilized to improve the performance of NNs [35]. TL with deep learning was also utilized to classify jamming signal power spectral density in the imaginary domain [55]. Similarly, CNN-based TL was implemented to classify the micro-seismic event waveform [56]. In [57], synchrophasor measurements were used to identify the power system event using a neural classifier with TL. In this work, offline training was performed on the NN models using simulated datasets that consist of different ranges of oscillations with a wide range of oscillation events with varying frequencies and damping ratios and their respective labels. The trained NN models were then fine-tuned and tested on field-recorded synchrophasor measurements from different regions of the US and RTE, France. The output layer of the trained model was adjusted to classify field-recorded event data based on damping characteristics. In addition, the moving window was continuously monitored, and an event was flagged when oscillation was detected in three consecutive sliding windows. The TL-based NNs capture complex patterns and features. This capability is leveraged after the network undergoes training with extensive sets of simulated datasets, enabling it to effectively generalize and apply the acquired knowledge to new, unseen field-measured datasets.

4.4 Transfer Learning (TL) Model Adaptability

In this phase of the study, the central question being investigated is whether each TL algorithm, which has been specifically tailored for its respective event type and associated features, can seamlessly adapt and maintain high performance when applied to the other two types of events. This exploration is crucial for determining the versatility and robustness of these TL algorithms across different scenarios. The primary goal is to ascertain whether vendors and potential users of such algorithms can employ any single TL algorithm as a universal solution capable of effectively handling all three types of events—line faults, oscillation events, and frequency events. To achieve this, each TL algorithm is rigorously tested across all available field-recorded datasets, encompassing a wide range of real-world conditions and variations. This comprehensive testing helps to identify if the adaptability and performance of the TL models are consistent across different datasets or if their effectiveness is dependent on specific data characteristics. Additionally, this step of the study sheds light on the potential limitations and strengths of each TL algorithm when faced with varying data distributions and event characteristics.

To answer this question, the following actions were taken:

i. Feature extraction was performed for line faults, frequency events, and oscillation events as described in Chapter 3.
ii. Line fault features are used to train and test the NN-based and FT-based TL models.
iii. Frequency event features were used to train and test the CORAL-based and NN-based TL models.
iv. Oscillation event features were used to train and test the CORAL-based and FT-based TL models.

4.5 Summary

This chapter presented the methodology to develop base ML/AI models to detect and classify the three event types with which this project is concerned. In several instances, open-source ML/AI algorithms have been employed after the critical steps of data preprocessing and feature extraction were completed. As a base classifier, the model to detect and classify line faults is SVM-based. For frequency events, several classifiers were tested: SVM, RF, XGBoost, *k*-nearest neighbors, and Gaussian Naïve Bayes. Due to the image-based nature of the features extracted for oscillation events, the base classifiers for those were using convolutional and multi-layer NNs.

To transfer the knowledge gained by these models to new datasets with different distributions of data, several TL algorithms were explored. For line faults, SVM was combined with both CORAL and TCA to evaluate which domain adaptation algorithm performed better. Fine-tuning was used to transfer the knowledge of frequency event classifiers, whereas an offline training method was used to transfer the knowledge of oscillation event classifiers. Finally, the adaptability of the developed TL models was tested by applying the developed models to the event types for which they were not originally developed. The results obtained are described in the next chapter.

5. Results and Discussion

5.1 Introduction

This chapter presents the outcomes of the experiments conducted to evaluate the performance of the ML/AI models developed in this study. The results are structured to highlight the effectiveness of the chosen algorithms, and the impact of various strategies employed, including supervised learning-based classifiers and transfer learning (TL) techniques, to detect and classify line faults, frequency events, and oscillation events. Each section of this chapter targets one event type, and within each section the results are presented for the supervised-learning models supported by simulated data first, followed by the results of applying TL methods to those base classifiers. The results of the developed models are also compared against two main open-source algorithms, SVM and RF, to demonstrate the utility of the new models. The results highlight the suitability of specific classifiers for particular event types, considering the unique features associated with each type.

The results of applying these TL algorithms to the new datasets and alternative event types are presented, focusing on their ability to maintain or improve the performance of the base classifiers. Detailed analyses are provided to illustrate how the differences in features and base classifiers influenced the outcomes of the TL algorithms. This leads to the results of the study on TL model adaptability in which the TL models developed for each task are tested for the task of detecting and classifying the other two event types. This section highlights the strengths and limitations of the models in real-world applications. The results for fault, frequency and oscillation events are presented in sections 5.2, 5.3, and 5.4 respectively.

5.2 Results of Detection and Classification of Line Faults

The approach took on two main stages to create and test ML/AI models for detecting and classifying line faults as outlined in Chapter 4 of this report: ML/AI model development and TL model development. At first several ML/AI models were evaluated to understand the effects of the use of different training and testing datasets on the performance of these ML/AI models. Next, the problem of varying data distributions that have emerged with each new dataset was tackled through the use of TL.

5.2.1 ML/AI Model Development Results

The first question to be answered by the new ML/AI models developed using the PSCAD simulated data as a training dataset was the effect of using this data specifically versus using a field-recorded dataset for training. Three models were created in which the training datasets were PSCAD data, PNNL1 data, and an integration of PSCAD and PNNL1 data. The testing dataset was RTE1 for all three cases and the ML/AI algorithm was set to be SVM. The results in each case were evaluated using the event log provided by RTE for all the events that were correctly detected. Misclassified instances were evaluated using visual inspection. In this stage of the project, only event instances were used to train and test these ML/AI models. The inclusion of normal operation time windows occurs in the TL stage. Including normal operation adds another layer of complexity to the problem where random disturbances and noise may be misclassified as an event, which explains the difference in F1-Score in Table 7 compared to Table 8.

The results of this experiment are presented in Table 7. This study revealed the importance of the availability of field-recorded data to train the classifier models. Although the simulated data provided a more balanced training dataset, it did not reflect the noise and random disturbances that exist in field-recorded PMU data. In this case, the data provided by PNNL (PNNL1) was large enough to offer several instances of each line fault type. Nevertheless, obtaining such large amounts of PMU data (spanning 2 years) proved to be more challenging with each new dataset received. This amount of data might not be regularly stored and collecting it requires further time and effort. Under the circumstances, it cannot be concluded that field-recorded data is always the best source for training data. The scenario that provides both a realistic representation and a more balanced dataset is the use of an integrated dataset of both simulated and field-recorded data.

Table 7: Training dataset study for line fault detection and classification

Training Dataset	**Testing Dataset**	**Weighted F1-Score**
PSCAD	RTE1	0.91
PNNL1	RTE1	0.95
PSCAD+PNNL1	RTE1	0.95

The best performing SVM model, using PSCAD+PNNL1 data for training, showed promising success when tested on RTE1 data. This success was highlighted by the detection and classification of five-line faults that were not included in the original event log provided by RTE for this data. One of these five events is shown in Figure 19. One possible reason that this event, among others, was detected by the SVM model is the fact that the event was only observed well by one PMU. This result highlights the benefits of feature engineering and understanding the nature of each event type.

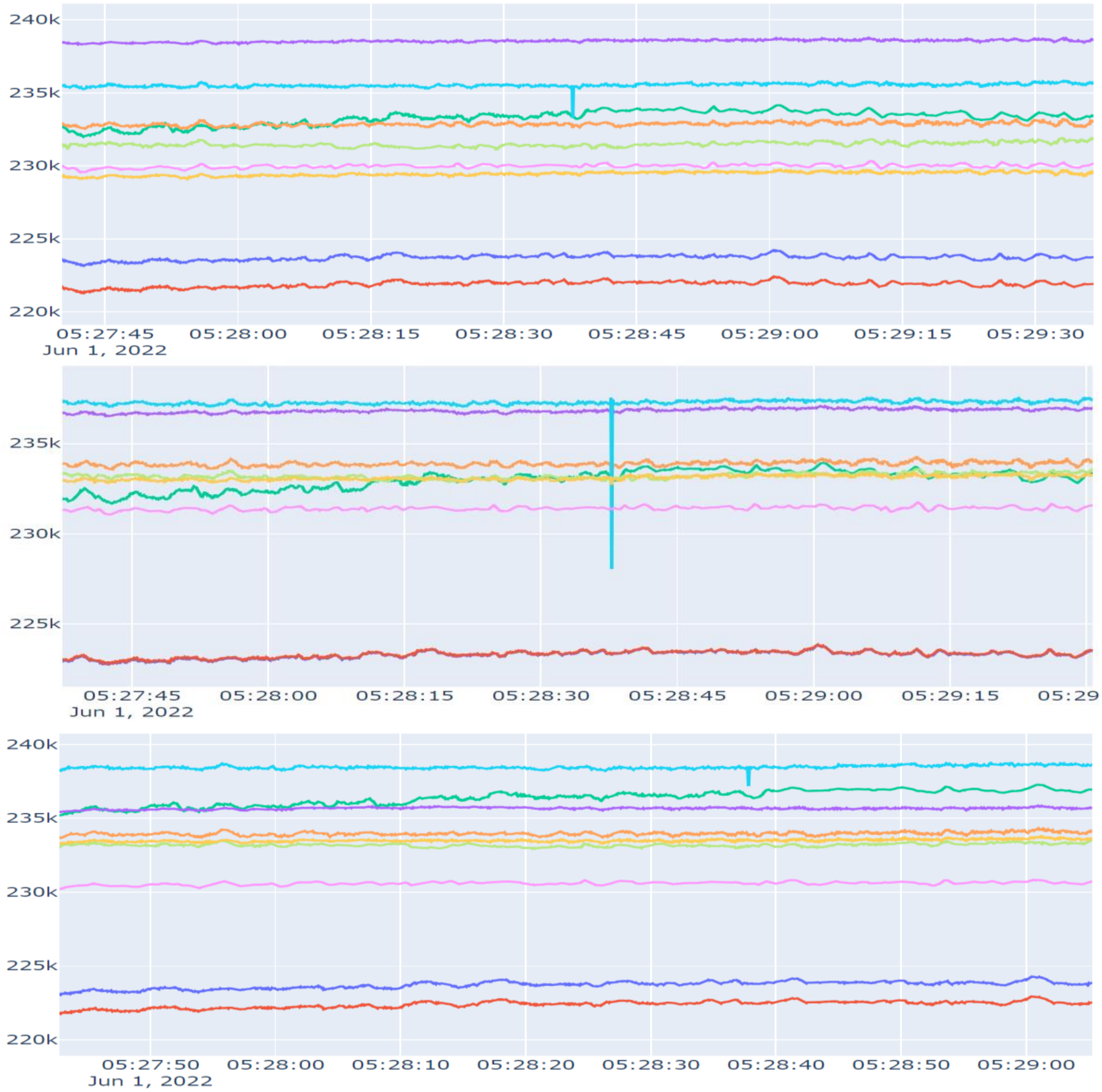


Figure 19: top to bottom: phases a, b, and c of a sample unlabeled event from RTE1

5.2.2 Transfer Learning (TL) Model Development Results

As mentioned in Chapter 4, the TL technique chosen for line faults was the use of domain adaptation methods that realign the feature spaces of the source and target datasets. These TL tools ensure that the work done to engineer a physics-based feature specific to line faults is preserved but those feature distributions are brought closer together to facilitate the transfer of knowledge gained by the model from one dataset to another. The two domain adaptation techniques explored were CORAL and TCA. CORAL and TCA were applied to transfer the knowledge gained by the

SVM model trained on the PSCAD+PNNL1 dataset to the RTE1 and RTE2 datasets. PSCAD+PNNL1 was used as the source dataset since it is the most comprehensive and diverse dataset, whereas RTE1 and RTE2 are both limited in size and class diversity. Table 8 summarizes the results of this experiment.

Table 8 also shows the comparison between applying TL to the SVM model and simply using the SVM model as is, to test on the RTE datasets. As a benchmark, SVM and RF were chosen as the base classifiers against which the results of TL are assessed. The F1 scores highlighted for SVM and RF in rows 1 and 3 show that these two base classifiers perform best when the training and testing datasets originate from the same source. In the following rows, a considerable decline in performance is visible due to the difference in distributions between the training and the testing datasets. Using TL, both CORAL and TCA, achieve better results when combined with SVM. However, CORAL exhibits the highest F1 score for both RTE1 and RTE2. A possible explanation for CORAL's superiority might be the fact that TCA involves a transformation to lower dimensional space to realign the features while CORAL does not.

Table 8: TL model development results for line fault detection and classification

Algorithm	**Training Data**	**Testing Data**	**F1-Score**
SVM	PSCAD + PNNL1	PSCAD + PNNL1	**0.98**
	PSCAD + PNNL1	RTE1	0.77
	PSCAD + PNNL1	RTE2	0.35
RF	PSCAD + PNNL1	PSCAD + PNNL1	**0.98**
	PSCAD + PNNL1	RTE1	0.64
	PSCAD + PNNL1	RTE2	0.21
SVM with CORAL	PSCAD + PNNL1	RTE1	**0.93**
	PSCAD + PNNL1	RTE2	**0.90**
SVM with TCA	PSCAD + PNNL1	RTE1	0.78
	PSCAD + PNNL1	RTE2	0.86

5.3 Results of Frequency Event Detection

5.3.1 Frequency Events Identification Using the Empirical Approach

Using the empirical approach described in Chapter 2.3.1, 89 unique frequency events were identified in the PNNL2 dataset. The maximum number of PMUs that captured a unique frequency event in this dataset was 19. A total of 5 unique frequency events were identified in the RTE1 dataset and 12 unique frequency events were found in the RTE2 dataset.

The unique Frequency Event #39 from the PNNL2 dataset is analyzed next. Table 9 shows the details of the 19 PMUs that respond to this event. Specifically, maximum and minimum frequency, maximum RoCoF, and their instance of occurrence are specified. The positive sequence voltage magnitude, positive sequence current magnitude, and frequency plots for Unique Event #39 are shown in Figure 20. Figure 21 and Figure 22 depict the disturbance recorded in frequency information captured by two PMUs for Unique Event #39 with the highest and lowest values of RoCoF response, respectively.

Table 9: PMU responses to unique Frequency Event # 39 in the PNNL2 dataset and its details

PMU ID	Date	Hour	Max F Index	Max F (in Hz)	Min F Index	Min F (in Hz)	Max |RoCoF| Index	Max |RoCoF| (in Hz/s)
C708	20161008	10	26606	60.129	26605	59.913	26606	7.38
C827	20161008	10	26607	60.059	26605	59.936	26605	4.37
C151	20161008	10	26607	60.071	26605	59.926	26605	4.31
C730	20161008	10	26607	60.055	26605	59.942	26605	4.3
C175	20161008	10	26607	60.06	26605	59.937	26605	4.27
C110	20161008	10	26607	60.059	26605	59.938	26605	4.25
C211	20161008	10	26607	60.055	26605	59.943	26605	4.24
C277	20161008	10	26607	60.059	26605	59.938	26605	4.24
C313	20161008	10	26607	60.055	26605	59.943	26605	4.22
C722	20161008	10	26607	60.058	26605	59.943	26605	4.21
C837	20161008	10	26607	60.07	26605	59.929	26605	4.19
C526	20161008	10	26607	60.056	26605	59.946	26605	3.95
C865	20161008	10	26607	60.051	26605	59.953	26605	3.88
C143	20161008	10	26607	60.052	26605	59.955	26605	3.6
C397	20161008	10	26607	60.053	26605	59.954	26605	3.59
C396	20161008	10	26607	60.046	26605	59.964	26605	3.22
C250	20161008	10	26607	60.046	26605	59.964	26605	3.18
C682	20161008	10	26607	60.046	26605	59.964	26605	3.17
C893	20161008	10	26607	60.045	26605	59.969	26605	2.78

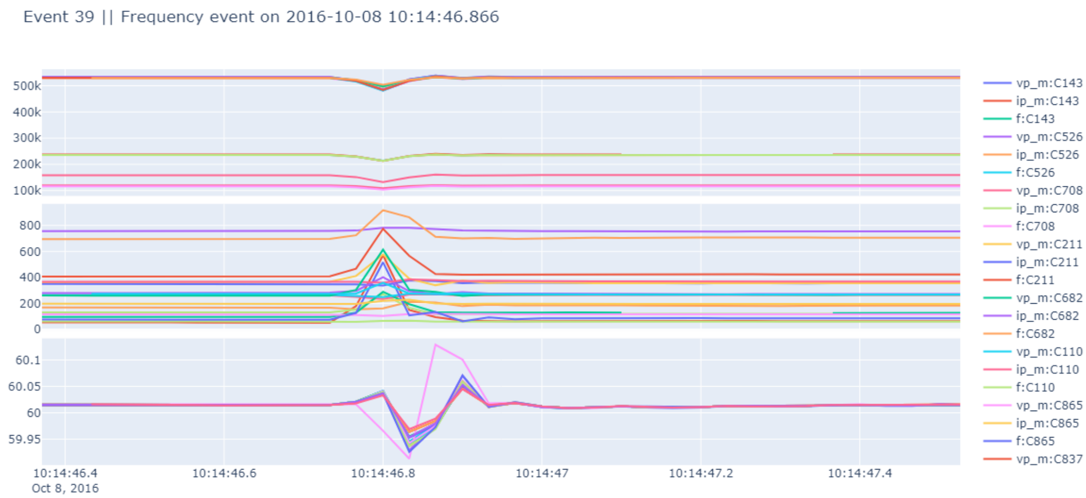


Figure 20: Voltage, Current, and Frequency recorded by PMUs for Unique Frequency Event #39

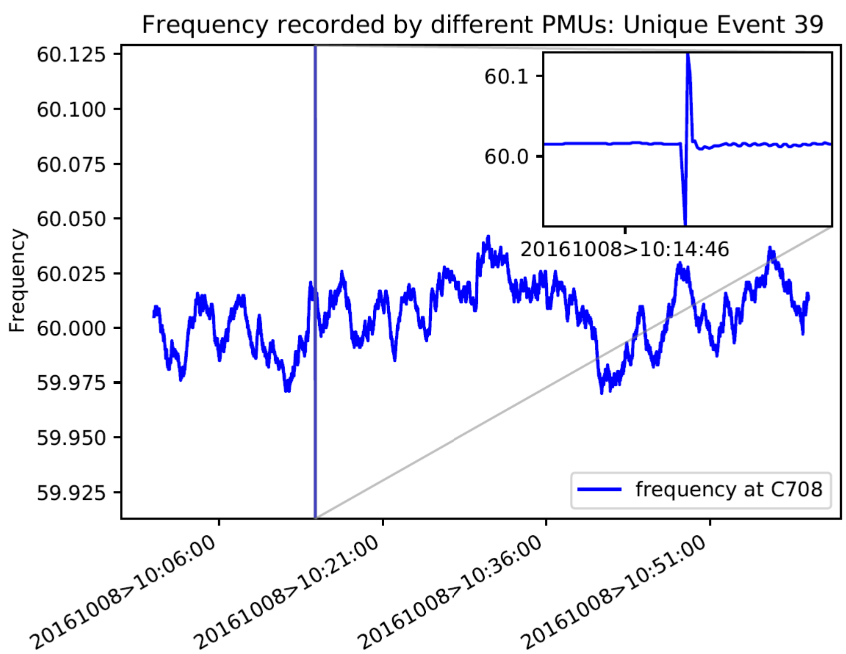


Figure 21: Frequency recorded by PMU C708 (highest RoCoF) for Unique Frequency Event #39

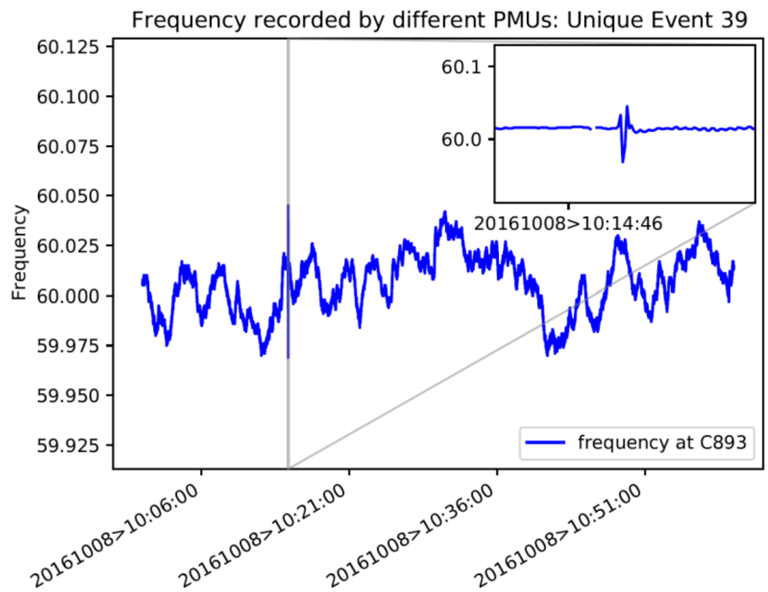


Figure 22: Frequency recorded by PMU C893 (lowest RoCoF) for Unique Frequency Event #39

The duration of frequency events is estimated next by computing the initiation time of frequency events and their end, by using the successive difference of the average value of frequency technique as described in Chapter 2.3.2. Figure 23, for instance, shows the approximate start and end times of the unique frequency event #44 in the PNNL2 dataset.

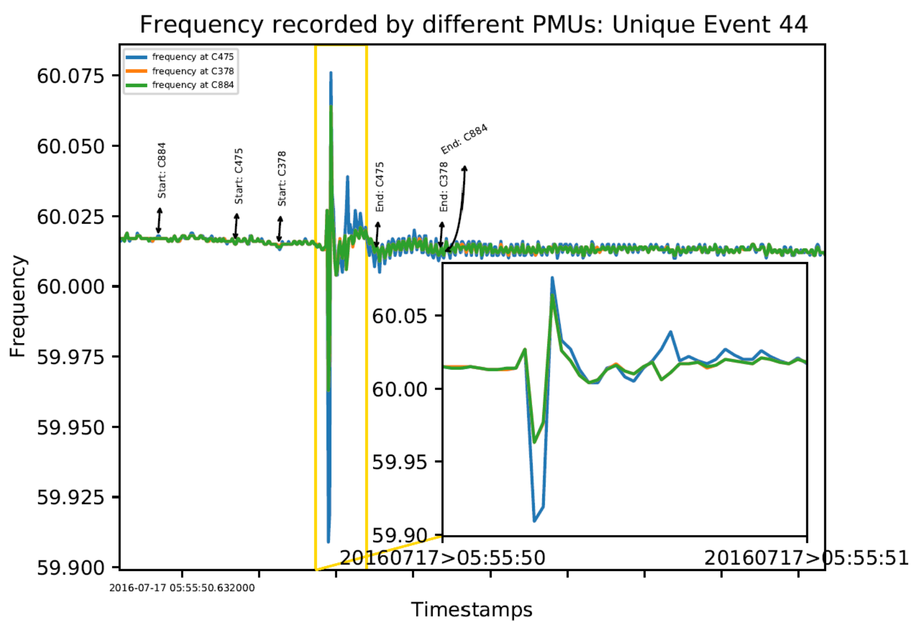


Figure 23: Frequency behavior showing start and end by different PMUs for Unique Frequency Event #44

5.3.2 ML/AI Classifier Results

Modal features previously extracted using the Prony method for frequency events from the simulated PSLF data were used to train ML/AI classifiers described in Chapter 4.2.2. The number of PMUs used for creating F_{param} was determined by analyzing performance of the base classifiers as the number of neighboring PMUs was varied; the best performance was seen with 10 PMUs. If 10 PMUs are not present for a particular contingency, appropriate number of zeros were padded to F_{param}. Note that the features were extracted from the post disturbance time-series data for the four signals of interest (V_{mag}, V_{ang}, Frequency, RoCoF).

Table 10 shows the performance in terms of accuracy, precision, recall, and F1-score which are standard metrics to assess a classifier's performance. It was seen from the table that the XGBoost classifier had the best performance among the five classifiers that were evaluated.

Table 10: Performance of ML Classifiers on the PSLF dataset containing 21362 post-processed frequency events

Classifier Type	**Accuracy**	**Precision**	**Recall**	**F1-Score**
XGBoost	0.968	0.968	0.968	0.967
Random Forest	0.903	0.909	0.903	0.889
Support Vector Machine	0.892	0.887	0.892	0.886
K-Nearest Neighbors	0.836	0.838	0.836	0.835
Gaussian Naïve Bayes	0.417	0.607	0.417	0.378

The confusion matrix for the XGBoost classifier for the frequency events in the simulated PSLF dataset is shown Figure 24. Labels 1, 2, 3, and 4 correspond to events representing a normal scenario, generator loss, load loss, and line fault, respectively.

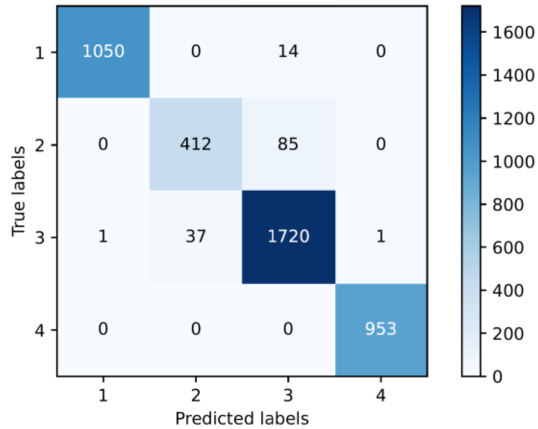


Figure 24: Confusion Matrix using XGBoost Classifier: for Simulated dataset from PSLF

The three best base classifiers SVM, RF, and XGBoost in the training stage were used with the tuned hyperparameters directly and tested on the events in the PNNL2 and the combined RTE datasets. The performance is shown in Table 11. However, all three classifiers performed poorly when tested on PNNL2 and combined RTE datasets.

Table 11: Testing on unseen events in the PNNL2 and RTE datasets

Classifier	Event Type	Training Data	Testing Data	F1-Score
SVM	Frequency Events	PSLF	PSLF	0.88
		PSLF	PNNL2	0.52
		PSLF	RTE1+ RTE2	0.29
RF	Frequency Events	PSLF	PSLF	0.89
		PSLF	PNNL2	0.51
		PSLF	RTE1+ RTE2	0.50
XGBoost	Frequency Events	PSLF	PSLF	0.97
		PSLF	PNNL2	0.09
		PSLF	RTE1+ RTE2	0.47

From the results obtained in Table 11, it can be inferred that even the best performing ML models on one dataset are unable to achieve a satisfactory performance on similar events in a different dataset. This motivated us to use transfer learning via fine-tuning as explained in Section 4.3.2 and the results are presented for the same in the following section.

5.3.3 Fine-Tuning (FT) Results for Frequency Events

Since XGBoost outperformed SVM and RF classifiers on the simulated PSLF data for frequency events (see Table 11), this classifier was fine-tuned to predict labels for the PNNL2 and combined RTE datasets. Labels for these datasets were found through the empirical approach explained in Section 2.3.2. The results obtained are shown in Table 12 and Figure 25. The confusion matrices are shown in Figure 25 and a comparison of Table 12 with the results obtained in Table 11 indicate the superiority of FT in successfully classifying the frequency events in both the PNNL2 and combined RTE datasets. The results also reaffirm the usefulness of the PSLF simulated data (created from a large 60 Hz WECC system) in aiding the data driven and fine-tuned XGBoost model to make it capable of distinguishing field recorded events in real systems with different system frequency (50 Hz European system).

Table 12: Results using Fine Tuning for frequency events

TL Algorithm	Event Type	Training Data	Testing Data	F1-Score
XGBoost with Fine-tuning	Frequency Events	PSLF + PNNL2	PNNL2	1.0
		PSLF + RTE1 + RTE2	RTE1 + RTE2	1.0

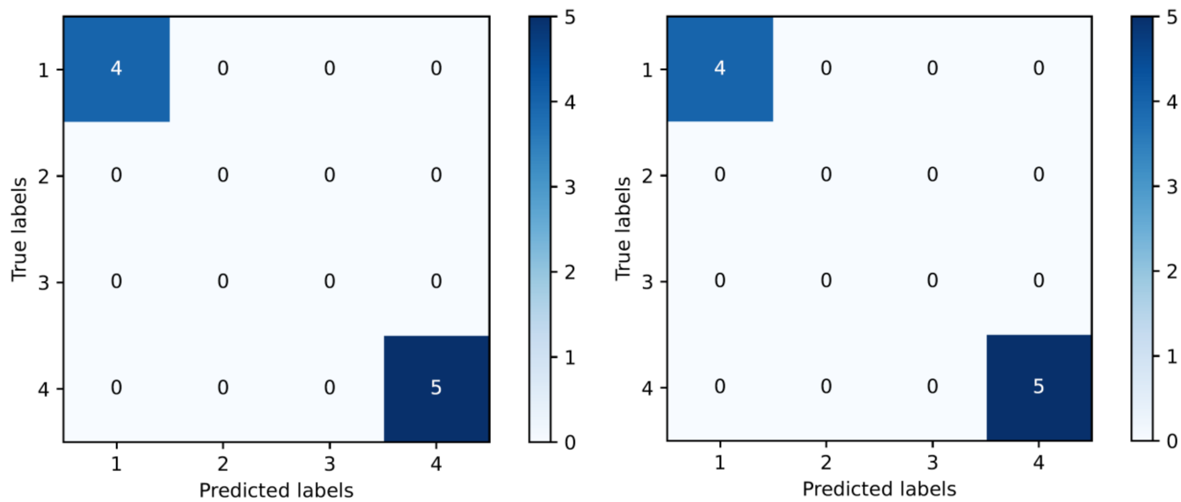


Figure 25: Confusion matrices using fine-tuned XGBoost on the field recorded datasets PNNL2 (left) and RTE1+RTE2 (right)

The underlying workflow and results for frequency events described in this study can be represented pictorially by Figure 26 shown below.

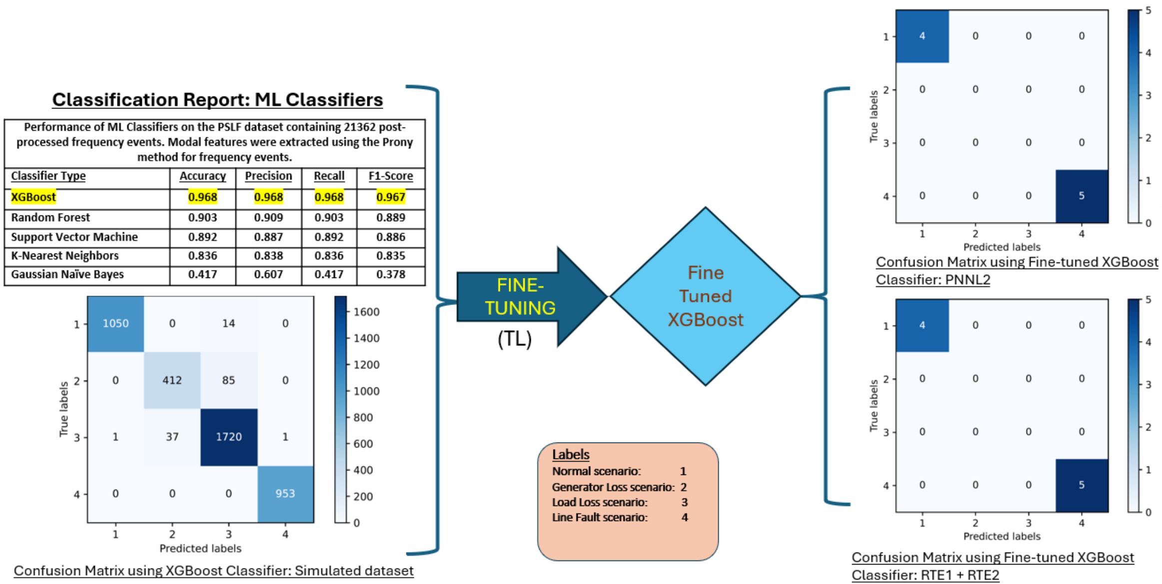


Figure 26: Model Development Workflow and Results for Frequency Events

5.4 Results of Oscillation Events Classification

5.4.1 Neural Network (NN) Results

In this work, the NN approach for oscillation classification was implemented in two stages. In the first stage of the work, NN models were solely trained with the datasets consisting of a single mode. In the process of training, 80% of the simulated datasets were used as the training sets, and 20% datasets were used for the testing. Figure 27 shows the confusion matrix for the MLNN when trained with the single-mode oscillation datasets.

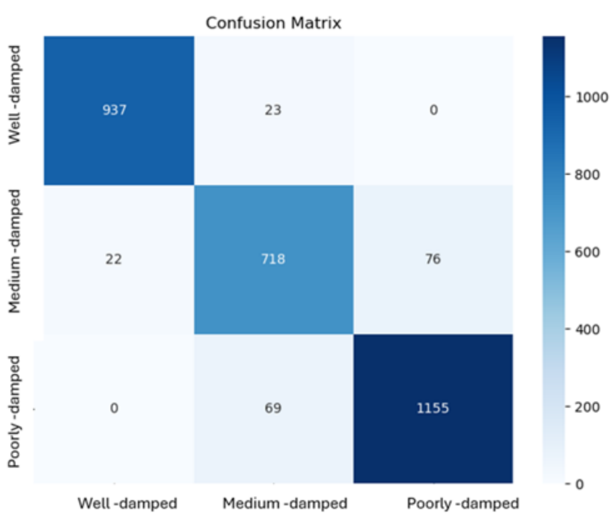


Figure 27: Confusion Matrix for MLNN

A trained MLNN model was used to test the field-recorded measurements from the US and RTE. Table 13 shows the performance of MLNN where WD stands for well-damped, MD stands for medium damping, PD stands for poorly damped, M stands for misclassified, and C stands for confidence. The confidence of the classification shows that the trained model was able to classify 7 out of 10 signals as well-damped in the RTE event case. Similarly, for the WECC ringdown events, confidence in the testing from the trained MLNN model was found to be greater than 90%

Table 13: Oscillation classification using MLNN

Labels	**WD**	**MD**	**PD**	**M**	**C**
Number of signals (RTE)	7	0	0	3	70%
Number of signals (US), first ringdown event	106	0	0	10	90%
Number of signals (US), second ringdown event	109	0	0	13	92%

The confidence of the field-recorded measurements from the trained model was validated using a well-established physics-based oscillation modal analysis method [44]. In our testing, we used the Hankel total least squares (HTLS) to find the modal parameters of the ringdown events recorded in the RTE and the US datasets. The results obtained are shown in the tables below.

From Table 14 and Table 15, we noticed that the ringdown events in the RTE and the US datasets were well-damped which validated the confidence shown by the trained models aligned with the physics-based approach.

Table 14: Modal results for the RTE data

Mode Frequency (Hz)	0.76	0.25
Mode Damping Ratio (%)	17.23	3.83
Mode Relative Energy (%)	90.61	9.39

Table 15: Modal results for the US data

Mode Frequency (Hz)	0.41	0.69	0.82
Mode Damping Ratio (%)	9.72	8.78	9.29
Mode Relative Energy (%)	66.47	16.51	12.59

5.4.2 TL Model Development Results

In the second stage of the work, the improved training datasets consisting of multiple modes for ML training were utilized. Figure 28 and Figure 29 shows the performance of the MLNN and CNN using multiple modes oscillation training datasets. From the confusion matrices, it is observed that the performance of the multiple modes training has significantly improved in comparison to the single-mode training.

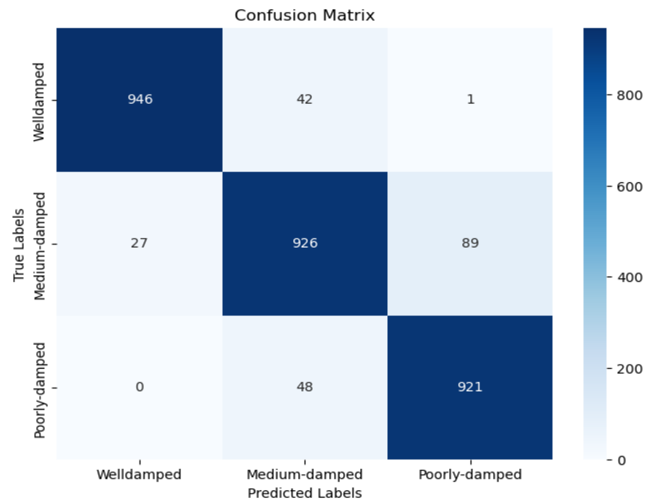


Figure 28: Confusion Matrix for MLNN using multiple modes training datasets

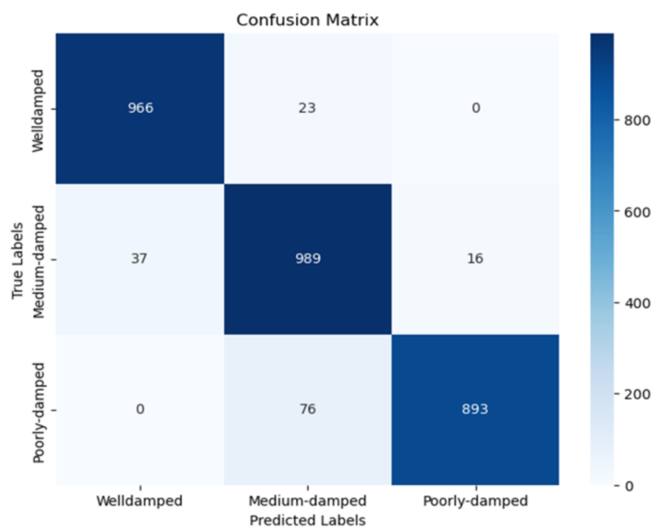


Figure 29: Confusion Matrix for CNN using multiple modes training datasets

Table 16 shows the performance of the multiple modes trained NN models in terms of F1-score. The F1 score achieved by TL for CNN surpassed 0.95, while TL for MLNN's F1 score was below 0.9. This observation highlights the superior performance of TL for CNN in the field of oscillation image classification.

Table 16: F1-score of MLNN and CNN on different testing datasets

Model	ML Algorithm	Training Data	Testing Data	Training F1-Score	Testing F1-score
1	MLNN	Simulated Multiple modes system	RTE-1	92.96	88.89
2	MLNN		WECC brake test at 15 mins		84.14
3	MLNN		WECC brake test at 25 mins		86.84
4	CNN		RTE-1	95.9	100
5	CNN		WECC brake test at 15 mins		96.71
6	CNN		WECC brake test at 25 mins		98.67

5.5 Results for TL Model Adaptability

In this subsection, testing was conducted to evaluate the features extracted from one event by applying them to another ML model, which had shown good performance with different event features, to understand the feature transfer capability between the ML models. Figure 30 shows the performance of the CORAL TL when tested on features relevant for frequency events and oscillation events. Similarly, Figure 31 shows the result of line faults and frequency events features when tested on the TL-MLNN. A similar cross-comparison as Figure 30 and Figure 31 was infeasible with the frequency event's model for line fault and oscillation event features as there were not enough events of all classes in the test cases to apply FT of the XGBoost algorithm after training on the respective base datasets. It was observed that the features designed for individual TL models were unique and could not be transferred to other models. This is due to the fact that the training process of each ML model is specific to that event, and the nature of events and duration of each event is different.

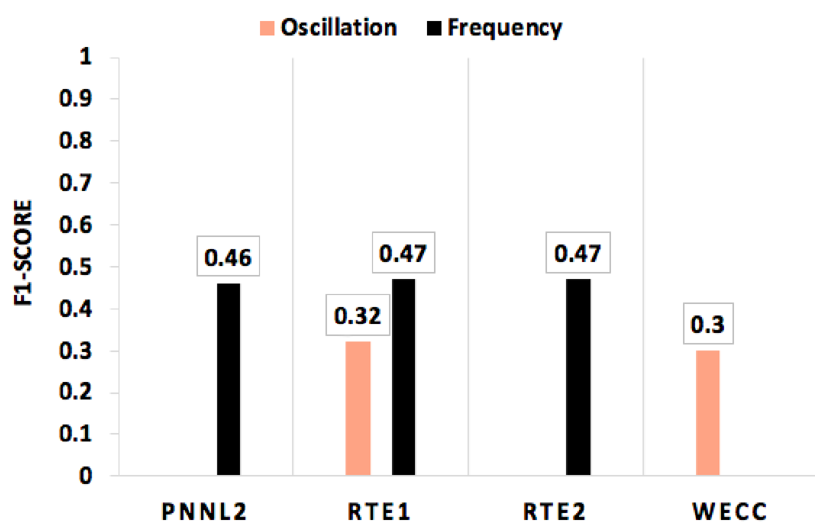


Figure 30: TL using CORAL results for oscillation and frequency events

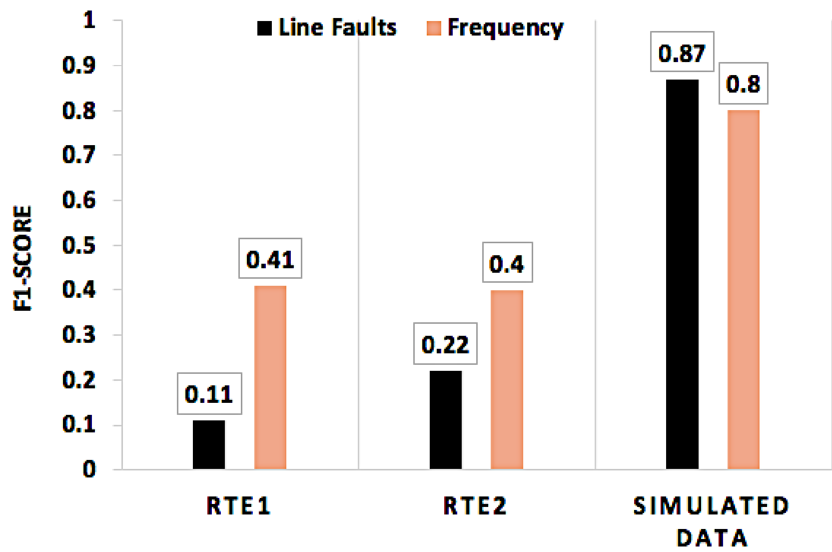


Figure 31: TL using NN results for line faults and frequency

5.6 Summary

ML/AI models were trained for different types of events with their respective features. The performances of the ML/AI models were measured in F1-score and confusion matrices. Simulated datasets along with field measurements were utilized for training the ML/AI models. Field-recorded events from the US and the RTE datasets were tested on the trained models. Offline training with a wide range of events and TL capability of ML/AI models shows great potential of using trained models for real-time monitoring of events in the system. The key takeaway from the results section is that the features are unique for each type of event and cannot be transferred from one event to another.

6. Conclusions

This section offers insights into the unique experiences that the project team has gained through the course of this project. The conclusions reflect on the major study findings, challenges with data quality, and finally make some recommendations for the best industry practices on handling the synchrophasor data so that it can be more useful for subsequent ML/AI applications.

6.1 Major Study Findings

The major findings of this project are the following:

- The process of developing ML/AI models that precedes training and testing different algorithms is the most crucial. It includes data preprocessing, feature engineering, and the choice of training and testing algorithms. A step that proved to have the most influence is the preparation of labels. Label accuracy in terms of class type and timestamp may be the factor that determines whether all the ML/AI models considered will perform as expected. Ensuring that the labels are of high quality is achieved either through the visual inspection of field-recorded labels or through the manual creation of labels that accompanies the simulations.

- Developing models for the detection and classification of line fault events, frequency events, and oscillation events benefit most from the use of an integrated dataset containing both simulated and field-recorded PMU data for training. Such training datasets ensure that there is a balance in the number of events of each class which can be achieved through the simulation flexibility and that the noise and disturbances that exist in field-recorded PMU data are represented in the overall training process.

- Transfer learning (TL) using CORAL, which is a well-developed TL algorithm, with an SVM-based classifier was shown to be the most successful method to transfer the knowledge learned by an SVM model for line fault detection from a large, diverse dataset to a smaller, less-diverse dataset. The availability of limited line fault event labels made this TL approach unique for line faults as opposed to the use of neural networks, for example, which require larger amounts of labeled events.

- Modal-based (Prony) features with a fine-tuned XGBoost classifier successfully demonstrated the usefulness of temporal dynamics as features to detect and classify frequency events in field-recorded data.

- Oscillation detection and classification using the image-based ML/AI approach showed promising results on simulated datasets and field-recorded measurements. The results show that the training process can be carried out offline with a wide range of simulated oscillation events with different damping ratios and frequencies, and the trained model can be utilized to test PMU measurements from different regions independent of system topology and number of PMUs in the system.

- This study mainly emphasizes classifying oscillation types based on their damping status rather than estimating the frequency of oscillation modes. However, this approach could potentially be extended in the future to classify the nature of oscillations, such as distinguishing between slow electromechanical oscillations and fast sub-synchronous oscillations.

- The extracted features for each application type worked best with the algorithms they were originally engineered for, which proves that the feature engineering process needs to be tuned to specific applications of interest.

6.2 Reflecting on Data Quality Issues

Additionally, data quality issues were prevalent across all sources of PMU data. With the aim that our experiences of highlighting them will enable better practices for synchrophasor data measurement and storage, some of the issues that were ubiquitous and presented challenges including but not limited to feature design for ML/AI models are listed below:

- Missing blocks of data: There were missing blocks of data in the PNNL dataset. The duration of missing ranged from a few milliseconds to nearly the entirety of an hour (in the hourly files available). In those cases, the number of frames per hour was less than expected for a typical hour (108000 frames in total for a 30 frames/s reporting rate).
- Stale data: Despite having a good status word (which represents data quality at a timestamp level), the data did not refresh after a certain measurement for a period. This was prevalent across both PNNL and RTE datasets.
- Channels with all zero values: This problem was similar to the case with stale data but with an additional observation that all data points took zero values. This was seen mainly in the RTE dataset.
- Imprecise timestamps: Event logging practices were found to be inconsistent between different data sources. Some events were only timestamped up to the minute in which the event occurred. In other instances, the timestamps identifying the event start times were incorrect, often a couple of seconds before or after the actual event.
- Inaccuracy/lack of labels: One of the most significant challenges encountered with most new datasets was the lack of event labels. Additionally, when labels were present, some labels were identified to be incorrect after visual inspection.

6.3 Recommendations

It is important that the root causes for the aforementioned data quality issues are found and an attempt must be made to eliminate/mitigate them so that future synchrophasor based ML/AI applications can achieve an improved event detection and classification compared to what is possible today. The following are some recommendations inferred from this team’s experience:

- Data storage and management must become a priority for field-recorded PMU data. The development of reliable ML/AI models requires large amounts of "useful" data containing several instances of each event that is of interest. Therefore, databases must be constantly updated and maintained to be used to train for new ML/AI tools that are being deployed by vendors.
- A standardized event labeling method must be implemented at every level of the power system such that the need to modify and relabel event logs is eliminated from the data preprocessing stage. A systematic list of event label syntax must be adopted so as to avoid having two labels describing the same event, for example, "3P" and "ABC".
- Timestamps that accompany event labels must be checked for accuracy before entering these labels into the event log. When these checks are performed regularly, the quality of the training process for ML/AI models can be significantly enhanced, and less time can be spent on improving these labels.

  Regular data quality checks must be performed to mitigate any data quality issues that arise during the regular operation of PMUs or after major events in the system. These data quality checks must focus on missing data, stale data, unreasonable values, missing/incorrect event labels, and imprecise timestamps.

6.4 Expected Outcomes

Following the recommendations in Section 6.3 is expected to significantly enhance the effectiveness of synchrophasor-based ML/AI applications in detecting and classifying power system events. Prioritizing data storage and management would ensure the availability of a rich dataset containing multiple instances of each event, which leads to the development of more reliable ML/AI models. Implementing a standardized labeling system would reduce the time and effort required to prepare data for training ML/AI models usually spent modifying inaccurate labels. Additionally, conducting regular data quality checks can mitigate issues such as missing data, stale data, unreasonable values, incorrect timestamps, or missing event labels. These are often the obstacles faced during the data preprocessing stage.

The use of simulations might still be needed in order to bring balance to data by creating enough instances of rarely occurring events. High-quality, well-labeled, and abundant datasets would enable the direct training of ML/AI models, potentially minimizing the need for TL to adapt models to new datasets. However, following the recommendations is not expected to change the result of the model adaptability study, as the TL models developed in this project were task-specific, performing best when applied to the event types they were created for.

References

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

[2] North American SynchroPhasor Initiative, “Data mining techniques and tools for synchrophasor data,” *NASPI White Paper*, NASPI-2018-TT007, January 2019.

[3] T. Dokic, R. Baembitov, A. A. Hai, Z. Cheng, Y. Hu, M. Kezunovic and Z. Obradovic., “Machine Learning Using a Simple Feature for Detecting Multiple Types of Events from PMU Data,” in 2022 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA), 24-26 May 2022 2022, pp. 1-6.

[4] R. Baembitov, T. Dokic, M. Kezunovic and Z. Obradovic, “Fast Extraction and Characterization of Fundamental Frequency Events from a Large PMU Dataset Using Big Data Analytics,” HICSS-54 Conference, Hawaii, USA, January 2021, DOI: 10.24251/HICSS.2021.389.

[5] T. Mohamed, M. Kezunovic, Z. Obradovic, Y. Hu and Z. Cheng, “Application of Machine Learning to Oscillation Detection using PMU Data based on Prony Analysis,” 2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Novi Sad, Serbia, 2022, pp. 1-5, doi: 10.1109/ISGT-Europe54678.2022.9960589.

[6] H. Otudi, T. Dokic, T. Mohamed, M. Kezunovic, Y. Hu, Z. Obradovic, “Line faults classification using machine learning on three phase voltages extracted from large dataset of PMU measurements,” *Hawaii Int. Conf. on Syst. Sciences (HICSS-55*), Hawaii, USA, Jan. 2022.

[7] H. Otudi, T. Mohamed, M. Kezunovic, Y. Hu, Z. Obradovic, “Training machine learning models with simulated data for improved line fault events classification from 3-phase PMU field recordings,” *Hawaii Int. Conf. on Syst. Sciences (HICSS-56)*, Hawaii, USA, Jan. 2023.

[8] M. Kezunovic, Z. Obradovic, Y. Hu, “Automated system-wide event detection and classification using machine learning on synchrophasor data,” *CIGRE General Session*, Paper # 224, Paris, France, August 2022.

[9] D. Nguyen, R. Barella, S. A. Wallace, X. Zhao and X. Liang, "Smart grid line event classification using supervised learning over PMU data streams," *2015 Sixth International Green and Sustainable Computing Conference (IGSC)*, Las Vegas, NV, USA, 2015, pp. 1-8.

[10] M. Z. Che Wanik, A. Sanfilippo, N. Singh, A. Jabbar and Z. Cen, "PMU analytics for power fault awareness and prediction," *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, College Station, TX, USA, 2019, pp. 1-8.

[11] F. L. Grando, A. E. Lazzaretti, M. Moreto, and H. S. Lopes, “Fault classification in power distribution systems using PMU data and machine learning,” *2019 20th International Conference on Intelligent System Application to Power Systems (ISAP)*, Dec. 2019. doi:10.1109/isap48318.2019.9065966

[12] M. Rafferty and X. A. Liu, "Automatic Power System Event Classification Using Quadratic Discriminant Analysis on PMU Data," *2020 IEEE Power & Energy Society General Meeting (PESGM)*, Montreal, QC, Canada, 2020, pp. 1-6.

[13] E. Klinginsmith, R. Barella, X. Zhao, and S. Wallace, “Unsupervised clustering on PMU data for event characterization on Smart Grid,” *Proceedings of the 5th International Conference on Smart Cities and Green ICT Systems*, 2016. doi:10.5220/0005863702330240
[14] S. Pandey, A. K. Srivastava, and B. G. Amidan, “A real time event detection, classification and localization using Synchrophasor Data,” *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4421–4431, Nov. 2020. doi:10.1109/tpwrs.2020.2986019
[15] H. Li, Z. Ma and Y. Weng, "A transfer learning framework for power system event identification," *IEEE Trans. Power Syst.*, vol. 37, no. 6, pp. 4424-4435, Nov. 2022, doi: 10.1109/TPWRS.2022.3153445.
[16] A. A. Hai et al., "Transfer learning for event detection from PMU measurements with scarce labels," *IEEE Access*, vol. 9, pp. 127420-127432, 2021, doi: 10.1109/ACCESS.2021.3111727.
[17] PJM, “Primary Frequency Response Stakeholder Education,” 2017.
[18] Western Electric Coordinating Council, “WECC Off-Nominal Frequency Load Shedding Plan,” 2011.
[19] R. B. T. Blalock, B. Cummings, “Frequency Response Initiative - Generator Governor Frequency Response,” *North American Electric Reliability Corporation*, 2015.
[20] S. Homan and S. Brown, “An analysis of frequency events in Great Britain,” *Energy Reports*, vol. 6, pp. 63–69, 2020, doi: https://doi.org/10.1016/j.egyr.2020.02.028.
[21] H. T. Nguyen, G. Yang, A. H. Nielsen, and P. H. Jensen, “Challenges and research opportunities of frequency control in low inertia systems,” in *E3S Web of Conferences*, 2019, vol. 115, p. 2001.
[22] Y. Guo, K. Li, D. M. Laverty, and Y. Xue, “Synchrophasor-Based Islanding Detection for Distributed Generation Systems Using Systematic Principal Component Analysis Approaches,” *IEEE Trans. Power Deliv.*, vol. 30, no. 6, pp. 2544–2552, 2015, doi: 10.1109/TPWRD.2015.2435158.
[23] M. A. M. Ariff and B. C. Pal, “Coherency Identification in Interconnected Power System—An Independent Component Analysis Approach,” *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1747–1755, 2013, doi: 10.1109/TPWRS.2012.2217511.
[24] N. Taghipourbazargani, G. Dasarathy, L. Sankar, and O. Kosut, “A machine learning framework for event identification via modal analysis of PMU data,” *IEEE Trans. Power Syst.*, vol. 38, no. 5, pp. 4165–4176, 2023, doi: 10.1109/TPWRS.2022.3212323
[25] J. Ma, Y. V Makarov, C. H. Miller, and T. B. Nguyen, “Use multi-dimensional ellipsoid to monitor dynamic behavior of power systems based on PMU measurement,” in *2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*, 2008, pp. 1–8.
[26] M. Zhou, Y. Wang, A. K. Srivastava, Y. Wu, and P. Banerjee, “Ensemble-based algorithm for synchrophasor data anomaly detection,” *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2979–2988, 2018.
[27] L. Xie, Y. Chen, and P. R. Kumar, “Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis,” *IEEE Trans. Power Syst.*, vol. 29, no. 6, pp. 2784–2794, 2014.
[28] Recommended Oscillation Analysis for Monitoring and Mitigation Reference Document, North American Electric Reliability Corporation, available at

https://www.nerc.com/comm/RSTC_Reliability_Guidelines/Oscillation_Analysis_for_Monitoring_And_Mitigation_TRD.pdf

[29] T. Wu, S. A. Nezam Sarmadi, V. Venkatasubramanian, A. Pothen and A. Kalyanaraman, "Fast SVD Computations for Synchrophasor Algorithms," in *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1651-1652, March 2016

[30] H. Khalilinia, L. Zhang and V. Venkatasubramanian, "Fast Frequency-Domain Decomposition for Ambient Oscillation Monitoring," in IEEE Transactions on Power Delivery, vol. 30, no. 3, pp. 1631-1633, June 2015

[31] V. S. Perić and L. Vanfretti, "Power-System Ambient-Mode Estimation Considering Spectral Load Properties," in IEEE Transactions on Power Systems, vol. 29, no. 3, pp. 1133-1143, May 2014.

[32] T. Wu, V. M. Venkatasubramanian and A. Pothen, "Fast Parallel Stochastic Subspace Algorithms for Large-Scale Ambient Oscillation Monitoring," in IEEE Transactions on Smart Grid, vol. 8, no. 3, pp. 1494-1503, May 2017.

[33] J. Pierre, et. al., "Identification of electromechanical modes in power systems", *IEEE Task Force Report*, Special Publication TP462, Jun. 2012.

[34] Y. Zhu, C. Liu, and K. Sun, “Image embedding of PMU data for deep learning towards transient disturbance classification,” in Proc. IEEE Int. Conf. Energy Internet, 2018, pp. 169–174

[35] Kim HE, Cosa-Linan A, Santhanam N, Jannesari M, Maros ME, Ganslandt T. Transfer learning for medical image classification: a literature review. BMC Med Imaging. 2022 Apr 13;22(1):69. doi: 10.1186/s12880-022-00793-7.

[36] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh and G. Stefopoulos, "Missing Data Recovery by Exploiting Low-Dimensionality in Power System Synchrophasor Measurements," in IEEE Transactions on Power Systems, vol. 31, no. 2, pp. 1006-1013, March 2016, doi: 10.1109/TPWRS.2015.2413935.

[37] Guoping Liu, J. Quintero and V. M. Venkatasubramanian, "Oscillation monitoring system based on wide area synchrophasors in power systems," 2007 iREP Symposium - Bulk Power System Dynamics and Control - VII. Revitalizing Operational Reliability, Charleston, SC, 2007, pp. 1-13, doi: 10.1109/IREP.2007.4410548.

[38] H. Liu, Y. Qi, J. Zhao and T. Bi, "Data-driven subsynchronous oscillation identification using field synchrophasor measurements," *IEEE Trans. Power Del*., vol. 37, no. 1, pp. 165-175, Feb. 2022.

[39] Ju Liu. Wei Yao. Jinyu Wen. Haibo He. Xueyang Zheng. "Active power oscillation property classification of electric power systems based on SVM." *J. Appl. Math.* 2014 (SI07) 1 - 9, 2014. https://doi.org/10.1155/2014/218647

[40] I. S. Association, “IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces,” *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003)*. pp. 1–138, 2018, doi: 10.1109/IEEESTD.2018.8332112.

[41] “PSCAD,” https://www.pscad.com/

[42] “EMTP reference models for transmission line relay ...,” *Power System Relay and Control Committee*, https://www.pes-psrc.org/kb/report/074.pdf.

[43] “Positive sequence load flow (PSLF) software: GE Vernova,” geenergy-consulting, https://www.gevernova.com/consulting/software/pslf

[44] T. Mohamed et al., "Automated Detection and Classification of Critical Power System Events using ML on PMU Data," 2024 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA), Washington, DC, USA, 2024, pp. 1-6, doi: 10.1109/SGSMA58694.2024.10571526.

[45] S. Singh, “Application of Prony analysis to characterize pulsed corona reactor measurements”. *University of Wyoming*, 2003.

[46] M. Pavlovski, et al., "Hierarchical convolutional neural networks for event classification on PMU measurements," IEEE Trans. Instrum. Meas., vol. 70, pp. 1-13, no. 2514813, 2021.

[47] “1.4. Support Vector Machines,” scikit, https://scikit-learn.org/stable/modules/svm.html

[48] “Sklearn.ensemble.randomforestclassifier,”scikit,https://scikit-learn.org/stable/modules/generated/ sklearn.ensemble. Random ForestClassifier.html

[49] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). ACM

[50] B. Sun, J. Feng, and K. Saenko, “Correlation alignment for unsupervised domain adaptation,” Domain Adaptation in Computer Vision Applications, pp. 153–171, 2017. doi:10.1007/978-3-319-58347-1_8

[51] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” IEEE Trans. Neural Networks, vol. 22, no. 2, pp. 199–210, Feb. 2011. doi:10.1109/tnn.2010.2091281

[52] G. Vrbancic and V. Podgorelec, “Transfer learning with adaptive fine-tuning,” IEEE Access, vol. 8, pp. 196197–196211, 2020. doi:10.1109/access.2020.3034343

[53] Y. Guo et al., “SpotTune: Transfer learning through adaptive fine-tuning,” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019. doi:10.1109/cvpr.2019.00494

[54] B. Azimian, et al., “State and topology estimation for unobservable distribution systems using deep neural networks,” IEEE Trans. Instrum. Meas., vol. 71, pp. 1-14, Apr. 2022. doi:10.1109/TIM.2022.3167722

[55] C. J. Swinney and J. C. Woods, "GNSS Jamming Classification via CNN, Transfer Learning & the Novel Concatenation of Signal Representations," 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), Dublin, Ireland, 2021, pp. 1-9, doi: 10.1109/CyberSA52016.2021.9478250.

[56] Longjun Dong, Hongmei Shu, Zheng Tang, Xianhang Yan, Microseismic event waveform classification using CNN-based transfer learning models, International Journal of Mining Science and Technology, Volume 33, Issue 10, 2023, Pages 1203-1216, ISSN 2095-2686.

[57] J. Shi, K. Yamashita and N. Yu, "Power System Event Identification with Transfer Learning Using Large-scale Real-world Synchrophasor Data in the United States," 2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), New Orleans, LA, USA, 2022, pp. 1-5, doi: 10.1109/ISGT50606.2022.9817516.