A Machine Learning Framework for Power System Event Identification via Modal Analysis of Phasor Measurement Unit Data

by

Nima Taghipourbazargani

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree of
Doctor of Philosophy

Approved September 2023 by the
Graduate Supervisory Committee:

Oliver Kosut, Chair
Lalitha Sankar
Anamitra Pal
Gautam Dasarathy

ARIZONA STATE UNIVERSITY

December 2023

ABSTRACT

Event identification is increasingly recognized as crucial for enhancing the reliability, security, and stability of the electric power system. With the growing deployment of Phasor Measurement Units (PMUs) and advancements in data science, there are promising opportunities to explore data-driven event identification via machine learning classification techniques. This dissertation explores the potential of data-driven event identification through machine learning classification techniques.

In the first part of this dissertation, using measurements from multiple PMUs, I propose to identify events by extracting features based on modal dynamics. I combine such traditional physics-based feature extraction methods with machine learning to distinguish different event types. Using the obtained set of features, I investigate the performance of two well-known classification models, namely, logistic regression (LR) and support vector machines (SVM) to identify generation loss and line trip events in two datasets. The first dataset is obtained from simulated events in the Texas 2000-bus synthetic grid. The second is a proprietary dataset with labeled events obtained from a large utility in the USA. My results indicate that the proposed framework is promising for identifying the two types of events in the supervised setting.

In the second part of the dissertation, I use semi-supervised learning techniques, which make use of both labeled and unlabeled samples. I evaluate three categories of classical semi-supervised approaches: (i) self-training, (ii) transductive support vector machines (TSVM), and (iii) graph-based label spreading (LS) method. In particular, I focus on the identification of four event classes i.e., load loss, generation loss, line trip, and bus fault. I have developed and publicly shared a comprehensive Event Identification package which consists of three aspects: data generation, feature extraction, and event identification with limited labels using semi-supervised methodologies. Using this package, I generate eventful PMU data for the South Carolina 500-Bus synthetic network. My evaluation confirms that

the integration of additional unlabeled samples and the utilization of LS for pseudo labeling surpasses the outcomes achieved by the self-training and TSVM approaches. Moreover, the LS algorithm consistently enhances the performance of all classifiers more robustly.

ACKNOWLEDGMENTS

I would like to begin by expressing my deep gratitude to my supervisors, Professor Oliver Kosut and Professor Lalitha Sankar, whose unwavering guidance, continuous support, and remarkable patience have been instrumental throughout my journey in pursuing my PhD. Their profound knowledge and extensive expertise have been a constant source of inspiration not only in my academic pursuits but also in my daily life. I extend my sincere appreciation to Professor Gautam Dasarathy and Professor Anamitra Pal for their valuable technical insights and assistance in my research.

My heartfelt thanks go to all the members of the Sankar-Kosut Lab. Their generous assistance and camaraderie have enriched both my academic endeavors and my experience in the United States, making it a truly memorable period.

I would like to express my profound gratitude to my parents, Esmaeil and Habibeh, my brother Armin, and my sister Saba, my dear friends, Tina and Rasoul, as well as many others who have stood by my side. Their unwavering understanding and encouragement over the past years have played a crucial role in enabling me to successfully complete my studies.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Background

Given the increased penetration of intermittent renewable energy sources (e.g., solar and wind) as well as unconventional loads (e.g. electric vehicles) in the grid, real-time monitoring of system operating conditions has become more vital to ensure system reliability, stability, security, and resilience. Furthermore, power systems are prone to a variety of events (e.g. line trips and generation loss) and real-time identification of such events enhances situational awareness and assists system operators in quickly identifying events and taking suitable remedial control actions to avert disturbances in a timely manner [1]. However, power systems are inherently nonlinear with complex spatial-temporal dependencies; as a result, in many cases, it is not possible to develop accurate and sufficiently low order dynamical models that can be used to identify each distinct event [2]. This makes real-time identification of events a challenge.

Extensive research has been carried out on this problem which can be broadly categorized into traditional model based methods (see e.g., [3, 4, 5, 6]) and the state-of-the-art data-driven methods which have received considerable critical attention in recent years. The roots for the increasing significance of data-driven event identification in a wide variety of power system studies (e.g., monitoring and operation) stem from the following two main factors:

1) Model-based methods (see e.g., [3, 4, 5, 7]) involve modeling of power system components and estimation of the system states. The performance of model based methods highly depends on the accuracy of dynamic models of the system components (e.g., gener-

ators, loads, etc.). Given the ongoing integration of renewable energy technologies as well as unconventional loads with power electronic interfaces, it is difficult to develop accurate and sufficiently low order dynamical models which in turn limits the practical application of such methods in real world problems.

2) The increasing deployment of Phasor Measurement Units (PMUs) across the grid. PMUs provide time-synchronized current and voltage phasor measurements across the grid at high sampling rates (30–60 samples per seconds) thereby allowing operators to capture system dynamics with good precision and fidelity [8] which is a huge improvement over supervisory control and data acquisition (SCADA) measurements. The advancements in machine learning technologies and data science provide invaluable opportunities to investigate more advanced data-driven based event identification methods. The main advantage of such methods is their ability to distinguish between different types of power system disturbances from the collection of high-dimensional spatio-temporally correlated time-synchronized phasor measurements with high resolution rather than relying on the dynamic modeling of the power system components.

The first step in any data-driven based event identification scheme is to process the time-series data to infer information regarding the specific type of an event. Within this perspective, available literature in the context of event identification can be broadly categorized into two subgroups depending on whether they rely on the physics of the system to process the PMU data or not. (i) *model-free feature extraction methods:* References such as [9, 10, 11] extract features based on the properties (e.g., volume, rate of change of volume, center coordinates, projection of axes, etc.) of the minimum volume enclosing ellipsoid (MVEE) which is constructed from the collection of time-series PMU data. Within the same category, references [12, 13, 14, 15, 16, 17, 18, 19, 20] are examples of machine learning based event identification methods that use various model-free feature extraction techniques to transform the raw time series PMU data or their pruned version (see, for example, [18])

into numerical features that characterizes different types of events. (ii) *physics-based feature extraction methods:* References such as [8, 21, 22, 23] rely on the well-established signal processing techniques to extract physically interpretable features which can characterize various types of events based on the underlying dynamical behavior of the system. Well-studied physics-based signal processing methods such as modal analysis for feature extraction can be directly applied to PMU measurements to detect events. The key idea in such approaches, often referred to as mode decomposition, is to identify system events by thresholding the coefficients of some basis functions (see [8, 21, 22] and references therein). However, due to the diversity of power system events, choosing proper thresholds for different scenarios is not an easy task. More recently, purely data-driven classification methods using PMU measurements has begun to gain traction [24, 25].

Data-driven event identification approaches leverage machine learning and pattern recognition methods to perform statistical inference or decision-making based on available system measurements. The majority of existing literature in the context of event identification [12, 13, 14, 15, 16, 17, 18, 19, 20] belongs to the supervised learning paradigm. These methods require proper labeled data with detailed event types. However, given the fact that using expert knowledge for labeling various types of events can be expensive and tedious, proper labeled eventful PMU data is often scares.

Unsupervised and semi-supervised learning are common practices in machine learning when dealing with limited or no labeled data. Unsupervised learning aims to infer the underlying structure within the unlabeled data. Although they can distinguish between clusters of events [26, 10, 27, 28, 29, 30], they do not possess the ground truth to associate each cluster with its real-world meaning. Furthermore, when there is access to even a small amount of labeled data, supervised learning has been shown to perform better than unsupervised learning methods [26, 30]. Semi-supervised learning approaches, on the other hand, aim to label unlabeled data points using knowledge learned from a small number of labeled data points

which can significantly enhance the performance of a classification task [31]. Reference [32] uses several state-of-the-art approaches for feature extraction and semi-supervised feature reduction. Based on [32] relationships between labeled and unlabeled data are mainly extracted based on three fundamental semi-supervised assumptions as follows:

- *Manifold assumption:* data can be represented on a low dimensional manifold.
- *Cluster assumption:* data samples belonging to the same cluster are assumed to be of a same class.
- *Smoothness assumption:* samples in the dense regions share the same class label.

Reference [33] presents a framework for event detection, localization, and classification in power grids based on semi-supervised learning. A pseudo labeling (PL) technique is adopted to classify events using the convolutional neural network (CNN) backbone with cross-entropy loss. To overcome the limitations of pseudo labeling and re-training, which may lead to model homogenization and local minimum trapping, the authors propose integrating distribution alignment and uncertainty measurement techniques to enhance the performance. Distribution alignment normalizes prediction vectors and computes a running average for unlabeled samples, while uncertainty measurement selectively re-trains high-precision samples, utilizing the maximum entry of the normalized prediction vectors as an uncertainty measure. A semi-supervised event identification framework is proposed in [34] which utilizes a hybrid machine learning-based method to reduce biases of different classifiers. However, their proposed framework merely relies on the self-training semi-supervised algorithms, and fails to consider the distribution of labeled and unlabeled samples. In [35], the authors explore the application of deep learning techniques and PMU data to develop real-time event identification models for transmission networks. This is achieved by leveraging information from a large pool of unlabeled events, while also taking into account

the class distribution mismatch problem. However, the proposed approach for event identification faces challenges in terms of interpretability due to the extensive parameter space and non-linearity of the classifiers. In [36, 37], the authors propose HS3M, a novel data-driven event detection method that combines unlabeled and partially labeled data to address limitations in supervised, semi-supervised, and hidden structure learning. The approach introduces a parametric dual optimization procedure to enhance the learning objective and improve event detection accuracy. The learning problem involves optimizing a non-smooth function that may be convex or concave.

The contribtuions of this study can be broadly categorized into two main parts. The first part of the study focuses on our proposed framework for event identification in the supervised setting, while in the second part of the study, we propose a novel semi-supervised event identification approach to investigate the efficacy of including unlabeled samples on the performance of two categories of classical semi-supervised approaches. Further details regarding the specific objectives, methods, and findings of each part are provided in the following paragraphs.

Event Identification - supervised setting: We first introduce a framework that exploits the knowledge of the physics of the system to extract features, and subsequently applies ML techniques to produce a robust classifier from limited but feature-rich training data. Our key contributions are

- Characterizing events based on a set of features obtained from modal analysis of various spatio-temporally correlated PMU measurements.
- Determining an optimal subset of features that succinctly describes the system dynamics.
- Learning a set of classification models that can identify the type of an event (generation loss or line trip) using the chosen subset of features.

An overview of the proposed framework is shown in Fig. 1.1. In Step 1, using the fact that temporal effects in a power system event are driven by the interacting dynamics of the system components, we use mode decomposition as the framework with which to extract features. Considering the robustness of matrix pencil method (MPM) against noise [38, 39], we use it as the main tool to perform mode decomposition. Using different channels of PMU measurements (magnitudes and angles for positive sequence voltages and currents, and frequency) obtained from multiple PMUs, we apply multi-signal matrix pencil method (MSMPM) to find a single set of modes that best represent the underlying dynamical behavior of the system. Using this approach, one can obtain a characterization of power system events as a set of features, e.g., via angular frequencies, damping factors, and the corresponding residues. However, extracting features using all channels of PMU measurements across multiple PMUs will inevitably lead to a high-dimensional feature set, and thus, a key question is to determine which subset of these features can guarantee accurate classification performance.

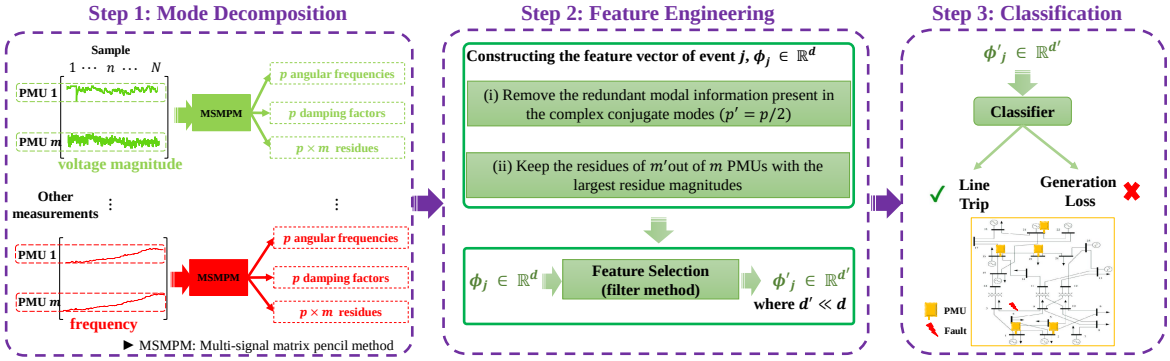


Figure 1.1: Overview of the proposed event identification framework

Our goal in Step 2 is to avoid overfitting while ensuring that multiple events can be distinguished by the same set of sufficient features. Common statistical approaches for selecting features include filter methods, wrapper methods, and embedded methods [40]. Wrapper methods interface with a classification model in an iterative manner, choosing

the features that most improve the performance of a classification model on a certain set of training dataset. Embedded methods involve integrating the feature selection algorithm with the classification model [40]. Filter methods use statistical measures to score the dependence between the features and target variable and select the most relevant features with the highest scores. Since filter methods in contrast with wrapper and embedded methods, are independent from classification models, they are computationally inexpensive and are more efficient for real time applications. Thus, due to its ease of analysis, we use filter methods in order to choose the best subset of features.

Finally, using the extracted features via steps 1 and 2, in Step 3 we investigate the performance of two well-known classification models, namely, logistic regression (LR) and support vector machine (SVM) with radial basis function (RBF) in identifying generation loss and line trip events. We use two datasets, one synthetic and one real, to evaluate the performance of each classifier. The synthetic dataset is obtained by simulating generation loss and line trip events in the Texas 2000-bus synthetic grid [41] using the power system simulator for engineering (PSS®E). The proprietary dataset is obtained from a large utility in the USA involving measurements from nearly 500 PMUs. It includes a total number of 70 labeled events (i.e., 23 generation loss and 47 line trip events). As detailed in Chapter 4, our results on both real and simulated datasets indicate that the proposed framework is promising for identifying the two types of events.

Event Identification - semi-supervised setting:

The existing literature on neural network-based event identification methods is marked by certain limitations and challenges. These encompass restricted interpretability in feature extraction, elevated computational intricacy, and the necessity for meticulous parameter calibration. Moreover, it is worth noting that, to the best of the authors' knowledge, a thorough investigation into the ramifications arising from the initial distribution of labeled and unlabeled samples has not been undertaken. This study introduces a semi-supervised

event identification framework to explore the potential benefits of incorporating unlabeled samples in enhancing the performance of the event identification task. To this end, we thoroughly investigate and compare the performance of various semi-supervised algorithms, including: (i) self-training with different base classifiers (i.e., support vector machine with linear kernel (SVML) as well as with radial basis function kernel (SVMR), gradient boosting (GB), decision trees (DT), and k-Nearest Neighbors (kNN)), (ii) transductive support vector machines (TSVM), and (iii) graph-based label spreading (LS) to explore their effectiveness. We chose these classical semi-supervised models for two primary reasons: firstly, the wide array of proposed semi-supervised classification algorithms in the past two decades (see, [42], and references therein) necessitates a comprehensive understanding of which models are most suitable and efficient for event identification; and secondly, they provide a more clear illustration and intuition of the impact of incorporating unlabeled samples compared to more advanced methods. Although there may not be a one-size-fits-all solution, each method has its own advantages and disadvantages, and it is important to evaluate their suitability. Notably, our experiments consistently illustrate the superior performance of the graph-based LS method compared to other approaches. Even in worst-case scenarios where the initial distribution of labeled and unlabeled samples does not necessarily reflect the true distribution of event classes, the graph-based LS method stands out in robustly and significantly enhancing event identification performance. Our key contributions are as follows:

- Introduction of a semi-supervised event identification framework that leverages physically interpretable features derived from modal analysis of PMU data.

- Thorough exploration of the influence of the initial distribution of labeled and unlabeled samples, along with the quantity of unlabeled samples, on the efficacy of diverse semi-supervised event identification techniques.

- Development of an all-inclusive Event Identification package [1] comprising of an event generation module based on the power system simulator for engineering (PSS®E) Python application programming interface (API), a feature extraction module utilizing methodologies from our previous research [43], and a semi-supervised classification module.

1.2 Outline of Thesis

The Thesis is structured as follows: Chapter 2 provides a brief introduction to modal analysis and MPM. Chapter 3 presents the proposed event identification framework in the supervised setting, feature extraction and selection techniques, and simulation results based on different feature selection methods. In Chapter 4, our proposed semi-supervised event identification framework and the simulation results are presented. The concluding remarks and future work are discussed in Chapter 5.

[1] https://github.com/SankarLab/PSMLEI-public

Chapter 2

FEATURE EXTRACTION AND ENGINEERING OF PMU TIME SERIES DATA

The first step in identifying a system event from PMU data is to extract the relevant features from the data stream. Due to the high sampling rate of the PMU data, one could plug in the raw data into a machine learning model. However, it is advantageous to use a set of delineating features that are likely to contain information regarding the event type (henceforth referred to as event class). Using the fact that temporal effects in a power system are driven by the interacting dynamics of system components, we propose to use mode decomposition as the framework with which to extract features. More specifically, we assume that each PMU data stream after an event consists of a superposition of a small number of dominant dynamic modes. Thus, the features will be the frequency and damping ratio of these modes, as well as the residual coefficients indicating the quantity of each mode present in each data stream. Note that to ensure an accurate estimation of the modes, we use the detrended PMU measurements prior to any modal analysis [44].

Several modal analysis techniques such as MPM, Prony analysis and dynamic mode decomposition [45, 46] have been proposed in literature. Relying on earlier observations that MPM is more robust to noise relative to the above mentioned methods, we will use MPM as the mode decomposition technique. In general, every PMU has multiple measurement channels, including positive sequence voltage magnitude (VPM) and corresponding angle (VPA), positive sequence current magnitude (IPM), and corresponding angle (IPA), and frequency (F). Furthermore, multiple PMUs across the grid can capture the dynamic response of the system after an event through different measurement channels. Therefore, for a chosen measurement channel, we will use the MSMPM to obtain one optimum set of mode estimates which can accurately represent the underlying dynamic behavior of the

system [38].

Oftentimes only a small number of modes are triggered after an event. In a noise-free system, it is fairly easy to extract these modes. However, in a noisy system, there exist many other low energy modes that are more likely related to the minor noise variations and can make the identificaiton of the events harder. To ensure accurate classifiers we use the low rank approximation of the Hankel matrix constructed from PMU measurements which allows (i) reducing the effect of noise on the accuracy of mode estimation, and (ii) extracting a small number of dominant modes from noisy PMU measurements.

So in Section 2.1, we briefly explain modal analysis as a method to capture signatures of an event. Then we discuss the background and theory behind single signal and multi-signal MPM in Section 2.2. Finally, in Section 2.3, we discuss the low rank approximation of the Hankel matrix obtained from PMU measurements to estimate the sufficient number of dominant modes.

2.1 Modal Representation of PMU Measurements

Consider an electric grid with m installed PMUs. Recall that each PMU has multiple channels through which we can obtain different types of measurements relative to the bus where the PMU is installed. For the sake of clarity, we focus on one channel (e.g., VPM). Let $y_i(n) \in \mathbb{R}$, $i = 1, \dots, m$, and $n = 0, \dots, N-1$, denote the VPM measurement obtained from i^{th} PMU at sample n with a sampling period of T_s. We assume that $y_i(n)$ after an event consists of a superposition of p common damped sinusoidal modes as

$$y_i(n) = \sum_{k=1}^{p} R_k^{(i)} \times (Z_k)^n + \epsilon_i(n), \quad i = 1, \dots, m \tag{2.1}$$

where $\epsilon_i(n)$ represents the noise in the i^{th} PMU measurement and Z_k is the k^{th} mode associated with the event. We represent each mode as $Z_k = \exp\left(\lambda_k T_s\right)$ where $\lambda_k = \sigma_k \pm j\omega_k$ and σ_k and ω_k are the damping factor and angular frequency of the k^{th} mode, respectively. Furthermore, residue $R_k^{(i)}$ corresponding to each mode k and i^{th} PMU measurement is defined

by its magnitude $|R_k^{(i)}|$ and angle $\theta_k^{(i)}$. Note from (4.1) that for all m PMU measurements, there is a single set of modes (i.e., $\{Z_k\}_{k=1}^{p}$). However, the corresponding residue of each mode will be distinct for each PMU measurement.

Let $\mathbf{Y}^{(i)} = [y_i(0), \dots, y_i(N-1)]^T \in \mathbb{R}^N$, $\mathbf{R}^{(i)} = [R_1^{(i)}, ..., R_p^{(i)}]^T \in \mathbb{R}^p$, and $\mathbf{Z} = [Z_1, \dots, Z_p]^T$. We define $\mathcal{V}_{\mathbf{Z}}(N) \in \mathbb{R}^{p \times N}$ as the Vandermonde matrix of the modes, $\mathbf{Z}$, as

$$\mathcal{V}_{\mathbf{Z}}(N) = \underbrace{\begin{bmatrix} 1 & Z_1 & \cdots & Z_1^{N-1} \\ 1 & Z_2 & \cdots & Z_2^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & Z_p & \cdots & Z_p^{N-1} \end{bmatrix}}_{p \times N} \tag{2.2}$$

Then (4.1), in the absence of noise can be written in compact form as

$$\mathcal{V}_{\mathbf{Z}}(N)^T \mathbf{R}^{(i)} = \mathbf{Y}^{(i)} \tag{2.3}$$

Once the modes, $\mathbf{Z}$, are estimated, the corresponding residues $\mathbf{R}^{(i)}$ for each PMU measurement stream, $i = 1, \dots, m$, can be obtained by solving (2.3).

2.2 Multi-Signal Matrix Pencil Method

As mentioned earlier, considering the robustness of MPM against noise, it will be used as the main tool to estimate the parameters of (4.1). The MPM involves constructing the Hankel matrix over a block of N samples obtained from the i^{th} PMU as

$$\mathcal{H}_i = \underbrace{\begin{bmatrix} y_i(0) & y_i(1) & \cdots & y_i(L) \\ y_i(1) & y_i(2) & \cdots & y_i(L+1) \\ \vdots & \vdots & \ddots & \vdots \\ y_i(N-L-1) & y_i(N-L) & \cdots & y_i(N-1) \end{bmatrix}}_{(N-L) \times (L+1)} \tag{2.4}$$

where L is the pencil parameter. We choose $L = N/2$, since it is known that this will result in the best performance of the MPM in a noisy environment (i.e., the attainment of a variance close to the Cramer-Rao bound)[38].

Using (2.4), let $\mathcal{H}_i^{(1)}$ and $\mathcal{H}_i^{(2)}$ be the matrices consisting of the first and last L columns of $\mathcal{H}_i$, respectively. In a noise free setting , as a consequence of (4.1), we can write $\mathcal{H}_i^{(1)}$ and $\mathcal{H}_i^{(2)}$ as

$$\mathcal{H}_i^{(1)} = \mathcal{V}_{\mathbf{Z}}(N-L)^T \mathbf{R}_D^{(i)} \mathcal{V}_{\mathbf{Z}}(L) \tag{2.5a}$$

$$\mathcal{H}_i^{(2)} = \mathcal{V}_{\mathbf{Z}}(N-L)^T \mathbf{R}_D^{(i)} \mathbf{Z}_D \mathcal{V}_{\mathbf{Z}}(L) \tag{2.5b}$$

where

$$\mathbf{Z}_D = \text{diag}(Z_1, Z_2, ..., Z_p), \tag{2.6}$$

$$\mathbf{R}_D^{(i)} = \text{diag}(R_1^{(i)}, R_2^{(i)}, ..., R_p^{(i)}). \tag{2.7}$$

Then, the matrix pencil is defined as

$$\mathcal{H}_i^{(2)} - \lambda \mathcal{H}_i^{(1)} = \mathcal{V}_{\mathbf{Z}}(N-L)^T \mathbf{R}_D^{(i)} (\mathbf{Z}_D - \lambda \mathbf{I}) \mathcal{V}_{\mathbf{Z}}(L) \tag{2.8}$$

and from (2.8), it is clear that for any $\lambda = Z_k$, $k = 1, \ldots, p$, the k^{th} row of $\mathbf{Z}_D - \lambda \mathbf{I}$ becomes zero and the rank of $\mathcal{H}_i^{(2)} - \lambda \mathcal{H}_i^{(1)}$ is reduced by one. Therefore, the parameters $\{Z_k\}_{k=1}^p$ are the generalized eigenvalues of the matrix pair $(\mathcal{H}_i^{(2)}, \mathcal{H}_i^{(1)})$ [47].

The matrix pencil method described above, which focuses on the measurements obtained from a single PMU, may be extended to find a single set of modes which best represent the underlying dynamical behavior of a set of measurements obtained from multiple PMUs. This is done by vertically concatenating Hankel matrices $\mathcal{H}_1, \ldots, \mathcal{H}_m$ corresponding

to each PMU measurements over a block of N samples as

$$\mathbf{H} = \underbrace{\begin{bmatrix} \mathcal{H}_1 \\ \vdots \\ \mathcal{H}_i \\ \vdots \\ \mathcal{H}_m \end{bmatrix}}_{m(N-L)\times(L+1)} \tag{2.9}$$

and the same method which is used for a single measurement stream (see (2.5) to (2.8)) is applied to the matrix $\mathbf{H}$ to identify a set of modes $\{Z_k\}_{k=1}^{p}$. Finally, we find the residues corresponding to each mode k and i^{th} PMU measurements by solving (2.3).

2.3 Model Order Approximation

Following the assumption that PMU measurements after an event can be represented as a superposition of p dynamic modes and considering the fact that only a small number of modes are enough to represent the underlying dynamical behavior of the system ($p \ll L$), one can show that $\text{rank}(\mathbf{H}) = p$ for noise free PMU measurements [48]. However, in practice PMU measurements are noisy and $\text{rank}(\mathbf{H}) > p$. In this case, for a given p, we can partly eliminate the noise by using the singular value decomposition (SVD) to find the rank p approximation of $\mathbf{H}$, denoted as $\mathbf{H}_p$. The approximation $\mathbf{H}_p$ results from keeping the p largest singular values of $\mathbf{H}$ (the remaining singular values are replaced by zero). Using $\mathbf{H}_p$ in MPM also provides minimum variance in the estimation of modes in noise-contaminated PMU measurements (we refer readers to [49, 50] for a comprehensive study of MPM performance in the presence of noise in the PMU measurements).

In practice, however, the parameter p is not known. A reliable way to approximate p in (4.1) is to find the best p over all the events in our dataset. To this end, we define the rank

p approximation error of $\mathbf{H}$ as

$$E_p = \frac{\|\mathbf{H} - \mathbf{H}_p\|_F}{\|\mathbf{H}\|_F} \tag{2.10}$$

where $\|\mathbf{H}\|_F$ is the Frobenius norm of the matrix $\mathbf{H}$. Furthermore, to verify that the estimated value of parameter p is sufficient for capturing the underlying dynamics of the system, we evaluate the reconstruction error of each PMU measurements, denoted as E_i, $i = 1, \ldots, m$, as

$$E_i = \frac{\|\hat{\mathbf{Y}}^{(i)} - \mathbf{Y}^{(i)}\|}{\|\mathbf{Y}^{(i)}\|} \tag{2.11}$$

where $\mathbf{Y}^{(i)}$ is the original measurement stream and $\hat{\mathbf{Y}}^{(i)}$ is the reconstructed one based on the mode decomposition.

Using the equations (2.10) and (2.11), the value of the parameter p is determined such that it ensures both E_p and E_i (obtained from various PMU channels) are less than a predefined threshold for all the events in the dataset. Throughout the report, we consider that this threshold is 1%.

To characterize the dynamic response of the power system after an event, modal analysis is conducted on each PMU channel (i.e., VPM, VPA, IPM, IPA, and F) obtained from multiple locations across the grid. For instance, using VPM channel measurements from m PMUs, we obtain a set of features consisting of p angular frequencies, p damping factors and the corresponding magnitude and angle of the residues for each of the m PMUs and p modes. Although mode decomposition is meant to focus on only the physically meaningful features of the dataset, there are still simply too many of them ($m \approx 500$ and $p = 6$ in our dataset). To avoid overfitting while ensuring that multiple events can be distinguished by the same set of features, a necessary pre-processing step is to select relevant and most informative features. To this end, we propose a two-step approach to reduce the features into a more manageable number. In the first step, we select a subset of features by removing the redundant modal information present in the complex conjugate modes and eliminating the

smallest residue magnitude to construct a vector of features that characterizes the dynamic response of the system after an event. The second step is to select the most informative and relevant features using a filter method. The details are provided in the following subsections.

2.4 Constructing the Feature Vector

As discussed in Chapter 2.3, parameter p represents the number of dominant modes in the PMU data streams and can be obtained by finding the best rank p approximation of $\mathbf{H}$. Based on our simulation results, we consider $p = 6$ is sufficient to ensure the accuracy and robustness of the estimated modes against noise (see Section. 3.2 for more details). In general, these modes can be real or complex conjugate pairs. In our dataset, typically these modes are complex conjugates (i.e., 3 complex conjugate pairs, yielding 6 modes in total). In order to remove redundant modal information present in the complex conjugate modes, we only keep one mode from each pair. Then, we choose $p' = p/2$ where p' is the number of distinct modes that will be used in the vector of features of each event. However, for a small portion of the events, modal analysis may result in different combinations of real and complex conjugate modes. For example, if $p = 6$, we could obtain 2 complex conjugate modes and 2 real modes for 4 distinct modes in total. In that case, we need to specify the number of modes that are used for feature selection, such that we obtain the same number of features for all the events. Since the residue coefficients indicate the quantity of each mode present in each PMU data steam, if there are any real modes in decomposition, we sort the modes based on their average residue across all the PMUs and we choose $p' = p/2$ modes with the largest average residues to be included in the vector of features. (The average residue corresponding to the k^{th} mode is $\frac{1}{m}\sum_{i=1}^{m}|R_k^{(i)}|$.) Moreover, since only a small portion of the PMUs ($m' < m$) capture the dynamic response of the system after an event, we only keep the residues of m' PMUs with the largest magnitudes in the vector of features.

Using the VPM channel measurements obtained from multiple PMUs, we define a row

vector of features, $\mathcal{F}_{\text{VPM}}$, as follows:

$$\begin{aligned}\mathcal{F}_{\text{VPM}} =&[\{\omega_k : k = 1, \dots, p'\}, \{\sigma_k : k = 1, \dots, p'\},\\ &\{|R_k^{(i)}| : i = 1, \dots, m', k = 1, \dots, p'\}\\ &\{\theta_k^{(i)} : i = 1, \dots, m', k = 1, \dots, p'\}]\end{aligned} \tag{2.12}$$

which consists of p' angular frequencies, p' damping factors and the corresponding magnitude and angle of the residues for each of the m' PMUs (with the largest residue magnitudes) and p' modes.

To make a meaningful comparison of the features, it is important to sort them consistently across all the events. We sort the modes based on their average residue across all the m' PMUs. In our notation in (4.2), $k = 1$ represents the mode with the largest average residue and $k = p'$ represents the mode with the smallest average residue. Moreover, for a given mode k, the residues for different PMUs, $i = 1, \dots, m'$, are sorted in a descending order based on the magnitude of their residues, $|R_k^{(i)}|$ and we use the same order to sort the corresponding $\theta_k^{(i)}$. Note that, for each mode, we do not expect that the same PMU to always have the largest residue. Thus, the same PMU could be represented using a different index.

In a similar manner, we obtain the set of features corresponding to other PMU channels, i.e., VPA, IPM, IPA, and F. Then each event j can be described as a vector of features as

$$\boldsymbol{\phi}_j = [\mathcal{F}_{\text{VPM}}, \mathcal{F}_{\text{VPA}}, \mathcal{F}_{\text{IPM}}, \mathcal{F}_{\text{IPA}}, \mathcal{F}_{\text{F}}]^T \tag{2.13}$$

where each $\mathcal{F}_s$, $s \in \{\text{VPM, VPA, IPM, IPA, F}\}$ consists of the modal analysis results corresponding to the selected PMU channel. Hence, assuming n_{ch} represents the number of channels at a PMU that are used for modal analysis, each event j can be described as a set of d features $\boldsymbol{\phi}_j = [\varphi_1, \dots, \varphi_d]^T \in \mathbb{R}^d$, where $d = 2n_{ch}(p' + m'p')$. For instance, for $m' = 25$, $p' = 3$, and using $n_{ch} = 5$ channels, we obtain a total of $d = 780$ features. When the number of labeled events is small (e.g., 70 labeled events in our proprietary dataset) which is typically the case in practice, a 780-dimensional feature set can be extremely large.

2.5 Feature Selection Using Filter Methods

Filter methods employ some measure of dependence between a feature and the event class to rank the features, and retain only the top ranked features. As the measure of dependence, various statistical tests, including one-way analysis of variance F-value test, sure independence screening, mutual information, Pearson correlation, and Kendall correlation have been used in literature [40]. Given that we are focusing on a classification setting, we are interested in determining the correlation between numerical features and a categorical target variable. To this end, we use F-value test (F)[51], sure independence screening (S)[52], and mutual information (M) [53] to quantify the correlation between features and the target variable. We use the off-the-shelf packages in Python to estimate the mutual information between discrete and continuous variables based on the nearest neighbor method (see [53] for more details).

As detailed in 2.4, each event j can be described as a set of d features $\boldsymbol{\phi}_j = [\varphi_1, \dots, \varphi_d]^T \in \mathbb{R}^d$ and a label ξ_j which describes the class of the event (i.e., line trips and generation loss events are labeled as 0 and 1, respectively). We define our dataset, $D = \{\boldsymbol{\phi}_j, \xi_j\}_{j=1}^{N_e}$ where N_e is the total number of labeled events. We use the Z-score to normalize our dataset [54]. Then we split the dataset into a training dataset with N_{tr} samples and a test dataset with N_{te} samples, denoted as D_{train} and D_{test}, respectively. In a standard filter method, we compute the correlation of each feature φ_i, $i = 1, \dots, d$ and the target variable, $\boldsymbol{\Xi} = [\xi_1, \dots, \xi_j, \dots, \xi_{N_{tr}}]^T \in \{0, 1\}^{N_{tr}}$ in the training dataset. Then we sort the features based on their correlation measure and then keep the d' features with the highest correlation.

However, due to the small number of samples, we need a more robust way of choosing the features. Therefore we will rely on a well-known approach in machine learning, bootstrapping. Bootstrapping is a technique of sampling with replacement to create multiple datasets from the original dataset, thereby selecting the most informative features with

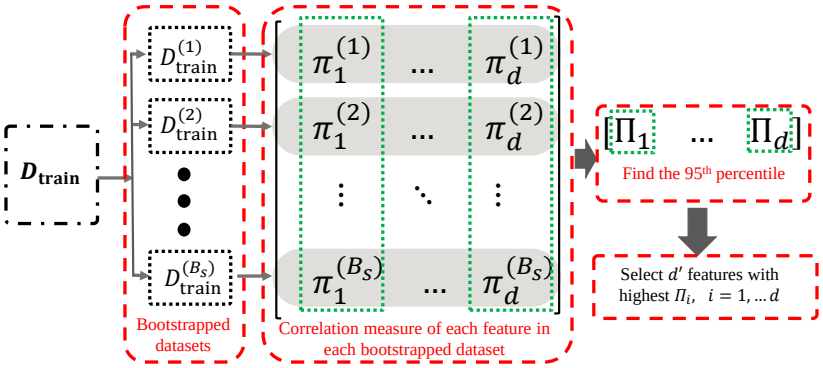


Figure 2.1: Overview of the feature selection step

some degree of statistical confidence. Note that the size of each bootstrapped dataset is the same as the original dataset.

The overview of feature selection step is shown in Fig. 2.1. The process begins by constructing B_s bootstrapped datasets, denoted as $D_{\text{train}}^{(b)}$, $b = 1, \ldots, B_s$, from the original training dataset, D_{train}. We define, $\pi_i^{(b)}$ as the correlation measure of feature φ_i and target variable Ξ over the b^{th} bootstrap samples. In order to robustly find a subset features, we compute the 95th percentile of the correlation measures of each feature over the B_s bootstrapped datasets and select d' features with the highest 95th percentiles. Using the selected d' features, we obtain a reduced order training dataset, denoted as D'_{train}.

We will also use bootstrapping for the classification (see Section 3.1). We have done extensive experiments without bootstrapping which confirms the advantage of using it for both feature selection as well as the classification. In the interest of clarity, we did not include those results in this report.

Chapter 3

EVENT IDENTIFICATION IN SUPERVISED SETTING

3.1 Proposed Event Identification Framework

The final step in our proposed framework for event identification is to use the subset of features (as described in Sections 2.4 and 2.5) to learn a classification model by finding decision boundaries between various event classes in the feature space. With any ML model, there is a tradeoff inherent in the choice of complexity of the classification model. A simpler model may be more easily interpreted and is less likely to encounter overfitting problems whereas a more complex model may be more capable of uncovering subtle characteristics of the underlying phenomena and may thereby perform better. Therefore, to investigate the impact of the model complexity on the accuracy of event classification, two well-known classifiers, namely, LR and SVM with RBF kernels are used to identify the two classes of events in our dataset (we refer readers to [54] for details of the two classification models). The LR is a relatively simple model compared to the SVM with RBF kernels.

In order to validate the performance of each classification model, we split the dataset into a training and a test datasets. All the filter methods are implemented on the training dataset to find the most relevant and informative subset of features and obtain reduced order training and test datasets, denoted as D'_{train} and D'_{test}, respectively. Due to the limited number of labeled generation loss and line trip events, we again use the bootstrap technique as a tool for assessing statistical accuracy. Using bootstrap sampling helps to address the problem of limited training samples and therefore justifies using the test data for validation of specific parameters, namely, the number of features to pick and the choice of the classification model.

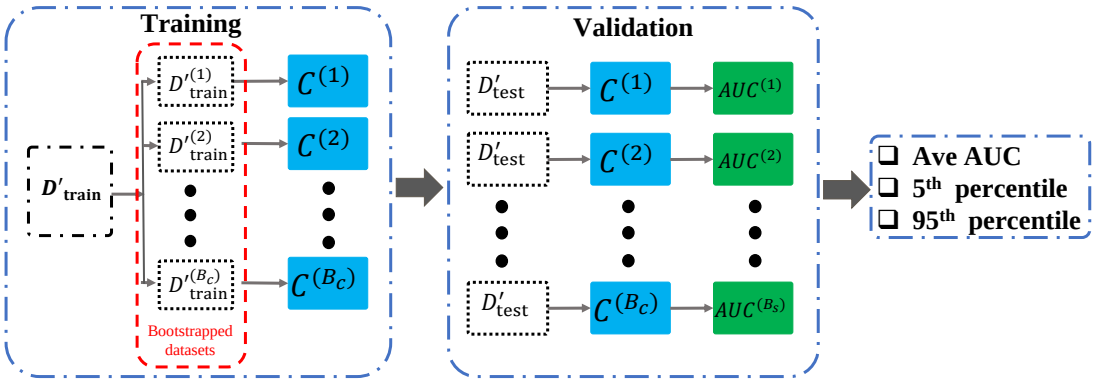


Figure 3.1: Overview of the model validation (D'_{train}, and D'_{test} are reduced order training and test data, respectively. $C^{(i)}$: learned model from the i^{th} bootstrapped reduced order training data.)

Using the reduced order training dataset, D'_{train}, we generate B_c reduced order bootstrapped datasets, denoted as $D'^{(b)}_{\text{train}}$, $b = 1, \ldots, B_c$, to learn a classification model, $C^{(b)}$, and classify the events in the D'_{test}. To evaluate the performance of a chosen classifier (for example, LR), we use the area under curve (AUC) of the receiver operator characteristic (ROC), which characterizes the accuracy of the classification for various discrimination thresholds [40]. (The discrimination threshold determines the probability at which the positive class is chosen over the negative class.) The ROC plot shows the relation between the true positive rate and the false positive rate at various threshold settings. The ROC AUC value is bounded between 0 and 1. The closer AUC to 1, the classifier has a better ability to classify the events. To quantify the accuracy of the learned classifier on the test dataset, we compute the average AUC, and the corresponding 5th and 95th percentiles of the AUC values over all the bootstrapped datasets. The aforementioned steps are summarized in Fig. 3.1.

3.2 Simulation Results

In order to evaluate the performance of the proposed framework for event identification, two different datasets are considered in this study. The first one is obtained from the

dynamic simulation of line trip and generation loss events in the Texas 2000-bus synthetic grid [41] using the power system simulator for engineering (PSS®E). The second dataset is a proprietary dataset with labeled generation loss and line trip events obtained from a large utility in the USA involving measurements from nearly 500 PMUs.

In the remainder of the section, we present our results for each dataset including: (i) the sufficient number of distinct modes, p', using the measurements obtained from different PMU channels, (ii) the reconstruction error of the PMU measurements using modal information obtained from MSMPM, and (iii) the performance of LR and SVM in identifying the events using the subset of features (as explained in Chapter 3).

3.2.1 Case 1: Synthetic Dataset

In order to generate synthetic PMU data with labeled events, we use the PSS®E dynamic data of the the Texas 2000-bus synthetic grid [55]. The single line diagram of the Texas 2000-bus system is shown in Fig. 3.2.

We allow the system to be in the normal operation condition for 1 second. Then, we apply a line trip or generation loss at time $t = 1$ and run the dynamic simulation to $t = 20$ seconds. The simulation time step for dynamic simulations is set to 0.0083 secs. In order to collect data at a rate of 30 sample/sec (PMU sampling rate), we record the measurements at each $0.033/0.0083 \approx 4$ time steps. We assume that 95 of the 500 kV buses (which are chosen randomly) across the grid are equipped with PMU devices. We generate a total number of 800 events including 400 generation loss and 400 line trip events. For each event class, 200 events are simulated under the normal loading and 200 with 80% of normal loading. Since PSS®E does not have any channel to directly measure the branches currents, only VPM, VPA, and F channels are used for extracting the features from the PMU measurement. To capture the dynamic response of the system, we use $N = 300$ samples after the exact start of an event.

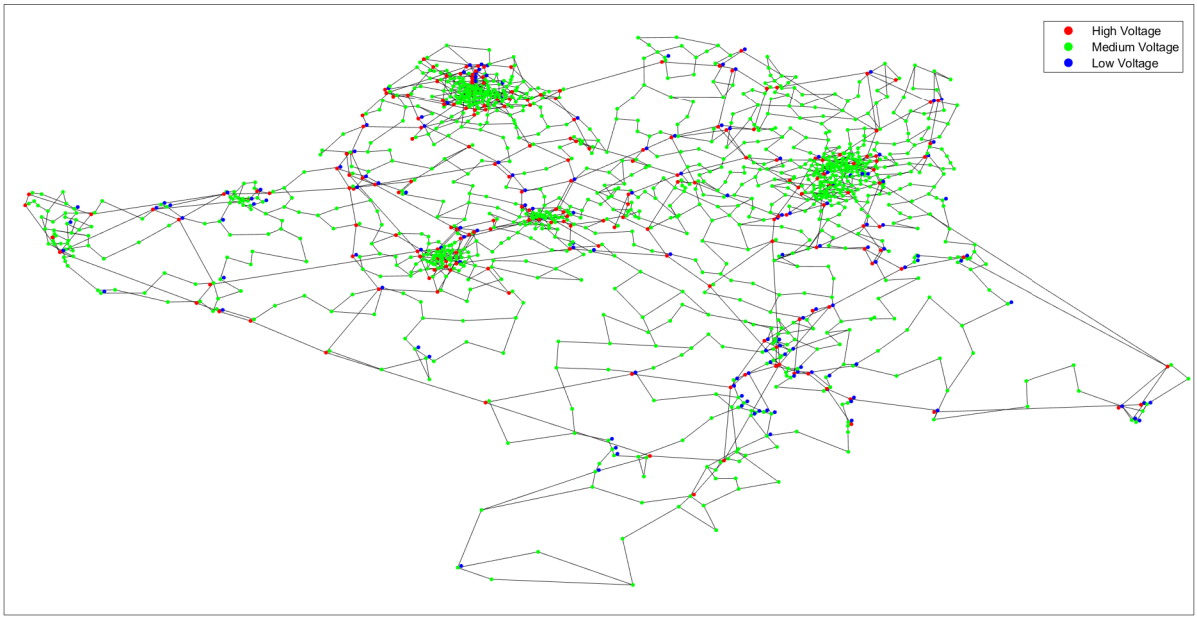


Figure 3.2: The single line diagram of the Texas 2000-bus synthetic grid.

To evaluate the performance of the classification models, we split our synthetic data into training and test datasets with 600 and 200 samples, respectively. The training dataset is used for feature engineering and learning the models and the test dataset is only used for evaluation and comparison of the models.

Using the VPM measurements obtained from 95 PMUs after a line trip event, we construct the matrix $\mathbf{H}$ based on (2.4). In Fig. 3.3, we illustrate the rank p approximation error of the matrix $\mathbf{H}$ that is given by (2.10). The matrix $\mathbf{H}$ is constructed over a block of $N = 300$ samples after the exact start time of the event with the pencil parameter of $L = 150$. Observe that if one chooses a threshold of 1% for the approximation error, then we only require $p = 6$ largest singular values; this is the case for all the events in our synthetic dataset.

Fig. 3.4 illustrates the envelope of the reconstruction error of all the PMU measurement streams (that are obtained from VPM channel) in the synthetic dataset. The average reconstruction error of the PMUs over all the events in our dataset is less than 1%. As detailed in the Section 2.3, this implies that using $p = 6$ modes is sufficient for capturing the underlying

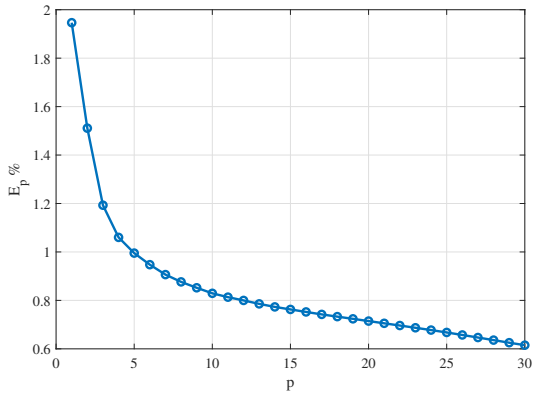


Figure 3.3: Rank p approximation error of the matrix $\mathbf{H}$ (which is obtained from VPM measurements from 95 PMUs after a line trip event) for different values of p.

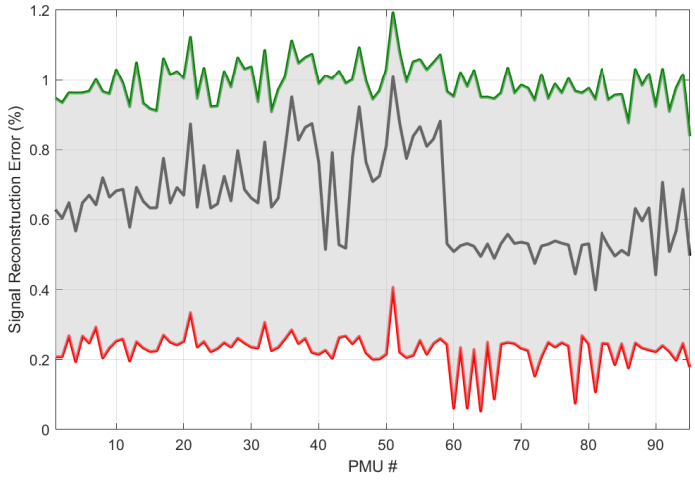
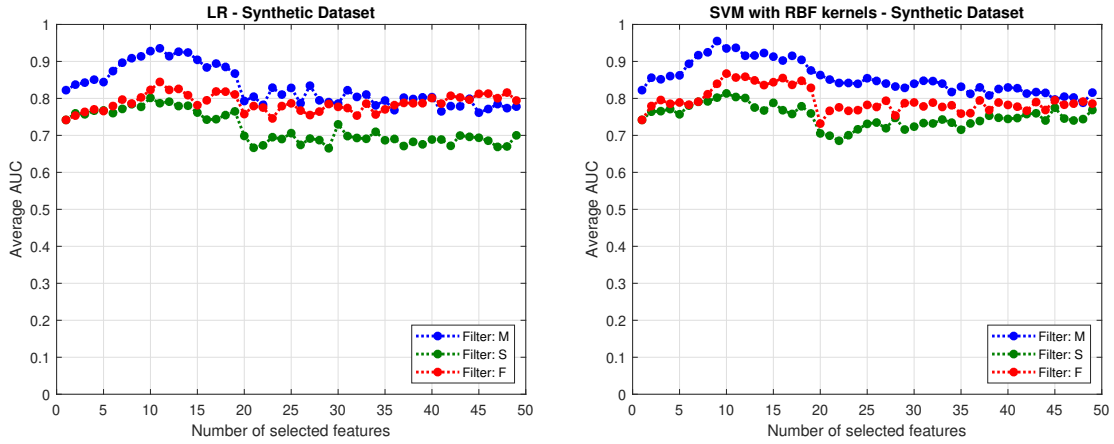


Figure 3.4: Envelope of the reconstruction error of all the PMU measurement streams that are obtained from VPM channel after 800 events in our dataset. Red, gray, and green lines represent the minimum, average and maximum reconstruction error, respectively.

dynamics of the system after an event.

As discussed in Section 2.4, to remove the redundant information present in the complex conjugate modes, we use $p' = 3$ distinct modes in the vector of features for each event. Furthermore, to determine the parameter m', we use the normalized residue for each PMU

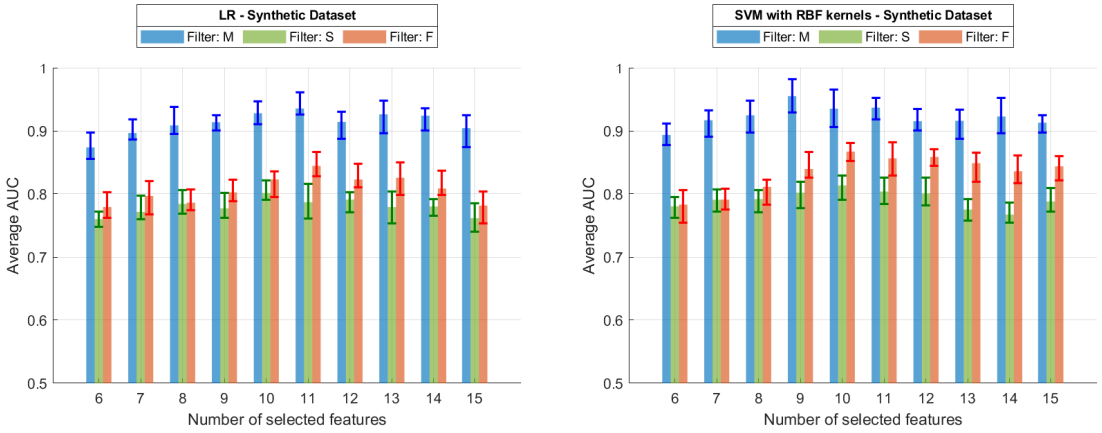
with respect to the one with the largest magnitude and pick the smallest number of PMUs for which more than 95% of the PMUs are less than a certain threshold. Based on this approach, we choose $m' = 20$ PMUs to capture the most significant residues in our synthetic dataset. Therefore, considering $p' = 3$, $m' = 20$, and $n_{ch} = 3$, each event in the synthetic dataset is characterized using $d = 378$ features. Then, we generate $B_s = 200$ bootstrapped datasets from the original training dataset to retain the features with the highest correlation.



(a) (b)

Figure 3.5: Performance of the classification models (a) LR, and (b) SVM in terms of average AUC over $B_c = 200$ bootstrapped datasets with respect to the number of selected features in the synthetic dataset.

Figure 3.5 shows the performance of the classification models, namely, (a) LR, and (b) SVM in terms of average AUC over $B_c = 200$ bootstrapped datasets with respect to the number of selected features. The selected features are the ones with the highest 95th percentiles obtained from various correlation measures (i.e., F, S, and M as detailed in Section 2.5). To further elaborate the performance of each classifier, using a subset of 6 to 15 features obtained from various correlation measures, the average AUC score as well its corresponding 5th and 95th confidence intervals are shown in Fig. 3.6.



(a) (b)

Figure 3.6: Performance of the classification models (a) LR, and (b) SVM in terms of the average AUC with respect to the number of selected features in the synthetic dataset. The error bars represent the 5th and 95th percentiles of the AUC scores.

Based on the simulation results, using the mutual information as the correlation measure to select a subset of features will result in a better performance of both classifiers. This is due to the fact that F-value and sure independence screening only consider the linear dependence of the features with the target variable whereas mutual information can also capture non-linear dependencies. The selected features include the angular frequency and first few residue magnitudes corresponding to the first mode of the VPM, VPA, and F measurement channels. Furthermore, it is clear that SVM with RBF kernel has a slightly better performance than LR in identifying the two classes of the events in our synthetic dataset. It is also clear that using a subset of about 10 features obtained from mutual information will result in the best performance of both classifiers. The error bars represent the 5th and 95th percentile of the AUC scores over B_c bootstrapped datasets and are an indication of the robustness of each learned classifier.

3.2.2 *Case 2: Proprietary Datset*

To further investigate the performance of our proposed framework, we use a proprietary PMU data obtained from a large utility in the USA involving measurements from nearly 500 PMUs. A total of 70 labeled events including 23 generation loss and 47 line trip events are used in this study. To characterize the dynamic response of the system after an event, VPM, VPA, IPM, IPA, and F measurement channels from multiple PMUs over a block of $N = 300$ samples (after the exact start time of the event) are used for extracting the features as discussed in Section 2.4. The envelope of the reconstruction error of all the PMU measurement streams (that are obtained from VPM channel) in the synthetic dataset. Fig. 3.7 illustrates that using $p = 6$ modes, the average reconstruction error of the PMUs over all the events in our real dataset is less than 1%. Using the same approach that is used in case 1, the parameters $p' = 3$ and $m' = 25$ are used to construct the vector of features for each event, thereby obtaining a total number of $d = 780$ features.

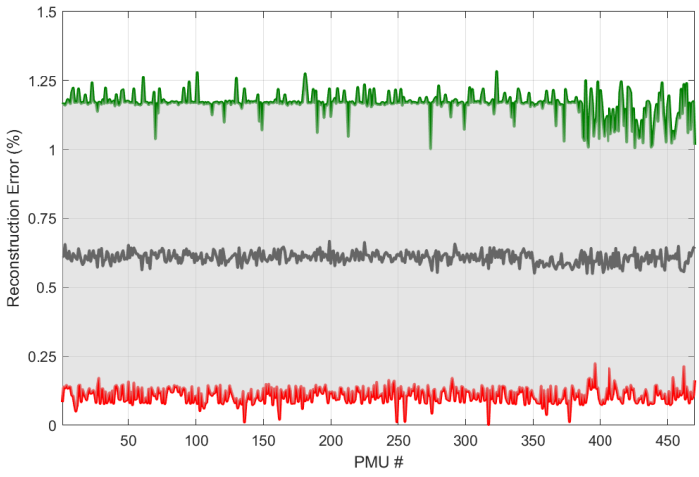
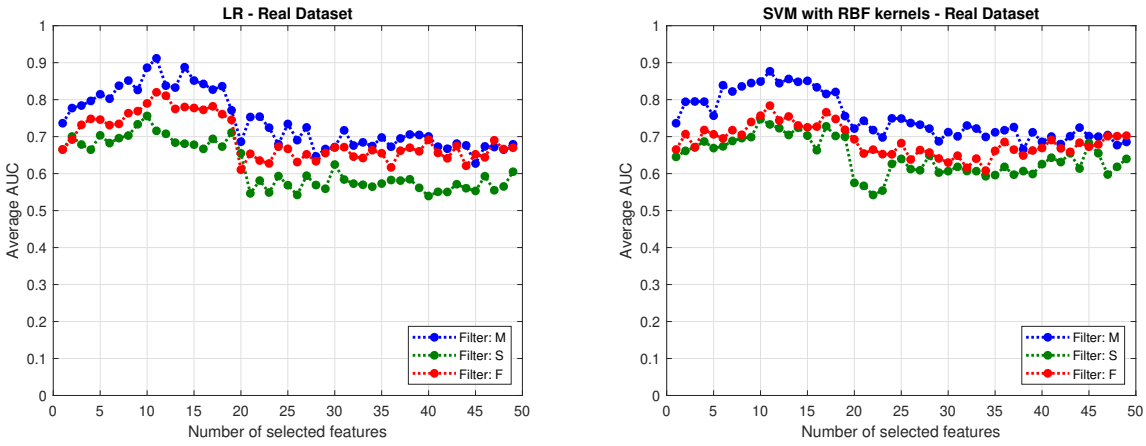


Figure 3.7: Envelope of the reconstruction error of all the PMU measurement streams for 70 events. Red, gray, and green lines represent the minimum, average and maximum reconstruction error, respectively.



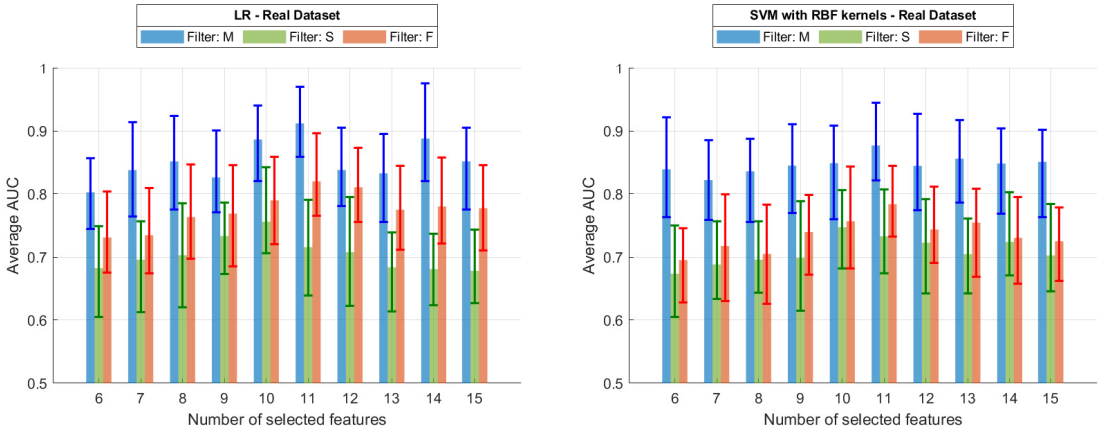
(a) (b)

Figure 3.8: Performance of the classification models (a) LR, and (b) SVM in terms of average AUC over $B_c = 200$ bootstrapped datasets with respect to the number of selected features in the real dataset.

A total number of 42 events are included in the training dataset. The same number of $B_s = 200$ bootstrapped datasets are used for feature selection and the final evaluation of the models. The performance of each classifier in terms of the average AUC scores are shown in Fig. 3.8. Further, the 5th and 95th percentiles of the AUC scores over $B_s = 200$ bootstrapped datasets are shown in Fig. 3.9.

The best performance of the both classifiers are obtained using a subset of 11 features that are selected based on the mutual information. An interesting observation is that in both case studies, the angular frequency and first few residue magnitudes corresponding to the first mode of VPM, VPA and F measurement channels are included in the subset of the selected features obtained from mutual information.

Compared to the synthetic dataset, the performance of the classification models in the real dataset have lower accuracies with wider confidence intervals. Possible reasons for this include (i) the limited number of events (70 labeled events), and (ii) variable system



(a) (b)

Figure 3.9: Performance of the classification models (a) LR, and (b) SVM in terms of the average AUC with respect to the number of selected features in the real dataset. The error bars represent the 5th and 95th percentiles of the AUC scores.

operating conditions as the data was collected over 3 years. Furthermore, in contrast to our simulation results for the synthetic dataset, the learned LR model demonstrates a slightly better performance compared to the learned SVM with RBF kernels model. This is most likely because SVM significantly increases the model complexity and given the small number of samples, is overfitting the training dataset and thus, will not perform as well on the test data [54].

Chapter 4

EVENT IDENTIFICATION FRAMEWORK IN SEMI-SUPERVISED SETTING

In Chapter. 3, we have proposed a novel machine learning framework for event identification based on extracted features obtained from mode decomposition of PMU measurements. Considering the high-dimensionality of the extracted features, we have considered different data-driven filter methods to choose a subset of features. We have investigated the performance of the two classification models (LR and SVM) in identifying the generation loss and line trip events for both synthetic and a proprietary real datasets.

Our simulation results indicate that using mutual information for feature selection results in better performance of the classifiers compared to the other filter methods that we tested, in both real and synthetic datasets. This is due to the fact that mutual information can capture the nonlinear dependencies between the features and the target variable. Our analysis also illustrates that bootstrapping can overcome the limitation of the small number of labeled events. However, when labeled data are limited, a less complex model such as LR can assure better accuracy than more complex models such as SVM. We have also shown that a relatively small number (10–15) of features is typically enough to achieve a good classification performance.

However, the proprietary dataset used in this study suggests that, in practice, a very small number of events are labeled when compared to the total number of events. Given the fact that using expert knowledge for labeling various types of events is expensive and tedious, availability of proper labeled eventful PMU data is still an undergoing challenge in the literature within this context. In this chapter, we aim to explore the potential benefits of incorporating unlabeled samples in enhancing the performance of the event identification task. To this end, we thoroughly investigate and compare the performance of various semi-

supervised algorithms, including: (i) self-training with different base classifiers (i.e., support vector machine with linear kernel (SVML) as well as with radial basis function kernel (SVMR), gradient boosting (GB), decision trees (DT), and k-Nearest Neighbors (kNN)), (ii) transductive support vector machines (TSVM), and (iii) graph-based label spreading (LS) to explore their effectiveness. We chose these classical semi-supervised models for two primary reasons: firstly, the wide array of proposed semi-supervised classification algorithms in the past two decades (see, [42], and references therein) necessitates a comprehensive understanding of which models are most suitable and efficient for event identification; and secondly, they provide a clearer illustration and intuition of the impact of incorporating unlabeled samples compared to more advanced methods. Although there may not be a one-size-fits-all solution, each method has its own advantages and disadvantages, and it is important to evaluate their suitability.

More specifically, to assess the effectiveness of integrating unlabeled data into event identification and to enable a meaningful comparison between various semi-supervised methods, we introduced a three-step pipeline. Firstly, using a semi-supervised model, denoted as F_1, we assign pseudo-labels to unlabeled samples in the training set with a mix of labeled and unlabeled samples. Next, we train a classifier, F_2 on the augmented set of labeled and pseudo-labeled samples. Finally, we evaluate the classifier's performance on previously unseen data in the hold out set. Notably, our experiments consistently illustrate the superior performance of the graph-based LS method compared to other approaches. Even in worst-case scenarios where the initial distribution of labeled and unlabeled samples does not necessarily reflect the true distribution of event classes, the graph-based LS method stands out in robustly and significantly enhancing event identification performance.

4.1 Background

Supervised learning algorithms' performance is influenced by various decisions, including the selection of datasets, their partitioning into training, validation, and testing sets, and tuning of hyperparameters. In semi-supervised learning, additional factors need to be considered, such as determining which data points should be labeled and which should remain unlabeled. In general, semi-supervised approaches utilize both labeled and unlabeled samples, but they are different in the way they incorporate the information from unlabeled samples in the learning process. Furthermore, the performance of the learner can be assessed on either the unlabelled data used for training or a completely separate test set. It is also essential to establish high-quality supervised baselines to assess the value of unlabeled data accurately which is crucial to avoid an unrealistic perspective on learning algorithms' performance. In research, common practice is to obtain unlabeled datasets by removing labels from existing labeled datasets when evaluating the performance of semi-supervised learning algorithms.

4.1.1 Semi-Supervised Learning Techniques for Classification

In the realm of semi-supervised learning, current approaches are commonly categorized as either inductive or transductive methods.

Inductive methods seek to leverage both labeled and unlabeled data points to create a model capable of accurately classifying unseen data points in a test set. can be further divided into three types of wrapper methods, unsupervised pre-processing, and intrinsically semi-supervised methods. The first type of inductive methods, wrapper methods, include training, co-training, and pseudo-labelled boosting methods [42]. Self-training uses one supervised classifier that is iteratively re-trained on its own most confident predictions, while co-training extends this approach to multiple classifiers that are iteratively re-trained on each

other's most confident predictions. Pseudo-labelled boosting methods build a classifier ensemble by constructing individual classifiers sequentially, where each individual classifier is trained on both labeled data and the most confident predictions of the previous classifiers on unlabeled data. The second type of inductive methods, unsupervised pre-processing, utilizes the unlabelled and labeled data in two separate stages. The unsupervised stage comprises the automated extraction or transformation of sample features from the unlabelled data (feature extraction), the unsupervised clustering of the data (cluster-then-label), or the initialization of the parameters of the learning procedure (pre-training). The third type of inductive methods, intrinsically semi-supervised, directly optimizes an objective function with components for labeled and unlabeled samples, and these methods do not rely on any intermediate steps or supervised base learners. Generally, these methods rely either explicitly or implicitly on one of the semi-supervised learning assumptions. For instance, maximum-margin methods rely on the low-density assumption, and most semi-supervised neural networks rely on the smoothness assumption.

In contrast, transductive methods are focused on generating label predictions for unlabeled data points without constructing a general model for classifying new data. Transductive methods typically define a graph over all data points, both labeled and unlabeled, encoding the pairwise similarity of data points with possibly weighted edges. An objective function is then defined and optimized, in order to achieve two goals: (i) for labeled data points, the predicted labels should match the true labels, and (ii)) similar data points, as defined via the similarity graph, should have the same label predictions. These methods encourage consistent predictions for similar data points while taking into account the known labels. These methods are often referred to as graph-based methods, and they are closely related to the inductive manifold-based methods, but with the difference that transductive methods only yield predictions for a given set of unlabeled data points.

4.2 Proposed Framework to Investigate the Impact of Including Unlabeled Data

This section outlines the proposed framework for evaluating the performance of different semi-supervised algorithms across various scenarios of labeled versus unlabeled samples ratios.

4.3 Generation of the Synthetic Eventful Time-series PMU Data

Consider an electric grid composed of set of loads, generators, lines, and buses. We investigate four distinct event classes denoted as $\mathcal{E} \in \{\text{LL, GL, LT, BF}\}$, representing load loss, generation loss, line trip, and bus fault events, respectively. Each PMU provides multiple measurement channels relative to its installation bus. In this study, we focus on voltage magnitude (V_m), corresponding angle (V_a), and frequency (F) channels for clarity, with potential inclusion of other channels. For any channel $c \in \mathcal{C} = \{V_m, V_a, F\}$, let $y_i^c(n) \in \mathbb{R}$ represent the n^{th} measurement, $n = 0, \ldots, N-1$, where the total number of samples is N, from the i^{th} PMU. Assuming PMU sampling period of T_s, we thus collect eventful data for $t_s = NT_s$ seconds. These measurements, for the c^{th} channel, are collated from m PMUs to form a matrix $\mathcal{Y}^{\mathbf{c}} = [\cdots, \mathbf{y}_i^c, \cdots]^T \in \mathbb{R}^{m \times N}$ where $\mathbf{y}_i^c$ is a N-length (column) vector for the i^{th} PMU with entries $y_i^c(n)$, for all n. We use superscript T to denote the tranpose operator. Finally, for each event, we define $\mathcal{M} = [[\mathcal{Y}^{V_m}]^T, [\mathcal{Y}^{V_a}]^T, [\mathcal{Y}^F]^T]^T \in \mathbb{R}^{|C|m \times N}$ by aggregating all the phasor measurements from m PMUs, 3 channels, and for N samples.

Within this setting, we develop a publicly available Python code which leverages PSS®E software Python Application Program Interface (API) to generate synthetic eventful PMU data. To ensure realistic and diverse dataset, we consider the following two steps: Firstly, we linearly adjust all loads within a range of 95% to 105% of their normal loading conditions. Secondly, we add zero-mean random fluctuations, ranging from ±2% of the adjusted loads,

to simulate unpredictable variations observed in real-world power systems. [1] To generate eventful data, for each system component and loading condition considered, we employ the following systematic approach: (i) We begin by applying a new initial loading condition to each load in the system; a power flow analysis for this setting then gives us the initial state conditions for the next step. (ii) We use this initial condition to initiate a t_f-second flat run dynamic simulatsion. (iii) At the t_f second, we introduce a disturbance (i.e., LL, GL, and LT) to a selected component. For BF events, we clear the disturbance after t_{clr} seconds. (iv) Finally, we model the event simulation for additional t_s seconds which then allows us create the data matrix $\mathcal{M}$, representing the PMU measurements associated with the simulated event. We repeat this procedure to generate a desired number of events for each event type.

4.3.1 Generating Event Features Using Modal Analysis

The first step in identifying a system event is to extract a set of delineating features that are likely to contain information regarding the event class. Using the fact that temporal effects in a power system are driven by the interacting dynamics of system components, we use mode decomposition to extract features. More specifically, we assume that each PMU data stream after an event consists of a superposition of a small number of dominant dynamic modes. The resulting features then include frequency and damping ratio of these modes, as well as the residual coefficients indicating the quantity of each mode present. We briefly summarize the mathematical model and refer readers to our recent work [43] for additional details.

We assume that $y_i^c(n)$ after an event consists of a superposition of p common damped

[1] The load change intervals specified in this study can be adjusted depending on the stability of the system under study, ensuring that the system can return to an acceptable state of equilibrium following a disturbance.

sinusoidal modes as

$$y_i^c(n) = \sum_{k=1}^{p} R_{k,i}^c \times (Z_k^c)^n + \epsilon_i^c(n), \quad i \in \{1, \cdots, m\}, \quad c \in \mathcal{C} \tag{4.1}$$

where for any given channel $c \in \mathcal{C}$, $\epsilon_i^c(n)$ represents the noise in the i^{th} PMU measurement and Z_k^c is the k^{th} mode associated with the event. We represent each mode as $Z_k^c = \exp\left(\lambda_k^c T_s\right)$ where $\lambda_k^c = \sigma_k^c \pm j\omega_k^c$ and σ_k^c and ω_k^c are the damping factor and angular frequency of the k^{th} mode, respectively. The residue $R_{k,i}^c$ of the k^{th} mode for the i^{th} PMU is defined by its magnitude $|R_{k,i}^c|$ and angle $\theta_{k,i}^c$. For any given channel c, typically a small subset of the PMUs ($m' < m$) capture the dynamic response of the system after an event. Thus, we only keep the residues of a set of m' PMUs with the largest magnitudes. Note that the m' PMUs are not necessarily the same PMUs for different events (see, [43] for further details).

Using the above procedure, for each channel c, we define a row vector of features, $\mathcal{F}^c$, of length $2p(m' + 1)$ as:

$$\mathcal{F}^c = \left[\{\omega_k^c\}_{k=1}^p, \{\sigma_k^c\}_{k=1}^p, \{|R_{k,i}^c|\}_{k=1}^p, \{\theta_{k,i}^c\}_{k=1}^p\right]_{i \in \{1,\cdots,m'\}} \tag{4.2}$$

which consists of p angular frequencies, p damping factors and the corresponding magnitude and angle of the residues for each of the m' PMUs (with the largest residue magnitudes) and the p modes.

4.3.2 Generating the overall dataset

Let n_D be the total number of events generated over all event classes. Following modal analysis on the PMU measurements as described above, we can represent the i^{th} event, $i \in \mathcal{I}_D = \{1, \ldots, n_D\}$, as a $d = 2p|\mathcal{C}|(m' + 1)$-length vector $x_i^T = [\mathcal{F}^{V_m}, \mathcal{F}^{V_a}, \mathcal{F}^F]$. Considering a positive integer $j \in \{1, \cdots, |\mathcal{E}|\}$ as an event label, we associate a one-hot-encoded vector, $y_i \in \mathbb{R}^{|\mathcal{E}|}$, where $|\mathcal{E}|$ is the total number of event classes, $y_{ij} = 1$ if x_i is labeled as j, and $y_{ij} = 0$, otherwise.

Collating the events and labels from all event classes, we obtain a large data matrix $\mathbf{D} = \{\mathbf{X}_D, \mathbf{Y}_D\}$ where $\mathbf{X}_D = [x_1, ..., x_{n_D}]^T \in \mathbb{R}^{n_D \times d}$ and $\mathbf{Y}_D = [y_1, ..., y_{n_D}]^T \in \mathbb{R}^{n_D \times |\mathcal{E}|}$. Finally, to highlight the possible choices for labeled and unlabeled events from $\mathbf{D}$, we henceforth write $\mathbf{D} = \{(x_i, y_i)\}_{i \in \mathcal{I}_D}$.

4.4 Proposed Framework to Investigate the Impact of Unlabeled Data

To investigate the impact of incorporating unlabeled samples on event identification performance, and to ensure a fair comparison among various inductive (i.e., self-training) and transductive semi-supervised approaches (i.e., TSVM, LS), we utilize the k-fold cross-validation technique. First, we shuffle n_{D} samples in $\mathbf{D}$ and partition the data into n_K equally sized folds. We use $n_K - 1$ folds as a training set, denoted as $\mathbf{D}_T^{(k)} = \{(x_i, y_i)\}_{i \in \mathcal{I}_T^{(k)}}$ with $n_T = \lfloor (n_K - 1) n_D / n_K \rfloor$ samples, and reserve the remaining fold as a validation set, denoted as $\mathbf{D}_V^{(k)} = \{(x_i, y_i)\}_{i \in \mathcal{I}_V^{(k)}}$ with $n_V = n_D - n_T$ samples, and $k = 1, \ldots, n_K$. Here, $\mathcal{I}_T^{(k)}$, and $\mathcal{I}_V^{(k)}$ represents a subset of samples in the training set, and the validation set of the k^{th} fold, respectively, and $\mathcal{I}_T^{(k)} \cup \mathcal{I}_V^{(k)} = \mathcal{I}_D$. We repeat this process K times, with each fold serving as the validation set once.

To further investigate how the distribution of labeled and unlabeled samples affects the performance of various semi-supervised algorithms, we shuffle the samples in the training set for n_Q times and split it into a subset of n_L labeled samples, denoted as $\mathbf{D}_L^{(k,q)} = \{(x_i, y_i)\}_{i \in \mathcal{I}_L^{(k,q)}}$ and a subset of n_U unlabeled samples by ignoring their ground truth labels, denoted as $\mathbf{D}_U^{(k,q)} = \{(x_i, \cdot)\}_{i \in \mathcal{I}_U^{(k,q)}}$ where $\mathcal{I}_L^{(k,q)} \cup \mathcal{I}_U^{(k,q)} = \mathcal{I}_T^{(k)}$, and $q = 1, \cdots, n_Q$. To ensure the inclusion of samples from every class within the labeled subset, we verify the condition $B_{\min} \leq \frac{n_L^c}{n_L} \leq B_{\max}$ where n_L^c is the number of samples corresponding to class c, and $B_{\min}, B_{\max}$ are the specified balance range.

To illustrate the impact of increasing the number of unlabeled samples, we propose the following procedure. Given the number of samples that we want to add at each step, denoted

as Δ_U, we randomly select $n_U^{(s)} = s\Delta_U$ from the pool of n_U samples where $s = 0, \cdots, n_S$, and $n_S = \lfloor n_U/\Delta_U \rfloor + 1$ represents the number of steps. To further investigate the impact of the initial distribution of the labeled samples along with the unlabeled samples, the random selection of the $n_U^{(s)}$ samples at each step $1 \leq s \leq n_S - 1$, is performed n_R times.

Concatenating the labeled training samples, $\mathbf{D}_L^{(k,q)}$, in the k-th fold and q-th split, with a subset of $n_U^{(s)}$ unlabeled samples in the s-th step and r-th random selection ($r \leq n_R$), denoted as $\mathbf{D}_U^{(k,q,s,r)} = \{(x_i, \cdot)\}_{i \in \mathcal{I}_U^{(k,q,s,r)}}$, where $\mathcal{I}_U^{(k,q,s,r)} \subseteq \mathcal{I}_U^{(k,q)}$, we obtain a training dataset with mixed labeled and unlabeled samples, denoted as $\mathbf{D}_M^{(k,q,s,r)} = \{(x_i, y_i)\}_{i \in \mathcal{I}_L^{(k,q)}} \cup \{(x_i, \cdot)\}_{i \in \mathcal{I}_U^{(k,q,s,r)}}$. To account for the semi-supervised learning assumptions, we sort the $n_U^{(s)}$ unlabeled samples in the $\mathcal{I}_U^{(k,q,s,r)}$ based on their proximity to the nearest labeled sample. To improve clarity, for the given k, q, and r, we will modify the superscripts of the training (labeled and unlabeled) and validation samples throughout the remainder of this paper, i.e., $\mathbf{D}_L$, $\mathbf{D}_U^{(s)}$, $\mathbf{D}_M^{(s)}$, and $\mathbf{D}_V$ represent the subsets of n_L labeled, $n_U^{(s)}$ unlabeled, $n_M^{(s)} = n_L + n_U^{(s)}$ mixed, and n_V validation samples, respectively. A visual representation of the outlined approach is depicted in Fig. 4.1.

We can alternatively represent the labeled and unlabeled training samples in matrix format as described below. We define the matrix of event features with labeled samples as $\mathbf{X}_L = [\dots, x_i, \dots]^T$ and the corresponding matrix of labels as $\mathbf{Y}_L = [\dots, y_i, \dots]^T$ where $i \in \mathcal{I}_L^{(k,q)}$. Similarly, for the subset of unlabeled samples, we define $\mathbf{X}_U = [\dots, x_i, \dots]^T$, $i \in \mathcal{I}_U^{(k,q,s,r)}$. For the sake of notation coherency as well as implementation considerations (e.g., learning the classification models), we assign value -1 to the unlabeled samples, i.e., $\mathbf{Y}_U = [-1, \dots, -1]^T \in \mathbb{R}^{n_U^{(s)}}$. Hence, the mixed labeled and unlabeled training set can be expressed as

$$\mathbf{D}_M = \{\mathbf{X}_M, \mathbf{Y}_M\} \tag{4.3}$$

where

$$\begin{aligned} \mathbf{X}_M &= [\mathbf{X}_L{}^T, \mathbf{X}_U{}^T]^T, \\ \mathbf{Y}_M &= [\mathbf{Y}_L{}^T, \mathbf{Y}_U{}^T]^T. \end{aligned} \tag{4.4}$$

Similarly, the validation $\mathbf{D}_V$ in the k^{th} fold can be represented in the matrix format as $\mathbf{D}_V = \{\mathbf{X}_V, \mathbf{Y}_V\}$ where $\mathbf{X}_V = [\dots, x_i, \dots]^T$ and $\mathbf{Y}_V = [\dots, y_i, \dots]^T$, and $i \in \mathcal{I}_V^{(k)}$.

4.5 Semi-supervised Event Identification: Model Learning and Validation

Our procedure to test semi-supervised methods consists of three steps: (i) pseudo-labeling of unlabeled samples in the training set with mixed labeled and unlabeled samples, $\mathbf{D}_M^{(s)}$, (ii) training a classifier using the combined labeled and pseudo-labeled samples, and (iii) evaluating the classifier's performance on the validation set, $\mathbf{D}_V$.

The overview of the proposed approach is shown in Fig. 4.1. Given semi-supervised model $\mathcal{F}_1$ and a classifier $\mathcal{F}_2$, we start with the labeled samples within the k^{th} fold and the q^{th} split of the training set. Using these labeled samples, we perform grid search [56] to obtain hyper-parameters for the models $\mathcal{F}_1$ and $\mathcal{F}_2$, denoted as θ_1^* and θ_2^*. (Note that these hyper-parameters will differ based on k and q.) Subsequently, we use the matrix of event features and the corresponding matrix of labels in the $\mathbf{D}_M^{(s)}$ to assign pseudo-labels on the unlabeled samples using $\mathcal{F}_1$. Utilizing the obtained labeled and pseudo-labeled samples, $\widehat{\mathbf{D}}_M^{(s)}$, we then use model $\mathcal{F}_2 \in \{\text{SVMR, SVML, GB, DT, kNN}\}$ to assign labels to the events in the validation dataset $\mathbf{D}_V$. In the subsequent subsections, we will describe which models we use as $\mathcal{F}_1$ in this procedure.

Self-training

Self-training has proven to be effective in leveraging unlabeled data to improve supervised classifiers [57, 58, 59, 60, 61, 62]. Self-training works by assigning pseudo-labels to unla-

beled samples based on the model's predictions and then training the model iteratively with these pseudo-labeled samples. More specifically, for any given base classifier, we learn a model $\mathcal{F}_1 \in \{\text{SVMR, SVML, GB, DT, kNN}\}$ from the labeled samples in the $\mathbf{D}_M^{(s)}$. Then using the learned model, we predict the labels for each $n_U^{(s)}$ unlabeled samples to obtain the augmented labeled and pseudo-labeled samples, denoted as $\widehat{\mathbf{D}}_M^{(s)}$. Algorithm 1 outlines the steps involved in this procedure. Note that the parameter δ_U in this algorithm specifies the number of unlabeled samples (among the $n_U^{(s)}$ samples) that will be assigned pseudo-labels in each iteration.

Transductive Support Vector Machine (TSVM)

The TSVM approach is a modification of the SVM formulation that addresses the challenge of limited labeled data in classification tasks [63, 64, 42]. The TSVM optimization problem is given by

$$\min_{\mathbf{w},b,\boldsymbol{\eta},\boldsymbol{\zeta},\mathbf{z}} \quad C\left[\sum_{i\in\mathcal{I}_L}\eta_i + \sum_{j\in\mathcal{I}_U}\min(\zeta_j, z_j)\right] + \|\mathbf{w}\|^2 \tag{4.5a}$$

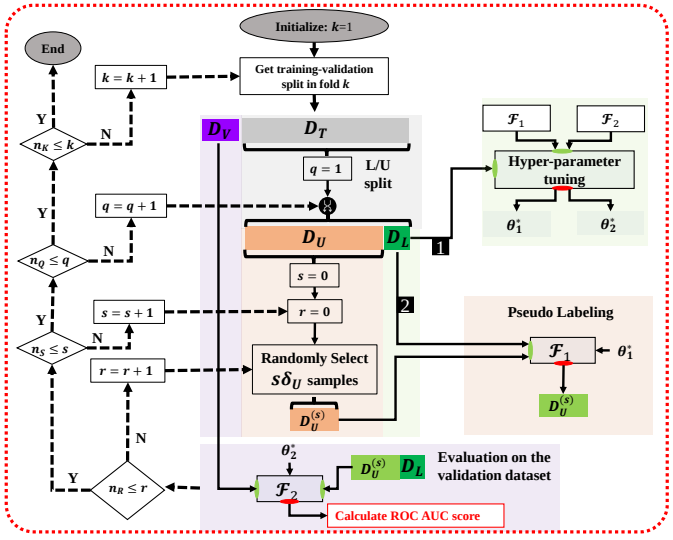


Figure 4.1: Overview of the proposed semi-supervised pipeline

Algorithm 1 Self-Training (for a given k, q, s, and r).

Input: $\mathbf{D}_M^{(s)}$

Output: $\widehat{\mathbf{D}}_M^{(s)}$

Initialize: $[\mathrm{f} : \mathrm{t}] = [1 : \delta_U]$ ▷ *from sample f to sample t*

$\tilde{\mathbf{X}}_L \leftarrow \mathbf{X}_L, \tilde{\mathbf{Y}}_L \leftarrow \mathbf{Y}_L, \tilde{\mathbf{X}}_U \leftarrow \mathbf{X}_U[\mathrm{f} : \mathrm{t}]$

while $\mathrm{t} \leq n_U^{(s)}$ **do**

  $\mathcal{F}_1 : \tilde{\mathbf{Y}}_L \leftarrow \tilde{\mathbf{X}}_L$ ▷ *Learning the model*

  $\widehat{\mathbf{Y}}_U = \mathcal{F}_1(\tilde{\mathbf{X}}_U)$ ▷ *pseudo-labeling*

  $\tilde{\mathbf{X}}_L \leftarrow [\tilde{\mathbf{X}}_L^T, \tilde{\mathbf{X}}_U^T]^T, \quad \tilde{\mathbf{Y}}_L \leftarrow [\tilde{\mathbf{Y}}_L^T, \widehat{\mathbf{Y}}_U^T]^T$ ▷ *Augmentation*

  $f \leftarrow f + \delta_U, \quad t \leftarrow t + \delta_U$

  if $\mathrm{t} > n_U^{(s)}$:

    $\mathrm{t} = n_U^{(s)}$

  $\tilde{\mathbf{X}}_U \leftarrow \mathbf{X}_U[\mathrm{f} : \mathrm{t}]$

$\widehat{\mathbf{Y}}_M \leftarrow \tilde{\mathbf{Y}}_L$

Return: $\widehat{\mathbf{D}}_M^{(s)} = \{\mathbf{X}_M, \widehat{\mathbf{Y}}_M\}$

subject to:

$$y_i(\mathbf{w}^T x_i - b) + \eta_i \geq 1, \quad \eta_i \geq 0, \quad i \in \mathcal{I}_L \tag{4.5b}$$

$$\mathbf{w}^T x_i - b + \zeta_j \geq 1, \quad \zeta_j \geq 0, \quad j \in \mathcal{I}_U \tag{4.5c}$$

$$-(\mathbf{w}^T x_i - b) + z_j \geq 1, \quad z_j \geq 0, \quad j \in \mathcal{I}_U \tag{4.5d}$$

where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ represent the direction of the decision boundary and the bias (or intercept) term, respectively. It introduces two constraints (i.e., (5c), and (5d)) for each sample in the training dataset calculating the misclassification error as if the sample belongs to one class or the other. The objective function aims to find $\mathbf{w}$ and b that, while maximizing the margin and reducing the misclassification error of labeled samples (i.e., $\boldsymbol{\eta}$), minimize the

minimum of these misclassification errors (i.e., $\boldsymbol{\zeta}$ and $\mathbf{z}$). This enables the TSVM to utilize both labeled and unlabeled samples for constructing a precise classification model. Subsequently, it assigns pseudo-labels to the unlabeled samples. For brevity, we refer readers to [63, 64] for more comprehensive details.

Label Spreading (LS)

Label spreading (LS) falls within the category of graph-based semi-supervised (GSSL) models [65]. It involves constructing a graph and inferring labels for unlabeled samples where nodes represent samples and weighted edges reflect similarities. Consider a graph $G_M = (\mathcal{V}_M, \mathcal{W}_M)$ which is constructed over the combined labeled and unlabeled training set. Each sample, $x_i, \forall i \in \mathcal{I}_L \cup \mathcal{I}_U$, can be represented as a node in a graph. For the resulting graph, we define the edge weights matrix as $\mathcal{W}_M \in \mathbb{R}^{n_M^{(s)} \times n_M^{(s)}}$. Defining $D_{ij} = ||x_i - x_j||^2$, the i^{th} row and j^{th} column of $\mathcal{W}_M$, denoted as w_{ij}, can be obtained as $w_{ij} = \exp\left(-D_{ij}/2\sigma^2\right)$ if $i \neq j$, and $w_{ii} = 0$. For such a measure of edge weight, proximal pairs of samples will have larger weights. Building on the classical intuition that proximal samples tend to have the same labels, the LS approach enables propagation of labels from the labeled to unlabeled samples through weighted edges where the weights carry the notion of similarity. In Algorithm 2, we detail the steps of the LS approach based on [66]. The update rule is captured in line 7 in Algorithm 2 wherein the labels for both the labeled and unlabeled samples are updated; in particular, for the labeled samples, such an update includes information from the neighbors (first term) while preserving the initial label (second term). The parameter α determines the weighting between neighbor-derived information and the sample's original label information.

Algorithm 2 Label spreading (for a given k, q, s, and r).

Input: $G = (\mathcal{V}, \mathcal{W}) \leftarrow \mathbf{D}_M^{(s)} = \{\mathbf{X}_M, \mathbf{Y}_M\}$

Output: $\widehat{\mathbf{D}}_M^{(s)}$

Compute: $\mathcal{D}_{ii} = \sum_j w_{ij}, \quad \forall i \in \mathcal{I}_L \cup \mathcal{I}_U$

Compute: $\mathbf{Z} = \mathcal{D}^{-1/2} \mathcal{W}_M \mathcal{D}^{-1/2}$

Initialize: $\begin{bmatrix} \mathbf{Y}_L|_{t=0} \\ \mathbf{Y}_U|_{t=0} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{Y}_L \\ \mathbf{Y}_U \end{bmatrix}$

while $\begin{bmatrix} \mathbf{Y}_L|_t \\ \mathbf{Y}_U|_t \end{bmatrix}$ converges **do** ▷ *Based on some threshold*

$\quad \begin{bmatrix} \mathbf{Y}_L|_{t+1} \\ \mathbf{Y}_U|_{t+1} \end{bmatrix} \leftarrow \alpha \mathbf{Z} \begin{bmatrix} \mathbf{Y}_L|_t \\ \mathbf{Y}_U|_t \end{bmatrix} + (1 - \alpha) \begin{bmatrix} \mathbf{Y}_L|_{t=0} \\ \mathbf{Y}_U|_{t=0} \end{bmatrix}$

$\quad t \leftarrow t + 1$

$\widehat{\mathbf{Y}}_M \leftarrow \begin{bmatrix} \mathbf{Y}_L|_t \\ \mathbf{Y}_U|_t \end{bmatrix}$

Return: $\widehat{\mathbf{D}}_M^{(s)} = \{\mathbf{X}_M, \widehat{\mathbf{Y}}_M\}$

4.6 Simulation Results

In order to investigate the performance of various semi-supervised learning algorithms, we first generate eventful synthetic PMU data, following the procedure described in Section 4.3. Our simulations were carried out on the South-Carolina 500-Bus System [67, 68]. We allow the system to operate normally for $t_f = 1$ second and then we immediately apply a disturbance. We then run the simulation for an additional $t_s = 10$ seconds, and record the resulting eventful measurements at the PMU sampling rate of 30 samples/sec. The t_{clr} for the BF events is 5 cycles (≈ 0.083 seconds). We assume that 95 buses (which are chosen randomly) of the Carolina 500-bus system are equipped with PMU devices and extract features for each such bus from the V_m, V_a, and F channels. We thus collect $N = 300$

samples after the start of an event for each channel. We use the modal analysis methodology as outlined in our recent prior work [43] to extract features using modal analysis. In total, we simulated 1827 events including 500 LL, 500 GL, 500 LT, and 327 BF events. Figure 4.2 illustrates the measurements (i.e., V_m, V_a, and F) recorded from a single PMU after applying LL, GL, LT, and BF events,

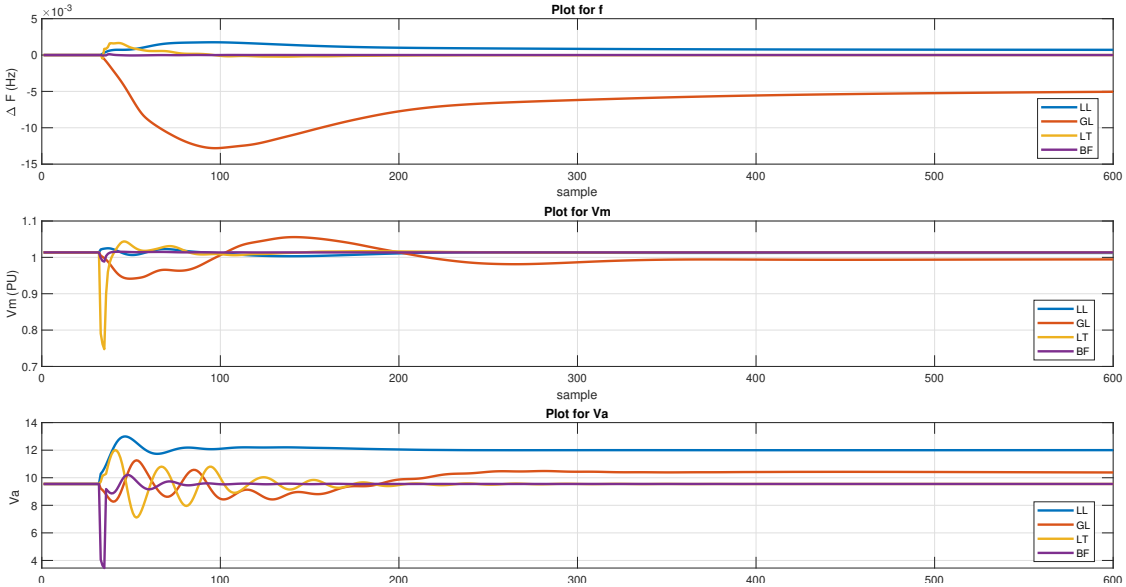


Figure 4.2: PMU measurements

To quantitatively evaluate and compare the performance of different semi-supervised learning algorithms across various scenarios, we employ the area under curve (AUC) of the receiver operator characteristic (ROC). This metric enables the characterization of the accuracy of classification for different discrimination thresholds [40]. The ROC AUC value, which ranges from 0 to 1, provides an estimate of the classifier's ability to classify events. A value of AUC closer to 1 indicates a better classification performance. For a specified set of parameters k, q, s, and r, we evaluate the performance of a given classifier $\mathcal{F}_2$ by assessing its ROC-AUC score in predicting event classes within the hold-out fold. This evaluation is based on the model learned from the augmented labeled and pseudo-labeled samples, which

are obtained using the pseudo-labeling model $\mathcal{F}_1$.

Given that the aim of this study is to provide insight into the robustness of various semi-supervised models, we compare them by evaluating the average, 5^{th} percentile, and 95^{th} percentile of the AUC scores based on the accuracy of the assigned pseudo-labels on the unlabeled samples and assess the impact of incorporating the assigned pseudo-labels on the accuracy of a generalizable model in predicting the labels of validation samples. We use the 5^{th} percentile of the AUC scores as our primary target performance metric for robustness, as it provides a (nearly) worst-case metric across different selections of the initial labeled and unlabeld samples. That is, if a method yields a high 5^{th} percentile performance, then it is likely to lead to accurate results, even if the initial set of labeled and unlabeled samples are unfavorable. Within this setting, to ensure a fair comparison among various inductive and transductive semi-supervised approaches, we consider two distinct approaches:

- **Approach 1 (Inductive semi-supervised setting):**
  $\mathcal{F}_1 \in \{\text{SVMR, SVML, GB, DT, kNN}\}$ represents the base classifier utilized in self-training for pseudo-labeling, and the same type of classifier will be used as $\mathcal{F}_2$.

- **Approach 2 (Transductive semi-supervised setting):**
  $\mathcal{F}_1 \in \{\text{TSVM, LS}\}$ represents a semi-supervised method used for pseudo-labeling, and $\mathcal{F}_2 \in \{\text{SVMR, SVML, GB, DT, kNN}\}$.

In our evaluation process, we take into account $n_K = 10$ folds and $n_Q = 30$ random splits of the training samples into labeled and unlabeled subsets. Other simulation parameters are provided in Table. 4.1. As depicted in Figure 4.3, the comparative performance of diverse classifiers (namely, SVML, SVMR, kNN, DT, and GB) is presented across distinct semi-supervised models (self-training, TSVM, and LS). The outcomes of this analysis highlight that the integration of additional unlabeled samples and the utilization of LS for pseudo-labeling surpasses the outcomes achieved by the self-training and TSVM ap-

proaches. Moreover, the LS algorithm consistently enhances the performance of all classifiers more robustly. The following subsections provides further insight on the performance of each semi-supervised model.

Parameter	**Description**	**Value**
n_D	Total No. of samples	1827
n_K	No. of folds	10
n_T	No. of training samples	1644
n_V	No. of validation samples	183
n_Q	No. of random splits of training samples into labeled and unlabeled samples	20
$(B_{\min}, B_{\max})$	Class balance range in the labeled samples	(0.2, 0.8)
n_L	No. of labeled samples	24
n_U	No. of Unlabeled samples	1620
δ_U	No. of unlabeled samples in each step	100
n_S	Total No. of steps	18
n_R	No. of random selection of $n_U^{(s)}$ samples at each step	10

Table 4.1: Parameters used in the simulations for semi-supervised event identification

4.6.1 Approach 1 - Inductive semi-supervised setting

The simulation results for the 5$^{\text{th}}$ percentile of the AUC scores of the SVML, SVMR, kNN, DT, and GB classifiers in predicting the labels of validation samples are shown in 4.3a. It is clear that using a limited number of labeled samples, results in poor performance for the

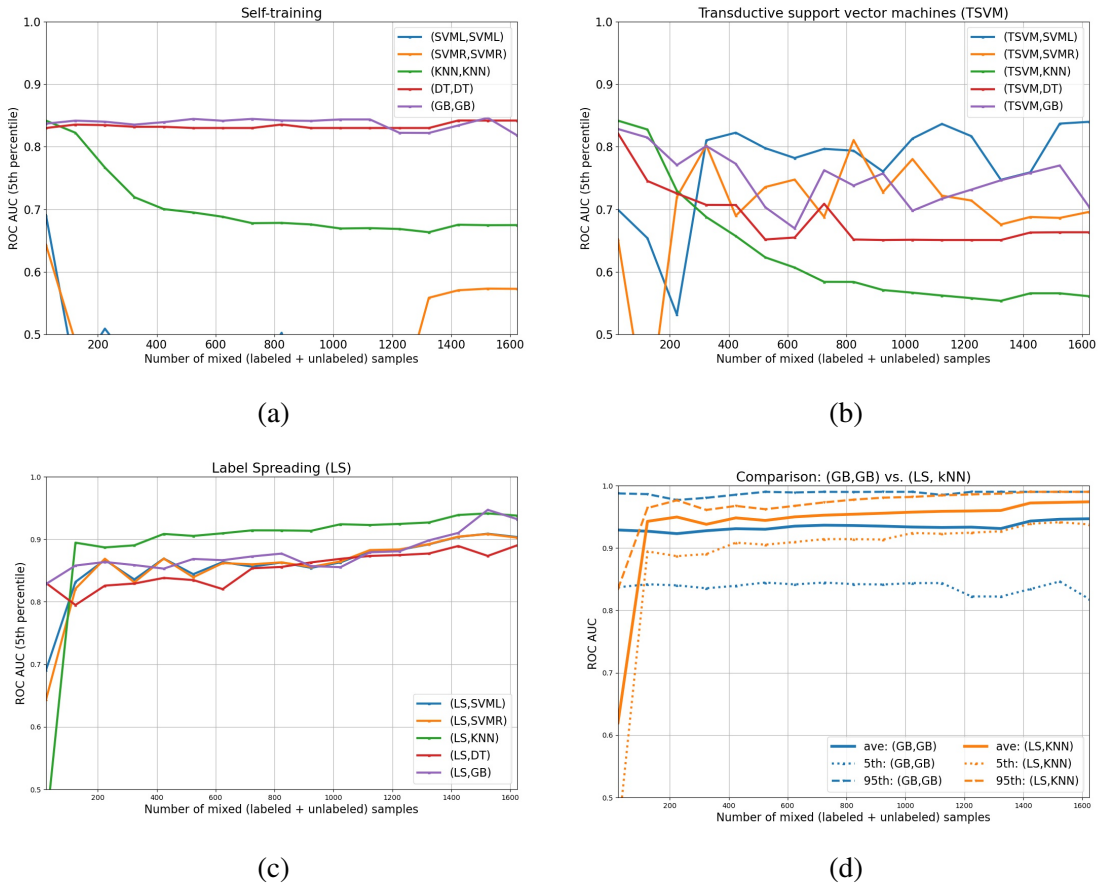


Figure 4.3: The 5th percentile of AUC scores for different classifiers using pseudo-labels obtained from: (a) Self-training method with various base classifiers, (b) TSVM, and (c) LS. (d) Comparison between (GB, GB) and (LS, kNN) in terms of average, 5th, and 95th percentile of AUC scores.

self-training method when utilizing SVMR, SMVL, and kNN base classifiers. Moreover, the utilization of GB and DT as base classifiers does not necessarily lead to an improvement in event identification accuracy. This primarily arises from the disparity between the pseudo-labels and the initial subset of labeled samples. Training with biased and unreliable pseudo-labels can result in the accumulation of errors. In essence, this pseudo-label bias exacerbates particularly for classes that exhibit poorer behavior, such as when the distribu-

tion of labeled samples does not accurately represent the overall distribution of both labeled and unlabeled samples, and is further amplified as self-training continues.

Another noteworthy observation is that self-training employing SVML or SVMR as the classifiers exhibits a high sensitivity to the distribution of both labeled and unlabeled samples. Due to the constraint of having a limited number of labeled samples, these techniques struggle to generate dependable pseudo-label assignments. On the other hand, although self-training with kNN as the base classifier performs better than SVML and SVMR cases, its performance deteriorates as we increase the number of the unlabeled samples. For the self-training with DT and GB base classifiers, it is evident that, although they exhibit more robust performance compared to other types of base classifiers, increasing the number of unlabeled samples does not enhance their performance.

4.6.2 Approach 2 - Transductive semi-supervised setting

The simulation results for the second approach in which TSVM is employed as the semi-supervised method for pseudo-labeling are illustrated in Fig. 4.3b. The weak performance of TSVM could be attributed to the specific characteristics of the dataset and the method's sensitivity to the distribution of labeled and unlabeled samples. If the distribution of these samples is unbalanced or exhibits complex patterns, the TSVM might struggle to accurately capture this distribution. As a result, it could assign inaccurate pseudo-labels. Furthermore, it becomes evident that the integration of pseudo-labels acquired through the TSVM algorithm, although yielding an overall performance advantage for SVML and SVMR when compared to the same models utilizing pseudo-labels from the self-training algorithm involving SVMR and SVML, still exhibits substantial sensitivity. This sensitivity is particularly apparent when assessing the 5% AUC scores, highlighting that the accuracy of assigned pseudo-labels remains highly contingent on the initial distribution of labeled and unlabeled samples. This phenomenon is also observable in the diminishing performance

of the kNN, GB, and DT classifiers, which, surprisingly, deteriorates to a level worse than their utilization as base classifiers within the self-training framework.

On the contrary, as shown in Fig. 4.3c, the results demonstrate that utilizing the augmented labeled and pseudo-labeled samples obtained from LS can significantly enhance the performance of event identification, as compared to the self-training and TSVM approaches. Furthermore, the performance of the event identification task improves with a higher number of unlabeled samples, which is particularly significant since labeled eventful PMU data is often scarce in practice. The principal advantage of the LS method, when compared to self-training and TSVM, primarily arises from its ability to leverage information from both labeled and unlabeled samples, as well as their inherent similarities, during the assignment of pseudo-labels. For some classifiers (specifically GB and DT), we find that LS improves the 5^{th} percentile line with more unlabeled samples, even though the average performance stays roughly unchanged. On the other hand, for the KNN classifier (as shown in Fig. 3d), the average, 5^{th}, and 95^{th} percentile lines all improve with more unlabeled samples. Indeed, LS with KNN seems to be the best overall classifier.

Chapter 5

CONCLUSIONS AND FUTURE WORK

In the first part of the dissertation , we have proposed a novel machine learning framework for event identification based on extracted features obtained from mode decomposition of PMU measurements. Considering the high-dimensionality of the extracted features, we have considered different data-driven filter methods to choose a subset of features. We have investigated the performance of the two classification models (LR and SVM) in identifying the generation loss and line trip events for both synthetic and a proprietary real datasets. Our simulation results indicate that using mutual information for feature selection results in better performance of the classifiers compared to the other filter methods that we tested, in both real and synthetic datasets. This is due to the fact that mutual information can capture the nonlinear dependencies between the features and the target variable. Our analysis also illustrates that bootstrapping can overcome the limitation of the small number of labeled events. However, when labeled data are limited, a less complex model such as LR can assure better accuracy than more complex models such as SVM. We have also shown that a relatively small number (10–15) of features is typically enough to achieve a good classification performance.

Considering the fact that in practice, a very small number of events are labeled when compared to the total number of events, in the second part of the dissertation we proposed a semi-supervised event identification approach to investigate the efficacy of including unlabeled samples on improving the performance of event identification. To evaluate the effectiveness of three classical semi-supervised approaches – self-training, TSVM, and LS methods – we employ a three-step pipeline. In the first step unlabeled samples are assigned pseudo-labels through a semi-supervised method. A classifier is then trained on

the combined set of labeled and pseudo-labeled samples, followed by evaluation on previously unseen data in the hold out set. The simulation results presents critical insights on the performance of various semi-supervised techniques and classifiers for event identification. Self-training methods with SVML or SVMR classifiers demonstrate sensitivity to the distribution of labeled and unlabeled samples, constrained by limited labeled data. While self-training with kNN initially performs well, its efficacy diminishes with more unlabeled samples, offering no guarantee of performance improvement in pseudo labeling or validation label prediction. The study underscores the robust performance of GB and DT classifiers, though augmenting unlabeled samples doesn't enhance their performance. TSVM exhibits an overall advantage for SVML and SVMR compared to self-training, but sensitivity persists across different AUC percentiles due to pseudo-label accuracy dependence on initial sample distribution. This sensitivity extends to kNN, GB, and DT classifiers too. Furthermore, incorporating pseudo-labels from TSVM enhances the robustness of kNN, GB, and DT classifiers, reinforcing the value of TSVM within the pseudo labeling process. Furthermore, the simulation results confirm that the integration of additional unlabeled samples and the utilization of the LS algorithm for pseudo labeling consistently outperform the self-training and TSVM approaches. The LS algorithm notably enhances classifier performance.

Future Work:

This study mainly concentrated on examining basic semi-supervised techniques to confirm our hypothesis that incorporating unlabeled samples has the potential to enhance the precision of our event identification framework. Hence, it is natural to explore more advanced semi-supervised techniques to further improve the performance of the proposed event identification framework. Furthermore, the proposed framework in this study rests upon a critical assumption – that the labeled events and the unlabeled event types perfectly align. In simpler terms, this supposition demands that utilities are capable of detecting and

recording all possible event types without fail. However, maintaining such an assumption in practical scenarios proves to be quite challenging. Unlabeled events often encompass a diverse array of new and unrecorded event types. A promising direction for future exploration involves addressing the challenge of the class distribution mismatch problem. The development of methodologies that proficiently handle unlabeled events containing diverse event types not previously encountered could substantially enhance the performance of the proposed event identification method. Another compelling path for future exploration lies in the investigation of techniques to actively pinpoint informative unlabeled samples for pseudo-labeling, guided by distinct criteria like power system-motivated similarity metrics.

REFERENCES

[1] Y. Zhou, R. Arghandeh, and C. J. Spanos, "Partial Knowledge Data-Driven Event Detection for Power Distribution Networks," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5152–5162, 2017.

[2] M. L. Crow and A. Singh, "The Matrix Pencil for Power System Modal Extraction," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 501–502, 2005.

[3] X. Ding, J. Poon, I. Čelanović, and A. D. Domínguez-García, "Fault Detection and Isolation Filters for Three-Phase AC-DC Power Electronics Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 4, pp. 1038–1051, 2013.

[4] J. E. Tate and T. J. Overbye, "Line Outage Detection Using Phasor Angle Measurements," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1644–1652, 2008.

[5] W. Wang, L. He, P. Markham, H. Qi, Y. Liu, Q. C. Cao, and L. M. Tolbert, "Multiple Event Detection and Recognition Through Sparse Unmixing for High-Resolution Situational Awareness in Power Grid," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1654–1664, 2014.

[6] H. You, V. Vittal, and X. Wang, "Slow Coherency-based Islanding," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 483–491, 2004.

[7] H. Zhu and G. B. Giannakis, "Sparse over complete representations for efficient identification of power line outages," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 2215–2224, 2012.

[8] S. Brahma, R. Kavasseri, H. Cao, N. R. Chaudhuri, T. Alexopoulos, and Y. Cui, "Real-Time Identification of Dynamic Events in Power Systems Using PMU Data, and Potential Applications—Models, Promises, and Challenges," *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 294–301, 2017.

[9] L. Fan, R. Kavasseri, Z. Miao, D. Osborn, and T. Bilke, "Identification of System Wide Disturbances Using Synchronized Phasor Data and Ellipsoid Method," in *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pp. 1–10, 2008.

[10] 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*, pp. 1–8, 2008.

[11] O. P. Dahal, H. Cao, S. Brahma, and R. Kavasseri, "Evaluating Performance of Classifiers for Supervisory Protection using Disturbance Data from Phasor Measurement Units," in *IEEE PES Innovative Smart Grid Technologies, Europe*, pp. 1–6, 2014.

[12] M. He, J. Zhang, and V. Vittal, "Robust Online Dynamic Security Assessment Using Adaptive Ensemble Decision-Tree Learning," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4089–4098, 2013.

[13] D.-I. Kim, "Complementary Feature Extractions for Event Identification in Power Systems Using Multi-Channel Convolutional Neural Network," *Energies*, vol. 14, no. 15, p. 4446, 2021.

[14] M. Al Karim, M. Chenine, K. Zhu, L. Nordstrom, and L. Nordström, "Synchrophasor-Based Data Mining for Power System Fault Analysis," in *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pp. 1–8, 2012.

[15] M. Biswal, Y. Hao, P. Chen, S. Brahma, H. Cao, and P. De Leon, "Signal Features for Classification of Power System Disturbances using PMU Data," in *2016 Power Systems Computation Conference (PSCC)*, pp. 1–7, 2016.

[16] M. Biswal, S. M. Brahma, and H. Cao, "Supervisory Protection and Automated Event Diagnosis Using PMU Data," *IEEE Transactions on Power Delivery*, vol. 31, no. 4, pp. 1855–1863, 2016.

[17] H. Ren, Z. J. Hou, B. Vyakaranam, H. Wang, and P. Etingov, "Power System Event Classification and Localization Using a Convolutional Neural Network," *Frontiers in Energy Research*, vol. 8, 2020.

[18] R. Ma, S. Basumallik, and S. Eftekharnejad, "A PMU-Based Data-Driven Approach for Classifying Power System Events Considering Cyberattacks," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3558–3569, 2020.

[19] J. Shi, B. Foggo, and N. Yu, "Power System Event Identification Based on Deep Neural Network With Information Loading," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5622–5632, 2021.

[20] Z. Li, H. Liu, J. Zhao, T. Bi, and Q. Yang, "Fast Power System Event Identification Using Enhanced LSTM Network With Renewable Energy Integration," *IEEE Transactions on Power Systems*, vol. 36, no. 5, pp. 4492–4502, 2021.

[21] Y. Ge, A. J. Flueck, D.-K. Kim, J.-B. Ahn, J.-D. Lee, and D.-Y. Kwon, "Power System Real-Time Event Detection and Associated Data Archival Reduction Based on Synchrophasors," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 2088–2097, 2015.

[22] E. Perez and J. Barros, "A Proposal for On-Line Detection and Classification of Voltage Events in Power Systems," *IEEE Transactions on Power Delivery*, vol. 23, no. 4, pp. 2132–2138, 2008.

[23] W. Li, M. Wang, and J. H. Chow, "Real-time Event Identification Through Low-Dimensional Subspace Characterization of High-Dimensional Synchrophasor Data," *IEEE Transactions on Power Systems*, vol. 33, no. 5, pp. 4937–4947, 2018.

[24] K. Venugopal, P. Madhusudan, and A. Amrutha, "Artificial Neural Network based Fault Prediction Framework for Transformers in Power Systems," in *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 520–523, 2017.

[25] S. Zhang, Y. Wang, M. Liu, and Z. Bao, "Data-Based Line Trip Fault Prediction in Power Systems Using LSTM Networks and SVM," *IEEE Access*, vol. 6, pp. 7675–7686, 2018.

[26] H. Li, Y. Weng, E. Farantatos, and M. Patel, "An Unsupervised Learning Framework for Event Detection, Type Identification and Localization Using PMUs Without Any Historical Labels," in *2019 IEEE Power Energy Society General Meeting (PESGM)*, pp. 1–5, 2019.

[27] M. Zhou, Y. Wang, A. K. Srivastava, Y. Wu, and P. Banerjee, "Ensemble-Based Algorithm for Synchrophasor Data Anomaly Detection," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2979–2988, 2019.

[28] J. Cordova, C. Soto, M. Gilanifar, Y. Zhou, A. Srivastava, and R. Arghandeh, "Shape Preserving Incremental Learning for Power Systems Fault Detection," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 85–90, 2019.

[29] L. Xie, Y. Chen, and P. R. Kumar, "Dimensionality Reduction of Synchrophasor Data for Early Event Detection: Linearized Analysis," *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 2784–2794, 2014.

[30] H. Li, Y. Weng, E. Farantatos, and M. Patel, "A Hybrid Machine Learning Framework for Enhancing PMU-based Event Identification with Limited Labels," in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, pp. 1–8, 2019.

[31] Y.-F. Li and Z.-H. Zhou, "Towards making unlabeled data never hurt," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 175–188, 2015.

[32] R. Razavi-Far, E. Hallaji, M. Farajzadeh-Zanjani, and M. Saif, "A semi-supervised diagnostic framework based on the surface estimation of faulty distributions," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1277–1286, 2019.

[33] F. Yang, Z. Ling, Y. Zhang, X. He, Q. Ai, and R. C. Qiu, "Event detection, localization, and classification based on semi-supervised learning in power grids," *IEEE Transactions on Power Systems*, pp. 1–15, 2022.

[34] H. Li, Y. Weng, E. Farantatos, and M. Patel, "A hybrid machine learning framework for enhancing pmu-based event identification with limited labels," in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, pp. 1–8, 2019.

[35] Y. Yuan, Y. Wang, and Z. Wang, "A data-driven framework for power system event type identification via safe semi-supervised techniques," *IEEE Transactions on Power Systems*, pp. 1–12, 2023.

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

[37] Y. Zhou, R. Arghandeh, I. Konstantakopoulos, S. Abdullah, and C. J. Spanos, "Data-driven event detection with partial knowledge: A hidden structure semi-supervised learning method," in *2016 American Control Conference (ACC)*, pp. 5962–5968, 2016.

[38] K. Sheshyekani, G. Fallahi, M. Hamzeh, and M. Kheradmandi, "A General Noise-Resilient Technique Based on the Matrix Pencil Method for the Assessment of Harmonics and Interharmonics in Power Systems," *IEEE Transactions on Power Delivery*, vol. 32, no. 5, pp. 2179–2188, 2017.

[39] D. Trudnowski, J. Johnson, and J. Hauer, "Making Prony Analysis More Accurate using Multiple Signals," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 226–231, 1999.

[40] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 56–70, 2020.

[41] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid Structural Characteristics as Validation Criteria for Synthetic Networks," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3258–3265, 2017.

[42] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, pp. 373 – 440, 2019.

[43] N. Taghipourbazargani, G. Dasarathy, L. Sankar, and O. Kosut, "A machine learning framework for event identification via modal analysis of pmu data," *IEEE Transactions on Power Systems*, pp. 1–12, 2022.

[44] A. R. Borden and B. C. Lesieutre, "Variable Projection Method for Power System Modal Identification," *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 2613–2620, 2014.

[45] L. L. Grant and M. L. Crow, "Comparison of Matrix Pencil and Prony Methods for Power System Modal Analysis of Noisy Signals," in *2011 North American Power Symposium*, pp. 1–7, 2011.

[46] W. Trinh and T. Overbye, "Comparison of Dynamic Mode Decompositon and Iterative Matrix Pencil Method for Power System Modal Analysis," in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, pp. 1–6, 2019.

[47] W. Trinh, K. Shetye, I. Idehen, and T. Overbye, "Iterative Matrix Pencil Method for Power System Modal Analysis," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[48] S. Zhang, Y. Hao, M. Wang, and J. H. Chow, "Multichannel Hankel Matrix Completion Through Nonconvex Optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 617–632, 2018.

[49] T. Becejac and T. Overbye, "Impact of PMU Data Errors on Modal Extraction Using Matrix Pencil Method," in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, pp. 1–8, 2019.

[50] T. K. Sarkar, F. Hu, Y. Hua, and M. Wicks, "A Real-Time Signal Processing Technique for Approximating a Function by a Sum of Complex Exponentials Utilizing the Matrix-Pencil Approach," *Digital Signal Processing*, vol. 4, no. 2, pp. 127–140, 1994.

[51] M. Sheikhan, M. Bejani, and D. Gharavian, "Modular Neural-SVM Scheme for Speech Emotion Recognitionusing ANOVA feature Selection Method," *Neural Computing and Applications*, vol. 23, no. 1, pp. 215–227, 2013.

[52] J. Fan and J. Lv, "Sure Independence Screening for Ultrahigh Dimensional Feature Space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 5, pp. 849–911, 2008.

[53] B. C. Ross, "Mutual Information Between Discrete and Continuous Data Sets," *PloS one*, vol. 9, no. 2, p. e87357, 2014.

[54] J. Friedman, T. Hastie, R. Tibshirani, *et al.*, *The Elements of Statistical Learning*, vol. 1. Springer series in statistics New York, 2001.

[55] "Activsg2000: 2000-bus synthetic grid on footprint of texas,"

[56] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization.," *Journal of machine learning research*, vol. 13, no. 2, 2012.

[57] B. Chen, J. Jiang, X. Wang, J. Wang, and M. Long, "Debiased pseudo labeling in self-training," *arXiv preprint arXiv:2202.07136*, 2022.

[58] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," *ACL*, 1995.

[59] R. Rosenfeld, "A maximum entropy approach to unsupervised word sense disambiguation," *ACL*, 1996.

[60] D. McClosky, E. Charniak, and M. Johnson, "Effective self-training for parsing," in *HLT-NAACL*, 2006.

[61] D.-H. Lee, H. S. Xu, X. Zhang, and N. Kwak, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," *ICML*, 2013.

[62] S. C. Fralick, "Learning to recognize patterns without a teacher," *IEEE Trans. Inf. Theory*, vol. 13, pp. 57–64, 1967.

[63] F. Gieseke, A. Airola, T. Pahikkala, and O. Kramer, "Sparse quasi-newton optimization for semi-supervised support vector machines.," pp. 45–54, 2012.

[64] T. Joachims *et al.*, "Transductive inference for text classification using support vector machines," in *Icml*, vol. 99, pp. 200–209, 1999.

[65] Z. Song, X. Yang, Z. Xu, and I. King, "Graph-based semi-supervised learning: A comprehensive review," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2022.

[66] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in neural information processing systems*, vol. 16, 2003.

[67] T. Xu, A. B. Birchfield, and T. J. Overbye, "Modeling, tuning, and validating system dynamics in synthetic electric grids," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6501–6509, 2018.

[68] T. Xu, A. B. Birchfield, K. S. Shetye, and T. J. Overbye, "Creation of synthetic electric grid models for transient stability studies," in *The 10th Bulk Power Systems Dynamics and Control Symposium (IREP 2017)*, pp. 1–6, 2017.