Machine Learning Approaches for Real-time Integration of Synchrophasor Data

Final Project Report

S-87

Power Systems Engineering Research Center
Empowering Minds to Engineer
the Future Electric Energy System

Machine Learning Approaches for Real-time Integration of Synchrophasor Data

Final Project Report

Project Team
Lalitha Sankar, Project Leader
Anamitra Pal
Arizona State University

Le Xie
Texas A&M University

Graduate Students
Harish Chandrasekaran
Antos Cheeramban Varghese
Pooja Gupta
Andrea Pinceti
Reetam Sen Biswas
Nima Taghipourbazargani
Arizona State University

Xiangtian Zheng
Texas A&M University

PSERC Publication 21-09

September 2021

For information about this project, contact:

Lalitha Sankar
Arizona State University
School of Electrical, Computer, and Energy Engineering
Goldwater Center, R# 436,
551 E Tyler Mall,
Tempe, AZ 85281
Phone: 480-965-4953
Email: lalithasankar@asu.edu

Power Systems Engineering Research Center

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

For additional information, contact:

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

Notice Concerning Copyright Material

Acknowledgements

At ASU, we wish to thank Prof. Oliver Kosut and Dr. Rajasekhar Anguluri (PI Sankar's postdoc shared with Kosut) for their feedback on both the research carried out and report writing.

We would also thank the following people for the helpful suggestions they provided by engaging in insightful conversations with us: Harvey Scribner (SPP, hscribner@spp.org), Evangelos Farantatos (EPRI, efarantatos@epri.com), and all our industry advisers who have always participated in our IAB meetings. Finally, we would like to acknowledge all the Industry Advisory Board members who supported us in this project.

Industry Advisory Board Members

Harvey Scribner (SPP)	Evangelos Farantatos (EPRI)
Phil Hart (GE)	Matthew Rhodes (SRP)
Liwei Hao (GE)	Alan Engelmann (ComEd)
Xiaohu Zhang (GEIRINA-USA)	Mahendra Patel (EPRI)
George Stefopoulos (NYPA)	Qiang Zhang, (ISO-NE)
Ruisheng Diao (GEIRINA-USA)	Yingchen Zhang (NREL)
Benjamin Kroposki (NREL)	

Executive Summary

Recently, the availability of fast time scale synchrophasor data has enabled both researchers and practitioners to tackle challenging and interesting problems related to monitoring, control, and security of power systems. This objective would not have been possible without the significant advancements in machine learning (ML)---specifically, deep learning---in terms of efficiency, reliability, and scalability. The overall goal of this project is to efficiently pre-process large volumes of synchrophasor data and then train suitable ML algorithms to (i) generate synthetic load and voltage data for downstream applications and (ii) detect islanding and events in a power system. Below we outline the key contributions of these two tasks (or parts) and we provide full details in later sections.

Part 1: Generation of synthetic load and voltage data using the state-of-the-art deep learning algorithms

Part 1 of the project considers two problems. The ASU team focused on generating synthetic load data for steady state applications, which can ultimately be used by the industries and practitioners to predict loads and unanticipated attacks. In particular, a framework for the generation of synthetic time-series transmission-level load data is presented. Conditional generative adversarial networks are used to learn the patterns of a real dataset of hourly-sampled week-long load profiles and generate unique synthetic profiles on demand, based on the season and type of load required. Extensive testing of the generative model is performed to verify that the synthetic data fully captures the characteristics of real loads and that it can be used for downstream power system applications, a few of them mentioned above.

The TAMU team focused on generating both synthetic load and voltage data to assess the dynamic (specifically, transient) stability of the power system. In particular, the team creates a massive synthetic eventful data from the given limited real data using the proposed two-stage method that leverages Generative Adversarial Nets (GAN) and Neural Ordinary Differential Equations (NeuralODEs). The key to the success of the proposed idea is to make sure that the synthetic data are realistic and diverse rather than a simple clone of the real data, which is achieved by the proposed PMU data creation algorithm. Finally, the proposed conditional learning framework can be leveraged for the generation of other datasets highlighting different characteristics, such as the level of penetration of renewables or electric vehicle charging.

Part 2: Real time detection, identification, and prediction of oscillatory events and contingencies using synchrophasor data: A machine learning approach

Part 2 of the project considers two problems. PI Pal's group at ASU focused on developing a graph-theoretic approach to detect if a contingency can create a saturated cut set. These potential cutsets, which basically serve as an alarm, can alert operators which geographical region of the power system may be disconnected from the main grid. Identification of such cutsets is important particularly in the presence of extreme weather events where successive contingencies can take place leading to uncontrolled islanding of the power system. To this end, the group proposed a feasibility test algorithm whose inputs are generated by deep neural networks and support vector regression methods that are extensively trained using synchronized bus power injections.

PI Sankar's group at ASU developed a physics-based ML approach for online event (e.g., generation loss and line trips) identification. The proposed approach takes advantage of knowledge of the system dynamics by characterizing event signatures based on their modal information that can be directly extracted from phasor measurement unit (PMU) measurements, and subsequently applying ML techniques to produce a robust classifier from limited but feature-rich training data. However, including all possible measurements channels at each PMU allows exploiting diverse features but also requires learning classification algorithms over a high-dimensional space. To this end, various feature selection methods are implemented in order to choose the best subset of features. Using the obtained subset of features, we investigate the performance of two well-known classification algorithms, namely, logistic regression and support vector machines (SVM) to identify generation loss and line trip events in two different datasets. The first dataset is obtained from simulated generation loss and line trip events in Texas 2000-bus synthetic grid using PSSE software and the second is a proprietary dataset with labeled generation losses and line trip events obtained from a utility in the US, involving measurements from nearly 500 PMUs. Our simulation results indicate that the proposed framework is promising for identifying the two types of events in both real and simulated datasets.

Project Publications:

[1] A. Pinceti, L. Sankar and O. Kosut, "Generation of Synthetic Multi-Resolution Time Series Load Data" 2021, arXiv:2107.03547 (Submitted to IEEE Transactions on Power Systems).

[2] A. Pinceti, L. Sankar and O. Kosut, "Synthetic Time-Series Load Data Via Conditional Generative Adversarial Networks", 2021 IEEE Power and Energy Society General Meeting, August 2021

[3] N. T. Bazargani, L. Sankar and O. Kosut, "Event Identification Framework Based on Modal Analysis of Phasor Measurement Unit Data" 2021, under preparation.

[4] X. Zheng, B. Wang, D. Kalathil, and L. Xie. "Generative Adversarial Networks-Based Synthetic PMU Data Creation for Improved Event Classification." IEEE Open Access Journal of Power and Energy 8 (2021): 68-76.

[5] X. Zheng, A. Pinceti, L. Sankar and L. Xie, "Synthetic Eventful Synchrophasor Data Over Time-varying Load Conditions: A Generative Adversarial Netts-based Approach," 2021, under preparation.

[6] R. S. Biswas, A. Pal, T. Werho, and V. Vittal, "Fast identification of saturated cut-sets using graph search techniques," in *Proc. IEEE Power Eng. Soc. General Meeting*, Montreal, Canada, pp. 1-5, 2-6 Aug. 2020.

[7] R. S. Biswas, A. Pal, T. Werho, and V. Vittal, "A graph theoretic approach to power system vulnerability identification," *IEEE Trans. Power Syst*., vol. 36, no. 2, pp. 923-935, Mar. 2021.

[8] R. S. Biswas, A. Pal, T. Werho, and V. Vittal, "Mitigation of saturated cut-sets during multiple outages to enhance power system security," accepted for publication in *IEEE Transactions on Power Systems*.

Student Theses:

[1] Andrea Pinceti. Machine Learning for the Analysis of Power System Loads: Cyber-Attack Detection and Generation of Synthetic Datasets. Ph.D. Dissertation, Arizona State University, Tempe, AZ, 2021.

[2] Reetam Sen Biswas. Power System Security Enhancement for Real-Time Operations during Multiple Outages using Network Science, Ph.D. Dissertation, Arizona State University, Tempe, AZ, 2021.

Part I

Generation of synthetic load and voltage data using the state-of-the-art deep learning algorithms

Lalitha Sankar
Andrea Pinceti, Graduate student

Arizona State University

Le Xie
Xiangtian Zheng, Graduate Student

Texas A&M University

For information about this project, contact

Lalitha Sankar
Arizona State University
School of Electrical, Computer, and Energy Engineering
Goldwater Center, R# 436,
551 E Tyler Mall,
Tempe, AZ 85281
Phone: 480-965-4953
Email: lalithasankar@asu.edu

Power Systems Engineering Research Center

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

For additional information, contact:

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

Notice Concerning Copyright Material

Table of Contents

List of Figures

List of Tables

1. Generation of synthetic time series load data for steady state applications

1.1 Introduction

In recent years, the field of machine learning (ML) has matured to the point where it can provide real value to power system operations; for this reason, a large portion of research work focuses on applying ML to power system applications. Within this new paradigm, the availability of large amounts of real data is crucial. Unfortunately, while power system models of all kinds are readily available, data is a much scarcer resource and the research community must rely on the very few and limited datasets that are publicly available.

The goal of our project is to develop a mechanism for the generation of synthetic time-series transmission-level load data. Leveraging a proprietary dataset of high-resolution measurements from hundreds of phasor measurement units (PMUs) across many years of operation, we can model the behavior of real system loads and subsequently generate realistic-looking data on demand. The focus on bus-level load data is motivated by the fact that loads are one of the main external drivers of power system behaviors; loads depend on phenomena outside of the power system itself (consumer behaviors, weather, etc.). Thus, realistic load profiles can be used as an input to existing power system programs to accurately determine electric quantities such as voltages and currents via dynamic simulation.

1.2 Generative Adversarial Network

1.2.1 Basic GAN

Generative adversarial networks [7] are a novel ML framework in which a generative model (or generator) is trained by making it compete against a discriminator. The goal of the generator G is to capture the distribution of the real data p_r, while the discriminator D is trained to distinguish the real data from the synthetic data produced by the generator. The generator is trained to learn a mapping $G(\boldsymbol{z};\Theta_g)$ from a known noise distribution p_z to p_g, where G is a differentiable function represented by a multilayer neural network with parameters Θ_g and $\boldsymbol{z}$ is a noise vector sampled from p_z. Given a data sample $\boldsymbol{x}$, the discriminator determines the probability $D(\boldsymbol{x},\Theta_d)$ that the sample came from the real data distribution p_r rather than from the generator p_g. The training of D and G is represented by a two-player minimax game, where the optimal solution is obtained when the discriminator assigns a probability of 0.5 to all samples, meaning that it cannot distinguish between real and generated data.

1.2.2 Conditional GAN

Conditional generative adversarial networks (cGAN) [8] are an improvement on the basic GAN framework which allow for a more targeted generation of synthetic data. The conditioning is performed by labelling the real data and then providing this label $\boldsymbol{y}$ as a further input to both the generator and the discriminator. By doing this, the generator learns the conditional distribution p_g over $\boldsymbol{x}|\boldsymbol{y}$ and the generation process can be guided by requesting synthetic data belonging to a specific class. The final structure of a generic cGAN can be seen in the figure below.

1.2.3 Dataset Description

The foundation of this project is a large dataset of real PMU data obtained from a utility in the USA. The data spans two years (2017 and 2018) and about 500 PMUs installed at the transmission level. Based on the system topology and the location of the measurement devices, we identified 12 load buses whose lines are entirely monitored by PMUs. The net injection at these buses represents the load demand and this allowed us to compute the active and reactive power of the 12 loads with a resolution of 30 samples per second. As discussed in the introduction, the focus of this work is the generation of week-long profiles at a resolution of 1 sample per hour for a total of 168 hours. The raw, PMU-speed load data is then down sampled by computing the hourly load average and broken into weeks. When combining all weekly profiles from all 12 loads, the final dataset is a 1158x168 matrix. The total number of profiles (1158) is lower than the theoretical maximum (2 years x 12 loads x 52 weeks = 1248) because profiles containing any data dropout have been discarded. Each week-long profile is normalized by dividing it by the average load over the week; the entire dataset is further normalized between 0 and 1 for the training of the cGAN.

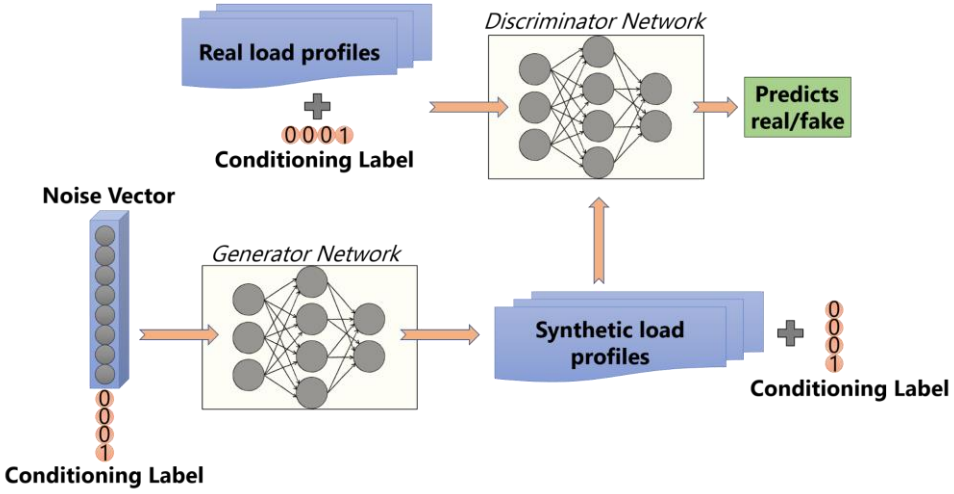

Figure 1.1 Structure of a conditional GAN.

1.2.4 Load Characteristics

Different factors influence the way system loads behave over time. To appropriately generate realistic synthetic load profiles for a given application, these elements need to be captured and modeled by the GAN. When looking at the week-long time-series data described in the previous section, two main driving factors can be identified: time of the year, and type of load.

The differences between load profiles due to seasonal changes in energy consumption can be easily visualized. The figure above shows four week-long profiles for a load, across the four different seasons. In winter and fall, the load pattern presents two daily peaks, one in the morning and one in the afternoon. During spring and especially summer, the load is more regular, with a large spike during the day and dips at night. This type of behavior can be observed in a more or less pronounced manner across all the loads in our dataset. For this reason, the season of the year to which a profile belongs is an important indicator (label) of the expected profile.

At the transmission level, each load represents the aggregate demand of one or more distribution feeders. Thus, the behavior of a load is given by the sum of all the customers at the distribution level that it serves. Because of this, the second main factor that determines the temporal profile of a load is its composition in terms of residential, commercial, and industrial portions since each of these types of loads tend to have very distinctive patterns. In our dataset, we have observed two classes of loads with very distinct behaviors: loads that are mainly residential and/or commercial and loads that are mainly industrial. The figure below shows some selected examples: on the top and bottom left are mainly residential loads from winter and summer respectively, while on the right are winter and summer profiles of a mainly industrial load. As we can see, loads that are mainly residential have very regular and predictable patterns, whereas industrial loads do not necessarily follow recognizable daily patterns. We used a k-means clustering algorithm to label each load as mainly residential or mainly industrial. When using 3 clusters, two main groups of loads are identified, each containing five and six loads, while one single load is grouped separately. We observed that the loads where the top two factors in terms of percentage composition are residential and commercial are clustered as one, while the loads in which the industrial component is first or second are grouped as another cluster. Thus, for training purposes, six loads are labeled as mainly residential and six as mainly industrial.

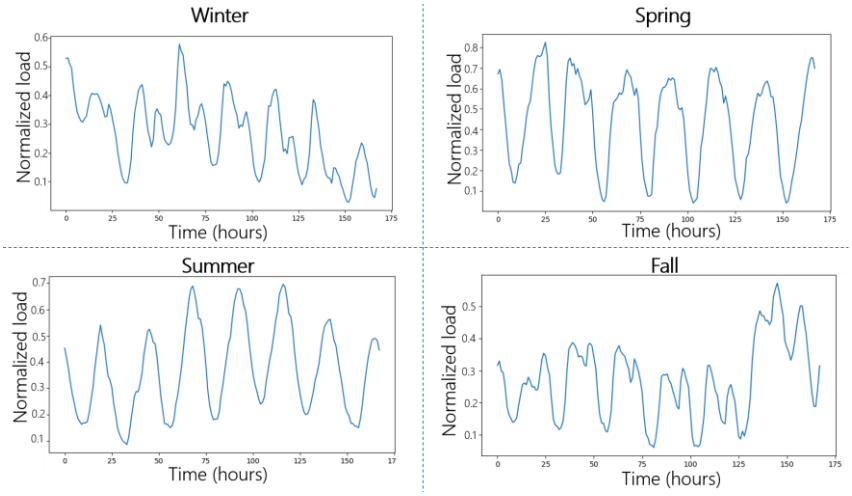

Figure 1.2 Examples of real load profiles; different seasons present different patterns.

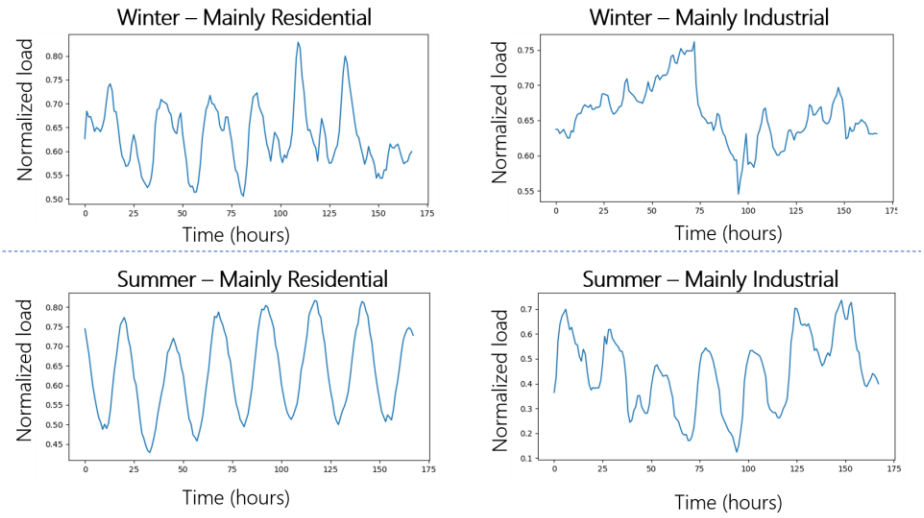

Figure 1.3 Examples of real load profiles. Top row: mainly residential winter load (left) and mainly industrial winter load (right). Bottom row: mainly residential summer load (left) and mainly industrial summer load (right).

1.3 cGAN Model

In this section, we will describe the implementation of the cGAN and its training process. Convolutional neural networks (CNNs) are chosen for the discriminator and the generator for their ability to learn multiple spectral properties of the data. While similar in size and complexity, the two models present some differences.

The discriminator receives two inputs: first, the raw time-series is processed by two convolutional layers, then the flattened output is concatenated to its label and fed to three fully connected layers. The output of the discriminator is a scalar indicating if a profile is real (1) or fake (0). In the generator, the two inputs are the load label and a 25-dimensional Gaussian noise vector. These are concatenated and fed to three fully connected layers and the output is up sampled via three transposed convolution layers. The final output is a synthetic load profile whose characteristics match the input label. The Architecture of the cGAN is illustrated below

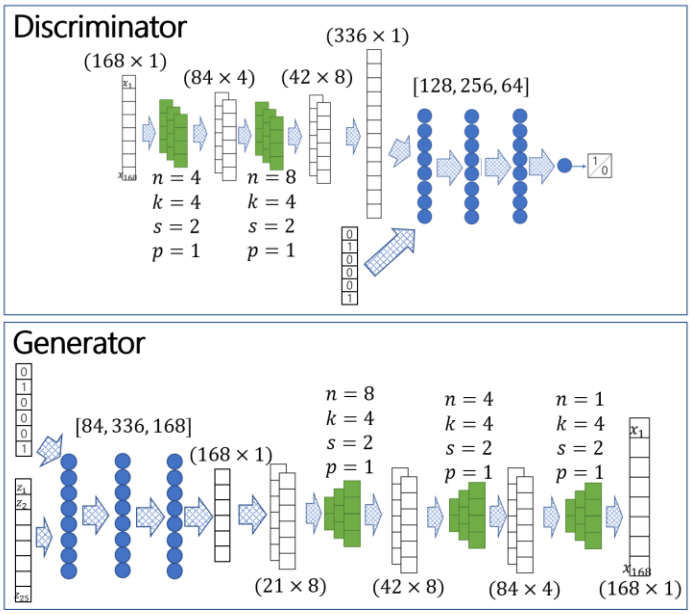

Figure 1.4 Structure of the cGAN used for the generation of week-long profiles.

As we have seen in the previous section, two characteristics of loads are used as labels for the conditional GAN: the season and whether a load is mainly residential or industrial. The factors are represented as one-hot encoded vectors, i.e., the seasons are represented via the following four labels: (1 0 0 0) for winter, (0 1 0 0) for spring, (0 0 1 0) for summer, and (0 0 0 1) for fall. Similarly, the load type is encoded as: (0 1) for mainly residential and (1 0) for mainly industrial.

The training process is performed by iteratively training the discriminator to distinguish between real and generated data and the generator to create realistic-looking profiles. It is to be noted that the discriminator is trained twice at every iteration to give it an advantage against the generator; this forces the generator to produce better samples. The figure below shows how the training process progresses as the epochs proceed. The three curves represent the average discriminator prediction at each epoch for three sample datasets: real data used during training (blue), real data never used during training (validation data, green), and fake data created by the generator (orange). We can see that at the beginning the discriminator easily distinguishes between real and fake data, assigning high values to both real datasets and low value to the generated data. As training progresses, the generator improves, and the discriminator is unable to differentiate between the two data sources. At around 3000 epochs, the training converges: the discriminator assigns very similar values to all three datasets. It is interesting to notice that some overfitting is happening (the blue curve reaches 0.53) but it is not very significant. More importantly, the discriminator assigns the same values to both the generated data and the validation data; this means the output of the generator matches the real data.

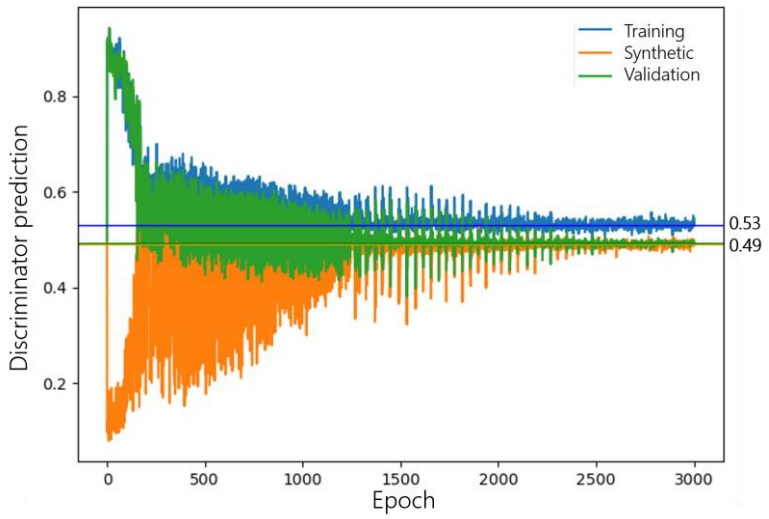

Figure 1.5 Training progress of the cGAN based on the predictions of the discriminator at each epoch. The blue curve shows the average prediction over a batch of real training profiles. The green curve represents real validation data and the orange one predictions on generated data.

1.4 Data Generation

Once the training process is terminated, the generator can be used to create any number of synthetic profiles. Based on the required data type, the user only needs to define the appropriate label and generate a noise vector according to the predetermined distribution; feeding these to the generator will result in a synthetic load profile. As an example, in the figure below, two synthetic summer profiles (right) are compared to two randomly selected real profiles (left) of the same label. The blue profiles (top) correspond to a mainly residential load and the green plots (bottom) to a mainly industrial load. It is important to notice that while the synthetic profiles present all the same characteristics as real data, they do not simply repeat real profiles.

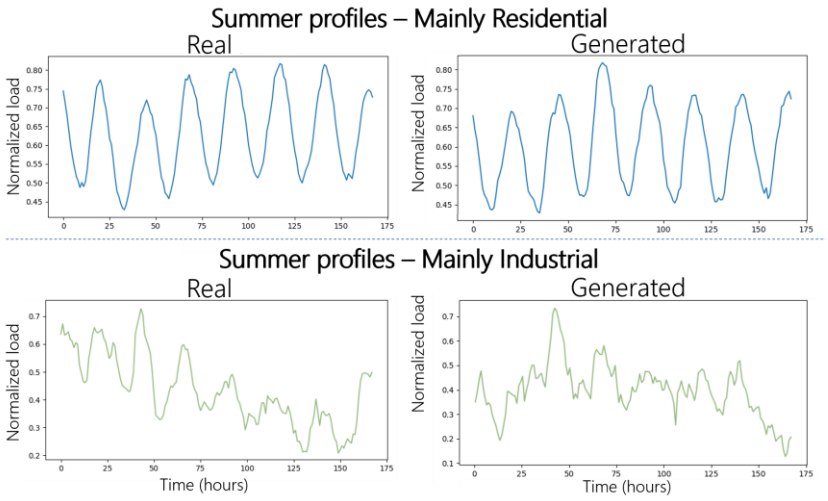

Figure 1.6 Comparison between some real summer profiles (left) and generated profiles (right). Top: mainly residential load. Bottom: mainly industrial load

1.4.1 Evaluation of Synthetic Data

While visual inspection does not yield clear differences between real and synthetic profiles, a quantitative analysis is required to verify that the generator captures all of the characteristics and behaviors present in the real data. In this section, we present multiple tests to validate the quality of the synthetic data.

1.4.2 Wasserstein Distance

As explained earlier, the goal of the generator is to learn a mapping function from the known noise distribution to the distribution of real data. Training is successful when the distribution of generated data matches that of the real data. Wasserstein distance is a measure of distance between two distributions, and it can be used to quantitatively assess how close the distributions of generated and real data are.

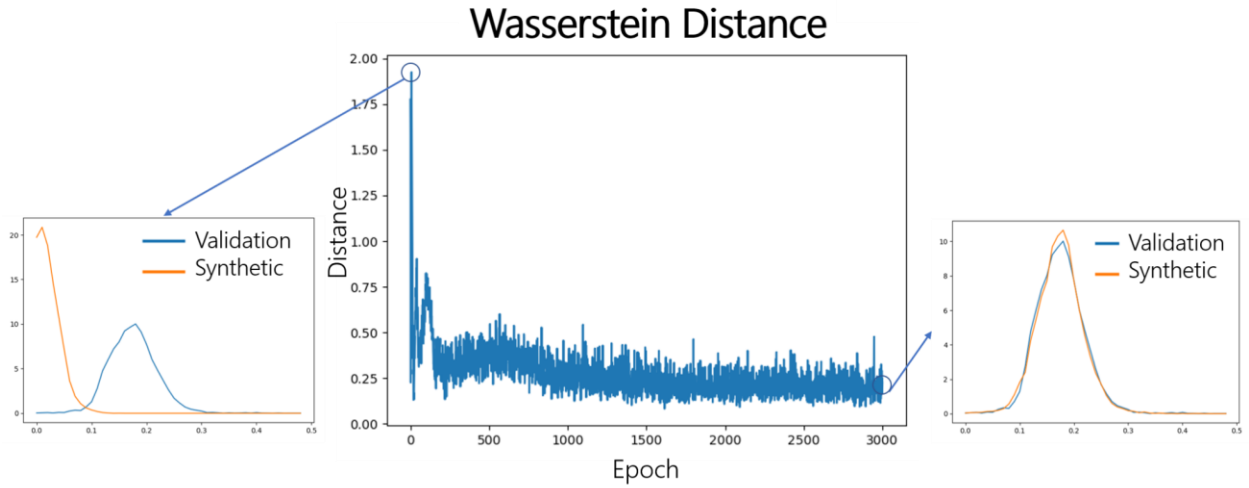

Figure 1.7 Center plot: Wasserstein distance between real and generated data as a function of the epochs. Side plots: comparison between the distribution of real data (blue) and generated data (orange) at epoch 0 (left) and epoch 3000 (right).

The center plot in the figure above shows the Wasserstein distance computed during training at each epoch between a batch of generated data and a batch of validation data. It can be seen that as the training progresses, the distance between the distributions tends to zero. This can be further seen by looking at the two smaller plots on either side. The plots to the left and right show the histograms (empirical distributions) of the real data (blue) and that of the generated data (orange), at epoch 0 (left plot) and at epoch 3000 (right plot), respectively. While initially the two distributions are very different, at the end of training the generated data almost perfectly matches the distribution of real data.

1.4.3 Power Spectral Density

An important characteristic of time-series load data is its periodicity. Because loads are tied to the routine and behavior of people, they present different recurring patterns. One way to identify these periodicities is by looking at the power spectral density of the time-series data. The figure below

shows the power spectral density for real data (blue) and generated data (orange). It can be seen that the two plots match very closely, confirming that the generated data captures the periodic behavior of real data. It is also interesting to look at the various peaks that appear in spectral density: in particular, the highest peak, which occurs at a frequency of 0.04/hour, corresponds to a time period of 24 hours, thus representing the daily load cycle.

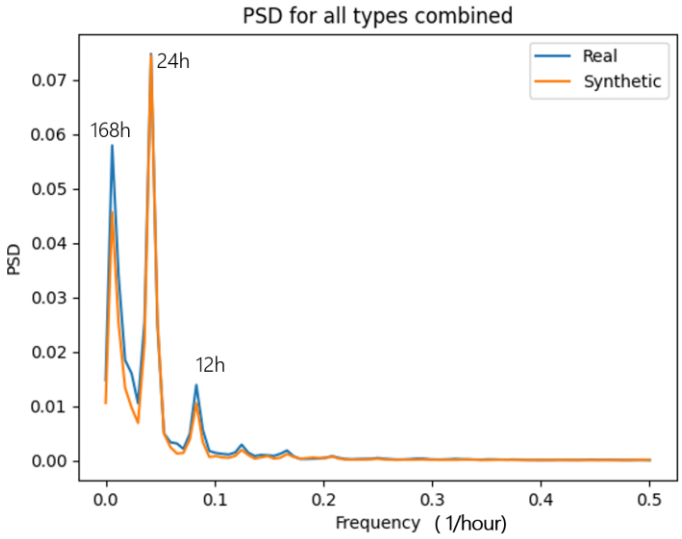

Figure 1.8 Comparison between the power spectral density of real data (blue) and generated data (orange).

1.4.4 Forecasting Application

The main goal of this paper is to create a mechanism for the generation of realistic synthetic load data that can be used by researchers when real data is either not available or not rich enough. In the next two sections we present the results of two example applications that show that the synthetic data successfully captures the behavior of real data and that it can be used for downstream applications.

One of the most common uses of time-series load data is the development of the forecasting algorithms needed for power system operations and markets. While many different techniques are used, often in combination, one of the latest advancements in ML-based forecasting is a class of recurrent neural networks called long short-term memory (LSTM). Because feedback loops are present, the LSTM architecture can process sequences of data (such as time-series data) maintaining a memory of the previous inputs. To verify the quality of our load data generator, we trained an LSTM on a batch of synthetic data and then tested the learnt model on the real data.

An LSTM network with three layers and 48 units per layer is trained to predict the value of a load at one point in time, given the previous 48 hours (48 points). This model is trained on two separate datasets independently: synthetic mainly residential summer profiles and synthetic mainly residential fall profiles. Each of the datasets consists of 1200 week-long profiles generated using the trained cGAN according to their respective labels. To evaluate the performance of the LSTM, for each of the two load types, the trained models are used to predict the load values of two batches

of profiles: new generated data and real data of the same class. The percentage error between the forecasted value and the actual value is computed for each profile in a batch and the first and second moments are computed. The table below summarizes the results of the forecasting test for the two types of loads (summer and fall residential). In both cases we can see that even though the model was trained only on synthetic data, the error when applied to real data is comparable. In general, this suggests that a user could train a ML model on our synthetic data and be confident that it would still capture the characteristics of real data.

Table 1.1 Comparison of the forecasting error between generated and real load data, for summer and fall residential profiles.

Load label	**Testing data**	**Percentage Error**	
		Mean	Std. Dev.
Summer - Residential	Synthetic	4.37	5.26
	Real	5.30	5.17
Fall - Residential	Synthetic	5.82	7.10
	Real	5.92	5.17

1.4.5 Optimal Power Flow

The synthetic data is also tested to verify that the generated profiles can be correctly mapped to a power system model. One way to check this is to ensure that all the resulting load cases form a feasible AC power flow. This test is performed by first generating individual, week-long profiles for each load in the Polish test case: this system model has 2383 buses and 1822 loads. Two datasets are generated: one corresponding to a winter week and one for a summer week. Each of these profiles is mapped to the Polish system loads: since the base case of the Polish system is a peak case, the profiles are matched so that the peak of each profile corresponds to the base case value. AC optimal power flow (OPF) is then run on each case corresponding to each of the 168 hours of the week. The results showed that OPF converged in every case to a solution with bus voltages and generator outputs within their predefined limits.

1.5 Conclusion

We presented a method to generate synthetic transmission load data at a bus level leveraging conditional generative adversarial networks. A user can specify the time of the year and type of load for which to generate time-series load profiles. Extensive testing is performed, and we have verified the validity of our method and quality of the generated data. Our trained generative model will be available to researchers to be used for any type of power system and ML application. Moreover, the proposed conditional learning framework can be leveraged for the generation of other datasets highlighting different characteristics, such as the level of penetration of renewables or electric vehicle charging. Finally, we are working on expanding the generative model to create a tool for the generation of synthetic datasets at any time resolution (from 30 samples/second to a few samples/week) and for any length of time (from a few minutes to multiple years).

2. Generation of synthetic voltage data for dynamic studies

2.1 Introduction

Over the past decade, thousands of PMUs have been deployed in backbone transmission systems across North America and around the world. Transient dynamic data captured by PMUs are of value to the research community of distinct research interests. However, these data are mostly prevented by the critical energy/electric infrastructure information (CEII) across the U.S. Besides, the eventful PMU data shortage is also partially attributed to the fact of limited real-world oscillation events recorded by PMUs. For example, we find only seven recorded system-wide voltage oscillation events in one real PMU data set from an anonymous large U.S. utility company that contains measurements of approximately 400 PMUs from 2017 to 2019. The shortage of eventful PMU data becomes a bottleneck preventing data-driven methods, such as oscillation detection and event identification, to develop, calibrate, and validate based on the real data set.

While researchers recently have contributed to the creation of large-scale realistic synthetic grid models [1] for analysis such as macro-scope energy portfolio transition [2] and major event reproduction [3], existing work is infeasible to transfer real eventful PMU data to synthetic grids and exploit their values via simulation. Alternatively, other recent research work contributed to developing machine learning-based power system data generation methods, such as load profiles generation, renewable generation scenario generation [4], and eventful PMU generation [5]. While the prior work proposed the potential use of synthetic PMU data, machine learning-based approaches that generate realistic multi-time-scale eventful PMU data of arbitrary length nevertheless need efforts to develop, with several gaps in the existing work. First, prior success of PMU data generation methods in small-scale IEEE standard systems may not meet the demand of synthetic data based on the real-world PMU data set. Second, the short horizon of synthetic data limits the generalization of its applications. Finally, the lack of incorporating time-varying load conditions may undermine the fidelity of synthetic PMU data of long length.

To this end, this section aims at creating massive realistic synthetic eventful data from the given limited real data using the proposed two-stage method that leverages Generative Adversarial Nets (GAN) [7] and Neural Ordinary Differential Equations (NeuralODEs) [6]. The key to the success of the proposed idea is to make sure that the synthetic data are realistic and diverse rather than a simple clone of the real data, which is achieved by the proposed PMU data creation algorithm. Here, with only access to the power flow model of the large-scale real system but no knowledge of the dynamic model, we will scale up the limited real eventful PMU data set by a multi-time-scale PMU voltage data generation method that leverages GAN in creating time-varying load conditions and Neural ODE in mimicking system transient behaviours during voltage oscillation events, which is potentially generalizable in other real power systems.

In summary, the proposed multi-time-scale eventful PMU voltage measurement generation method can create multiple realistic-looking PMU streams that capture the patterns of load change and system oscillation in distinct time scales. The synthetic time-series load data (generated by the method in the last section) can capture the inherent property of load change at varying resolutions by combining the load time series of various lengths and time resolutions. Voltage measurement

oscillation generation: The synthetic small-signal oscillation data can learn the pattern of real system voltage oscillation events dominated by the underlying ODE.

2.2 Problem Statement

Consider a set of historical eventful PMU measurements obtained covering pre-event, during event and post-event periods, where the label refers to the event type. For PMUi, we denote historical phasor domain voltage and current data byV_i and I_i, which are collected by a time window of $T$with a time step ΔT. Define one entire sample of an event as $S = [V_1, I_1, \ldots, V_{N_{PMU}}, I_{N_{PMU}}]$, where N_{PMU} is the total number of PMUs. The data creation problem tackled by this paper is to develop a data creation algorithm to create synthetic eventful PMU data of certain event type using the corresponding historical samples $\{S_i\}, i = 1, \ldots, N_s$ as the training data, where $N_S$is the number of the historical samples, in such a way that the created synthetic data exhibit relevant properties possessed by the historical data.

2.3 Review of Generative Models

GAN was proposed in 2104 which has now arguably become one of the most popular and successful deep generative models, of which the architecture is shown in the following figure.

Both generative model (generator) G and discriminate model (discriminator) D are implemented by neural networks, which are trained by optimizing the following objective function J:

$$min_G\, max_D\ J = \mathbb{E}_{\boldsymbol{x}} log(D(x))\ + \mathbb{E}_{\boldsymbol{z}}\, log(1 - D(G(z)))$$

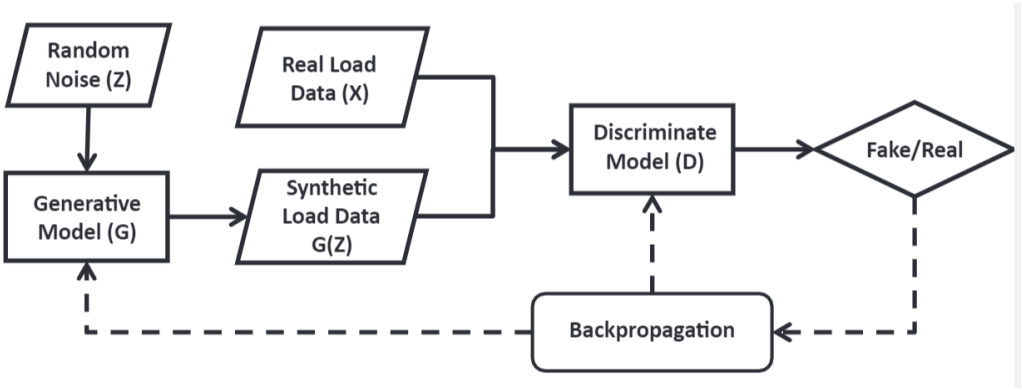

Figure 2.1 The architecture of vanilla GAN.

2.3.1 Review of NeuralODE model

The Neural ODE model, which is used for time-series modeling, contains two key components: a neural network and an ODE solver. Instead of specifying a discrete sequence of hidden layers, this model parameterized the derivative of the state, using a neural network f. It can be trained by supervised learning to minimize the scalar-valued loss function as follows:

$$in_f \; L(s) = \frac{1}{t_1 - t_0} \sum_{t=t_0}^{t_1} || \int_{t_0}^{t} f(s(\tau))d\tau + s(t_0) - s(t)||_2$$

where $\int_{t_0}^{t} f(s(\tau))d\tau + s(t_0)$ represents the estimated state based on the neural network given the initial state $s(t_0)$, $s(t)$ represents the measurements at time t, and the function f is a neural network parameterized by θ_f that indicates how the measurements evolve along the timeline.

2.4 Two-stage Synthetic Eventful Voltage Measurement Generation

2.4.1 Frame of Two-time-scale Eventful PMU Data Generation under Time-varying Load Condition

Here, we assume that the created multi-time-scale PMU measurements are a linear combination of steady state and small signal oscillation, which are respectively determined by the pattern of load change and the nature of system dynamics. Therefore, we separate the task of multi-time-scale eventful PMU data generation into two subtasks. First, we aim to create realistic time-varying load profiles and then estimate the steady-state voltage measurements via power flow simulation based on the accompanied real static PSS/E system model. Second, we learn the pattern of small signal voltage oscillation based on the pre-processed real eventful data. With such instructive principle, leveraging the GAN and Neural ODE, a novel two-stage PMU data generation algorithm is proposed as shown in the figure below, where GAN creates synthetic time-varying load profiles, while Neural ODE creates the voltage profiles under the given time-varying load condition. In the training process, we train the GAN and Neural ODE models separately with the limited real PMU data. In the data creation process, we combine the well-trained G and f models to generate the PMU data of an entire event of arbitrary length.

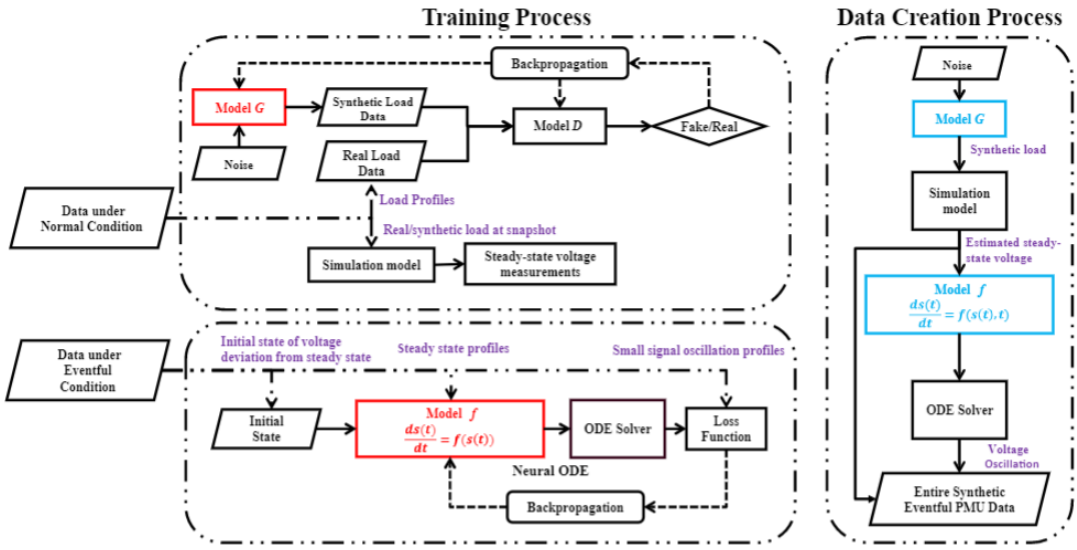

Figure 2.2 The proposed two-stage GAN-based algorithm incorporates the architecture of GAN and NeuralODE models.

The rest of this section will introduce the detailed algorithms of load profile generation, steady state estimation, and voltage oscillation generation, and show the methods of data quality check.

2.4.2 Load Profile Generation

This algorithm aims to develop a scheme to generate realistic time-series load data varying in length from a few minutes to a year and at varying resolutions (from one sample per week down to 30 samples per second). This approach relies on training a generative model for different time resolutions and length levels; when all the models are trained, the synthetic data can be generated at the appropriate levels and combined depending on the user's requirements. The basic framework for each level's training consists of using generative adversarial networks (GANs). GANs are a novel machine learning framework in which a generative model (or generator) is trained by making it compete against a discriminator. The goal of the generator G is to capture the distribution of the real data pr, while the discriminator D is trained to distinguish the real data from the synthetic data produced by the generator. The details of this technique are introduced in the last section, namely generation of synthetic load time series data for steady state application.

2.4.3 Steady-state Voltage Measurements Estimation

We estimate the steady-state voltage measurements under a certain load condition via simulation on the accompanied static PSS/E model. Note that the obtained system model is only compatible for solving power flow, rather than transient simulation due to the lack of dynamic modelling of generators.

Leveraging the well-trained load profile generation model, we can generate massive realistic load profiles during a certain time period, which possess similar pattern but meanwhile exhibit diversity. Assigning load profiles to load buses in the simulation model and proportionally scaling generation dispatch, we can obtain the steady-state voltage measurements by solving power flow at each moment. Note that applying realistic generation dispatch methods is out of this paper's scope.

2.4.4 Voltage Oscillation Generation

In the context of our problem, with the voltage measurements at all targeted buses as the state, the corresponding f function of the eventful time series is time-invariant during a short period. In other words, given an initial state, the entire trajectory is uniquely defined. Therefore, we intuitively use the NeuralODE to implement the event time series modelling by supervised learning. The NeuralODE model will be another core of the proposed two-stage networked eventful PMU data creation algorithm.

The NeuralODE model is trained independently by a supervised learning to minimize the scalar-valued loss function:

$$min_f \ L(s) = \frac{1}{t_1 - t_0} \sum_{t=t_0}^{t_1} || \int_{t_0}^{t} f(s(\tau))d\tau + s(t_0) - s(t)||_2$$

where $s(t)$ represents the measurement at time t; the integral $\int_{t_0}^{t} f(s(\tau))d\tau + s(t_0)$ represents the estimated state based on the neural network for an initial state, and the function f is a neural network parameterized by θ_f that indicates how the measurements evolve along the timeline. Note that since we assume the post-event power system model is a time-invariant dynamic system, we can randomly sample arbitrary PMU data segments in the post-event period as the training data of the Neural-ODE model.

The proposed two-stage algorithm has two advantages, respectively addressing the two challenges of physical constraints and computation burden. On one hand, this two-stage design makes the GAN model create the PMU measurements at only very few special time instants, which significantly reduces the size of the GAN model and improves the computational efficiency. On the other hand, this algorithm embeds the ODE format by incorporating the NeuralODE model, which can effectively learn the temporal correlation of the oscillation measurements from different PMUs.

Additionally, we verify the physical meaning of the post-event synthetic PMU data. In this paper, modal analysis is used to quantitatively show whether synthetic voltage oscillation data possess realistic dynamic characteristics of power systems, i.e., modal properties. Specifically, we use Prony analysis, a classical ring-down analysis method, to analyze the synthetic data and extract important modal properties including oscillation frequency and damping.

Since modes are the fingerprint of a given linear system (or the linearized part of a nonlinear dynamic system), we can compare the modes of the real data and synthetic data for validating the fidelity of the synthetic data. Without loss of generality, here we select the voltage angle measurements for validation. Details of the validation are summarized below:

- Modal Property Estimation: Calculate the oscillation frequency, damping coefficient, amplitude, and phase of all active modes via Prony analysis for real and synthetic voltage angle profiles.
- Modes Selection: Only the modes with amplitude greater than a threshold are selected as active modes. The threshold is fixed as the 10% of the maximum value of all mode amplitudes.
- Validation: For each synthetic profile, modal frequencies and damping ratios of the active modes are compared with those of all real profiles at the same generator bus. We say that the synthetic profile passes the test if all active modes of the tested synthetic profile appear in the real data. We declare a pair of modes as the same mode if their relative error is less than 5%.

2.5 Results of Voltage Oscillation Measurements Generation

Case Study on IEEE 39-bus System

The IEEE 10-machine 39-bus system is simulated to provide the training and test data sets. Specifically, we focus on three types of events: bus fault, line tripping and load shedding. Table below shows the simulation configuration of these three types of events. Note that both bus fault and line tripping events are triggered by solid three phase grounding faults, and all simulations start from the same steady state. We randomly sample the simulation parameters and get the

simulated eventful data from PowerWorld, which include voltage magnitude and angle profiles at all generator buses covering the pre-event, during event and post-event periods.

Table 2.1 Simulation setting of three event types, including bus fault, line tripping, and load shedding. Note that both of bus fault and line tripping events are triggered by solid three phase grounding faults.

Event Type	Parameter	Simulation Setting
Bus fault	Fault duration	0-0.1 s
	Fault location	Random bus
	Sampling frequency	60 Hz
	Time window	15 s
Line tripping	Fault duration	0-0.1 s
	Fault location	Random line
	Sampling frequency	60 Hz
	Time window	15 s
Load shedding	Shedding percentage	5-50%
	Fault location	Random load bus
	Sampling frequency	60 Hz
	Time window	15 s

Figure below illustrates the real (top row) and synthetic (bottom row) eventful voltage angle profiles at all generator buses in a 5s time window which includes the pre-event, during event and post-event periods. Note that we randomly select the synthetic sample for each event type and then determine the real sample by looking for the one that is closest to the synthetic sample. It is observed from the visual comparison that: (i) real and synthetic profiles have similar settling patterns; (ii) the ranges of real and synthetic profiles are nearly the same. These observations imply that the proposed two-stage GAN-based model can generate transient PMU data that capture the inherent temporal correlation.

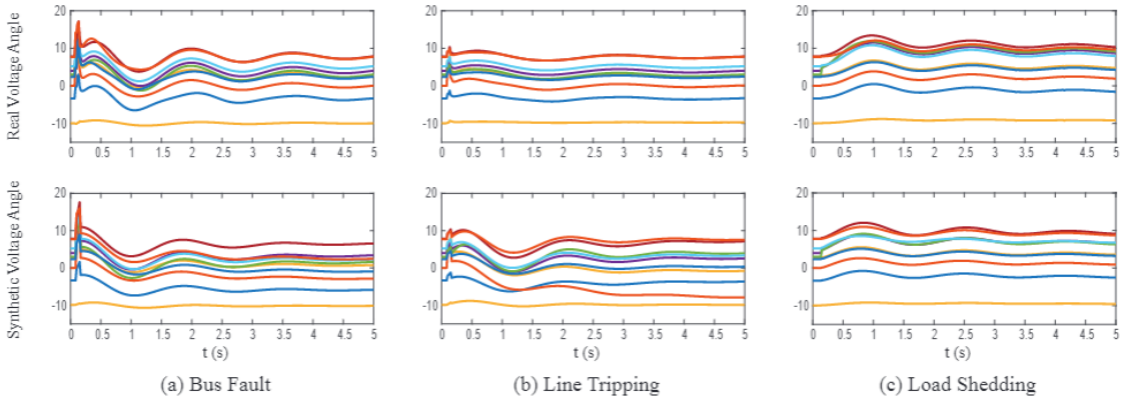

Figure 2.3 Visual comparison between real (top) and synthetic (bottom) eventful voltage angle profiles at generator buses. The examples of bus fault, line tripping and load shedding are respectively shown in the subfigure (a), (b) and (c).

According to the modal analysis-based synthetic data quality check method, we separately test the synthetic data of all event types, with the training and test data sets generated being the benchmark. The modal analysis test result of bus fault, line tripping and load shedding are shown in the table below. From this table, we can observe that: (i) fidelity: High success rate indicates that the synthetic data approximately capture the dynamic characteristics of the real data, (ii) diversity: The difference of the success rate between training and test benchmark indicates that the generative model does not simply memorize the training data but also it creates new and meaningful modes that exist in the test data set.

Table 2.2 Modal analysis test

Event type	**Bus fault**	**Line tripping**	**Load shedding**
Accuracy Train	88.10%	86.60%	80.28%
Accuracy Test	90.23%	87.30%	82.81%

Case Study on A Real Large-scale System

The available real PMU data set is obtained from an anonymous U.S. utility company, which covers the duration from 2017 to 2019 and contains single phase and positive sequence measurements of voltage and current at a rate of 30 samples per second. The accompanied static system model contains 7577 buses, 709 generators, 338 loads and 10338 branches, where there are a total of 366 PMUs deployed starting from 2017.

We implement the same voltage oscillation generation algorithm as implemented in the last subsection. Specifically, we achieve the model f in NeuralODE by a 3-layer fully connected neural network, of which the number of neurons is respectively 128, 128 and the number of measurements. To demonstrate the learning capacity of NeuralODE, we will show the results of synthetic voltage oscillation data in events of distinct durations that exhibit different dynamic behaviors. We train and test the NeuralODE model based on a 10-second event, as shown in the

figure below. The visual comparison and quantified average error demonstrate the strong learning capacity. Moreover, the extended synthetic profiles in the following additional 5 seconds show its flexibility of generating time series of arbitrary length.

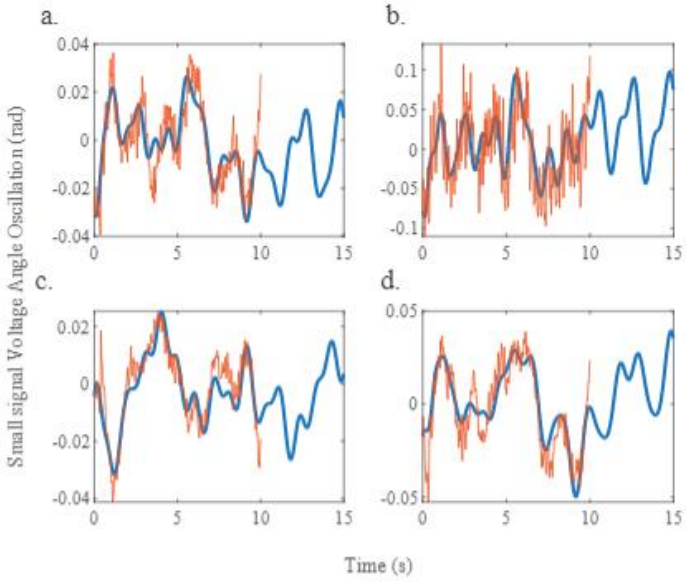

Figure 2.4 Visual comparison between real (red) and synthetic (blue) eventful voltage angle profiles at illustrated 4 PMUs.

Illustrated Application of Synthetic Voltage Oscillation Data

We show that the number of samples in the training data set has a significant impact on the event classification accuracy, and then show the improvement of the event classification accuracy by incorporating the synthetic PMU data based on the case of IEEE 39-bus system.

Based on the case of IEEE 39-bus system, we have a training data pool and a test data set for the event classifiers that respectively have 400 and 800 samples for each event type. We will randomly sample several training data from the training data pool and test the trained event classifiers on the whole test data set.

Firstly, we show the impact of the number of samples in the training data set on the event classification accuracy in the table below. The first row of the table represents the number of samples in the training data set for each event type. The notation in the first column means the classification methods. For example, PCA-SVM refers to a classification method that extracts the features by PCA and classifies the features by SVM. We randomly pick the given number of samples in the training data set, and then train all four event classification methods. To mitigate the randomness, we repeat this procedure 10 times and calculate the mean and standard deviation of the event classification accuracy as shown in each cell. The following can be observed from this table: although these four classification methods have distinct performances, the increasing number of training data always benefits their classification accuracy in the sense of mean value and the standard deviation. Given the fact that there are always only a limited number of real

events, the accuracy of these classification methods is expected to be low in the practical application. It also shows the need for more data to improve the event classification accuracy.

Table 2.3 Impacts of the number of training data on the event classification accuracy

Synthetic training data	**20**	**40**	**60**	**80**	**100**
PCA-SVM	68.97 ± 7.79%	72.10 ± 2.68%	76.15 ± 2.40%	74.80 ± 1.64%	76.28 ± 1.88%
WT-SVM	80.88 ± 3.11%	84.76 ± 2.40%	88.24 ± 2.29%	90.98 ± 1.01%	91.16 ± 1.36%
PCA-Ensemble	79.27 ± 3.85%	84.72 ± 2.19%	89.38 ± 1.55%	92.08 ± 0.97%	92.93 ± 0.76%
WT-Ensemble	82.73 ± 3.03%	90.90 ± 1.92%	95.71 ± 0.82%	96.83 ± 0.37%	97.28 ± 0.59%

Next, we show the improvement of the event classification accuracy by incorporating the synthetic PMU data as shown in the table below. Note that NeuralODE models used to generate the synthetic data are trained by only 20 real samples for each event type. Since the data creation method can generate massive synthetic data, we also randomly sample a given number of synthetic data samples and mix them with the real samples for training the event classifiers. Then, we use the created hybrid training data set to train all four event classification methods. To mitigate the randomness, we repeat this procedure 10 times as well and calculate the mean and the standard deviation of the event classification accuracy as shown in the table below. Comparing two tables, we have the following observations:

- Incorporating the synthetic data can effectively and consistently improve the event classification accuracy by 2 to 5 percent, compared to the results based on only 20 real training data for each event type.
- When the number of the real data increases from 20 to 100, it always leads to a better classification accuracy than the cases with synthetic PMU data, meaning the synthetic data can aid in the classification accuracy due to the lack of training data but cannot replace the real ones.
- When the synthetic data overwhelms the real training data, the classification accuracy is not negatively affected, implying that the synthetic data are of good quality.

Table 2.4 Improvement of event classification accuracy by incorporating synthetic data

Synthetic training data	**100**	**200**	**300**	**400**
PCA-SVM	$73.85 \pm 1.69\%$	$73.77 \pm 1.27\%$	$73.01 \pm 2.83\%$	$73.32 \pm 1.72\%$
WT-SVM	$82.73 \pm 2.29\%$	$82.69 \pm 1.27\%$	$81.76 \pm 2.02\%$	$81.52 \pm 2.41\%$
PCA-Ensemble	$85.47 \pm 2.94\%$	$82.32 \pm 2.23\%$	$82.52 \pm 2.82\%$	$81.74 \pm 2.77\%$
WT-Ensemble	$90.05 \pm 1.42\%$	$88.97 \pm 2.31\%$	$90.27 \pm 1.50\%$	$88.16 \pm 2.18\%$

2.6 Conclusions

In this project, we have presented a method to generate synthetic transmission load data at a bus level leveraging conditional generative adversarial networks. A user can specify the time of the year and type of load for which to generate time-series load profiles. Extensive testing is performed, and we have verified the validity of our method and quality of the generated data. Our trained generative model will be available to researchers to be used for any type of power system and ML application. Moreover, the proposed conditional learning framework can be leveraged for the generation of other datasets highlighting different characteristics, such as the level of penetration of renewables or electric vehicle charging. Finally, we are working on expanding the generative model to create a tool for the generation of synthetic datasets at any time resolution (from 30 samples/second to a few samples/week) and for any length of time (from a few minutes to multiple years). On the other hand, we propose to create synthetic voltage oscillation data via a two-stage generation method, which can be separated into steady-state and oscillation data generation. This approach can scale up the otherwise limited real-world PMU data as follows. First, it leverages the capability of GAN to guarantee the diversity of massive synthetic data. Second, it leverages NeuralODEs to provide meaningful voltage oscillation time-series data. Such synthetic data sets can then be fed into subsequent monitoring and decision-making processes. As an example, we show that the synthetically created PMU data improves the performance of data-driven event classification. We validate the fidelity of the synthetic data via visual comparison and modal analysis approaches and verify that the synthetic data can improve the accuracy of four selected event classification methods.

References

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

[2] Y. Xu et al., "U.S. Test System with High Spatial and Temporal Resolution for Renewable Integration Studies," *2020 IEEE Power & Energy Society General Meeting (PESGM)*, 2020, pp. 1-5.

[3] Wu, Dongqi, Xiangtian Zheng, Yixing Xu, Daniel Olsen, Bainan Xia, Chanan Singh, and Le Xie. "An Open-source Model for Simulation and Corrective Measure Assessment of the 2021 Texas Power Outage." *arXiv preprint arXiv:2104.04146 (2021).*

[4] Y. Chen, Y. Wang, D. Kirschen and B. Zhang, "Model-Free Renewable Scenario Generation Using Generative Adversarial Networks," *in IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3265-3275, May 2018.

[5] X. Zheng, B. Wang and L. Xie, "Synthetic Dynamic PMU Data Generation: A Generative Adversarial Network Approach," *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, 2019, pp. 1-6.

[6] Chen, Ricky TQ, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. "Neural ordinary differential equations." *arXiv preprint arXiv:1806.07366 (2018)*.

[7] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." *Advances in neural information processing systems* 27 (2014).

[8] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014. [Online]. Available: https://arxiv.org/abs/1411.1784.

Part II

Real time detection, identification, and prediction of oscillatory events and contingencies using synchrophasor data: A machine learning approach

Lalitha Sankar
Anamitra Pal

Harish Chandrasekaran, Graduate Student (M.S.)
Antos Cheeramban Varghese, Graduate Student
Pooja Gupta, Graduate Student
Reetam Sen Biswas, Graduate Student
Nima Taghipourbazargani, Graduate Student

Arizona State University

For information about this project, contact

Lalitha Sankar
Arizona State University
School of Electrical, Computer, and Energy Engineering
Goldwater Center, R# 436,
551 E Tyler Mall,
Tempe, AZ 85281
Phone: 480-965-4953
Email: lalithasankar@asu.edu

Power Systems Engineering Research Center

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

For additional information, contact:

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

Notice Concerning Copyright Material

Table of Contents

List of Figures

List of Tables

1. Event Detection and Prediction of Island Formation

1.1 Introduction

Extreme weather events such as hurricanes can cause unforeseen contingencies in power systems. Despite real time contingency analysis (RTCA) and security constrained economic dispatches trying to ensure N-1 reliability, these events can cause cascading failures. Power outages in Louisiana during Hurricane Gustav in 2008 [1], and power interruptions in Florida during Hurricane Irma in 2017 [2] are examples of such extreme weather events. The focus of this research is on detecting and predicting extreme events that result in island formation due to multiple outages.

A critical component of contingency analysis for dealing with extreme weather events is the speed of detection. The power flow based contingency analysis is not fast enough to perform an *exhaustive* N-1 RTCA [3]. In the event of successive outages, N-k contingency analysis has to be carried out and the computational burden increases even more [4]. Conversely, if the analysis is limited to small subsets, critical scenarios might be missed. A tool for enhanced situational awareness of critical contingencies in the power system is required especially in the presence of potential cascading failure-triggering outages.

A novel approach based on graph-theoretic principles and network analysis is proposed here to protect the system against uncontrolled islanding [5]-[7]. Initially, the focus is on identifying contingencies that will create vulnerable bottlenecks in power networks when multiple outages manifest in rapid succession [5]-[6]. Subsequently, mitigation strategies for the identified bottlenecks are discussed [7]. The proposed algorithm requires time synchronized bus power injections as one of the inputs. A machine learning (ML)-based framework is proposed to satisfy this requirement. Figure 1 describes the outline of the proposed approach, which consists of three research segments. The first segment delves with time synchronized bus power injection estimation. The second segment focuses on saturated cut-set detection. The final segment describes the mitigation strategies for alleviating saturated cut-sets.

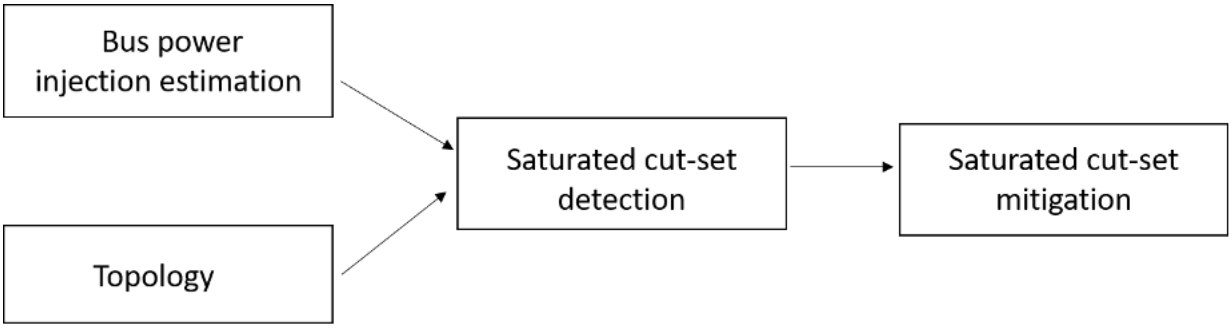

Figure 1.1 Outline of the proposed approach

1.1.1 Mathematical Background

A cut-set is defined as a set containing a minimum number of branches which when removed splits the network into two disjoint islands. Any cut-set which transfers more power from one area to another than is permissible by the maximum power transfer capability of that cut-set is a *saturated* cut-set. Let branches e_1, e_2,…, e_z belong to the cut-set K. If the flows through the cut-set branches are f_1, f_2,… f_z, and the ratings of those branches are r_1, r_2,… r_z, cut-set K is called a saturated cut-set if the following equation holds:

$$\sum_{i=1}^{z} f_i \leq \sum_{i=1}^{z} r_i \,, \forall e_{li} \in K \quad (1)$$

where $\sum_{i=1}^{z} f_i = F_K$ is the actual power flow occurring through cut-set K and $\sum_{i=1}^{z} r_i = R_K$ is the maximum power that can be transferred across cut-set K. The objective here is to identify if a contingency creates a saturated cut-set or not.

Saturated or overloaded cut-sets denote vulnerable bottlenecks in power grids across which islanding is likely [8]. Hence, it is important to detect contingencies that saturate (or overload) a cut-set. However, it must be noted that a single branch can be associated with multiple cut-sets. Thus, to assess the impact of the loss of any power system asset (e.g., line, transformer) on any cut-set of the power system, the power transfer capability of *all cut-sets associated with that asset* should be checked. For a big system containing thousands of buses, a single asset could be associated with numerous cut-sets. Therefore, quantifying the impact of an outage on any cut-set of the power network is a computationally intensive task.

As cascading failures can lead to system islanding, a methodology to detect such scenarios is proposed first. Subsequently, efficient methods to take corrective actions when a saturated cut-set is detected so that cascading failures are prevented, is discussed. The algorithm for the detection of saturated cut-set requires the time synchronized bus power injections and the topology of the network. The first segment of the research thus focuses on bus power injection estimation from phasor measurement unit (PMU) measurements.

1.2 Time Synchronized Bus Power Injection Estimation

The bus power injections are an essential input to the proposed saturated cut-set detection algorithm. In general, there are many challenges to estimating the time synchronized bus power injections at every bus in a network. A combination of classical static state estimation and circuit laws could have been thought of as a method to estimate the bus power injections. However, since this study focuses on situational awareness tools for extreme event scenarios, there is a high possibility that the classical state estimation algorithms might face convergence issues under such conditions [9]. Moreover, the bus power injections thus computed will not be time-synchronized. A PMU-based linear state estimator in a system that is completely and independently observed by PMUs, is a possible solution. However, due to the high cost of synchrophasor infrastructure [10], PMUs are placed only on some of the buses in the network (e.g., the highest voltage buses). Therefore, it makes sense to come up with a new methodology to estimate time synchronized bus power injections using an extremely limited number of PMUs. To attain this objective, a data attain

this objective, a data driven mapping rule estimation technique using PMU measurements is proposed. The proposed method learns the mapping rule between the PMU measurements and bus power injections using ML.

1.2.1 Kernel density estimation

ML algorithms require a large amount of historical data for their training. For a given loading condition, an optimal power flow (OPF) solution can be used to generate bus power injections and synthetic PMU measurements. The challenge, however, is generating realistic load scenarios for any test system. Hourly power consumption data for a year is publicly available for the 2000-bus synthetic Texas system [11]. However, this results in only 366 values per bus for a given hour. To create more data, a probability density function (PDF) is slapped onto the power injection data obtained at a given hour for every bus using Kernel Density Estimation (KDE). KDE is a non-parametric method to find the best-fit PDF to a given dataset. Once the PDF is obtained, any number of scenarios can be generated using it.

The different bandwidths in KDE introduces a bias-variance trade-off problem. This challenge is overcome by using the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality for bandwidth tuning. By employing the DKW inequality, a 95% confidence interval is maintained. The effect of this tuning can be observed in the following figures.

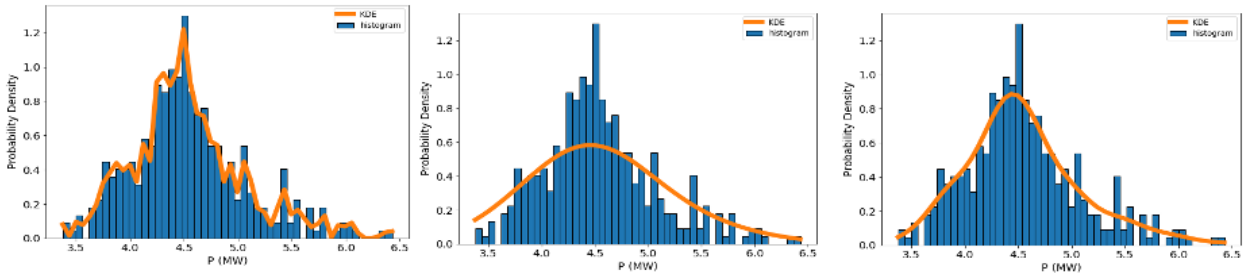

Figure 1.2 Bandwidth tuning for KDE

The three sub-figures in Figure 2 have the same histogram (blur bars) for active power injection at a bus. The estimated PDF of active power injection at the same bus using KDE is shown using the orange curve. The left subfigure shows the PDF obtained when a low bandwidth is used in KDE. It can be observed that the PDF obtained in this case is overfitting the actual histogram. Conversely, the middle subfigure corresponds to using a high bandwidth in KDE and consequently, the resulting curve underfits the actual histogram. The right most figure corresponds to optimal bandwidth obtained by using DKW inequality with 95% confidence interval. This PDF appropriately fits the actual histogram and was thus used for ML training.

The best-fit PDF was identified for different loads of the 2000-bus synthetic Texas system. The results for best-fit PDFs for different buses are shown below. Once a PDF is obtained, as many data samples as desired can be generated. Next, the buses of the 2000-bus synthetic Texas system were mapped to the load buses in the IEEE 118-bus test system based on similar power ratings. Subsequently, 15,000 load scenarios are sampled using these PDFs. For each of the 15,000 load

scenarios, OPF was solved in MATPOWER to generate simulated PMU measurements and bus power injections. This data was used to train ML algorithms to learn the underlying mapping rule.

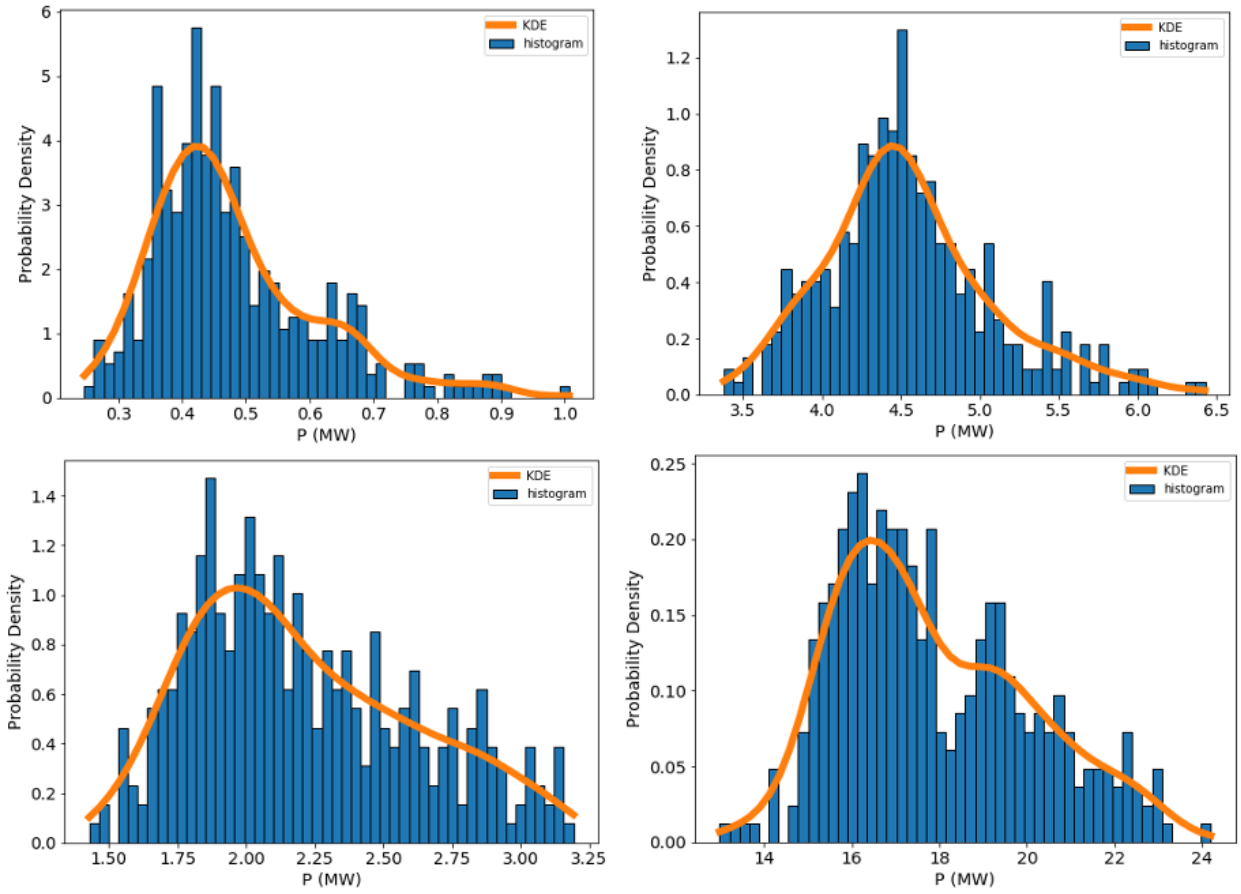

Figure 1.3 KDE for different buses

1.2.2 Data-driven methods for time synchronized bus power injection estimation

PMU measurements and the corresponding bus power injection data generated are used for the training of two ML algorithms, namely, deep neural networks (DNNs) and support vector regression (SVR). The objective is to estimate the mapping rule considering a setup where a limited number of PMUs are present in the network, thereby emulating a real-world scenario. In particular, IEEE 118-bus test system was selected for this experiment and PMUs were assumed to be placed on only the 11 high voltage buses of this network. The objective of this problem was to estimate the time-synchronized bus power injections of all the buses in the IEEE 118-bus system using PMU measurements from just the 11 high voltage buses.

First, a DNN was trained to learn the mapping relation between the 11 PMU measurements and the 118 bus power injections. The input features consist of voltage measurements of the buses on which PMUs are placed and the current measurements of the branches that are observed by the PMUs. It was found that a DNN with 4 hidden layers is giving optimal performance. The training

of the DNN was conducted using the ADAM optimizer. L2 regularizer was used to avoid the problem of overfitting. Table I shows the detailed list of hyperparameters. The mean absolute relative error (MARE) of the estimated bus power injections and the true bus power injections obtained using DNN are shown in Figure 4.

Table 1.1 DNN hyperparameters for bus power injection estimation

DNN Architecture	
Number of neurons in Input Layer	800
Number of neurons in first Hidden Layer	600
Number of neurons in second Hidden Layer	600
Number of neurons in third Hidden Layer	500
Number of neurons in fourth Hidden Layer	400
Number of neurons in output Layer	118
Loss Function	Mean Absolute Error (MAE)
Learning rate for ADAM optimizer	10^{-3}
Batch Size	64
L2 Regularization	On

Alternatively, an SVR was used to learn the mapping rule and estimate the bus power injections. SVR finds a hyperplane that maximizes the margin between the hyperplane and the support vectors. The learned hyperplane is then used to estimate the power injection from the testing data. A polynomial kernel of degree 2 was used to implement the SVR kernel. The SVR based method has similar set of inputs and outputs as those used for the DNN, i.e., a combination of magnitude and phase angle values of voltage and current measurements obtained from the PMUs placed at the selected buses of the system are used as inputs, and the bus power injections at every bus are the outputs. The MARE of the estimated bus power injections obtained using SVR is shown in Figure 5.

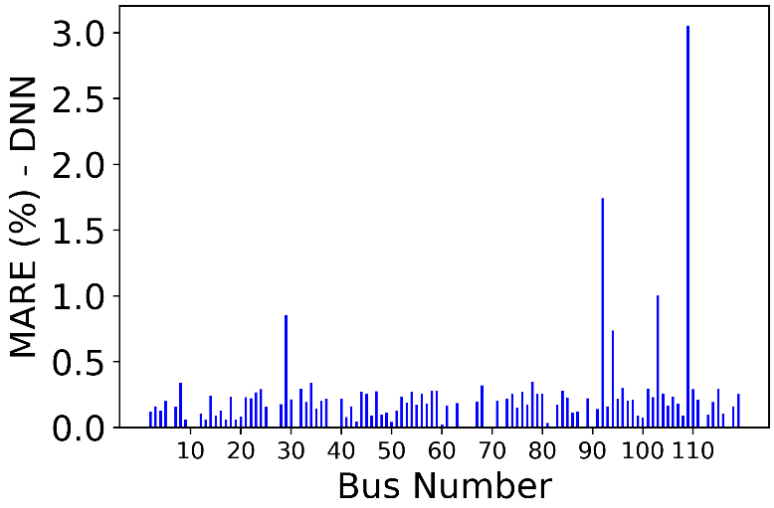

Figure 1.4 Result - DNN for bus power injection estimation

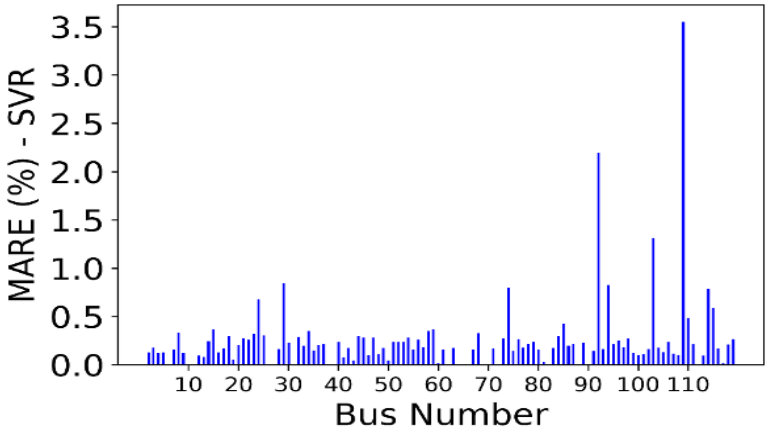

Figure 1.5 Result - SVR for bus power injection estimation

It can be observed that both DNN based method and the SVR based method was able to estimate the synchronized bus power injections to a very high degree of accuracy even though the PMUs were placed only on 11 of the 118 buses. In the case of DNN, the MARE was less than 0.5% for 95% of the load buses, and less than 3% for all the buses. The SVR was able to estimate 91% of the load buses with less than 0.5% MARE, while the highest error was less than 3.5%. Because only 11 PMUs were placed in the 118-bus test system, these data-driven methods are found to give bus power injection estimation to a very high degree of accuracy. The bus power injections thus obtained are used as inputs for the saturated cut-set detection algorithm.

1.3 Fast Identification of Saturated Cut-sets

To uniquely determine if a contingency saturates a cut-set in the network, a novel graph-theory based network analysis technique called the feasibility test (FT) algorithm is developed. The

working principle of the FT algorithm is described as follows. Let a branch e_l (transmission line or transformer) connect buses v_l^F and v_l^T as shown in Figure 6. Since branch e_l is a single element that joins bus v_l^F to v_l^T it is called a *direct path* from bus v_l^F to v_l^T. f_l marked on Figure 6 denotes the flow through branch e_l. There could be many other electrical paths to transfer the power from bus v_l^F to bus v_l^T. An *indirect path* is the path that contains multiple branches from v_l^F towards v_l^T.

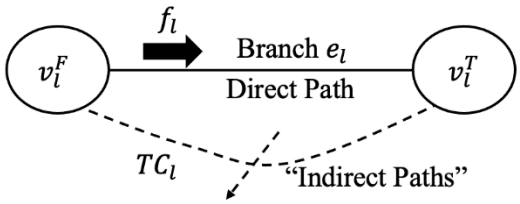

Figure 1.6 Network connectivity between two vertices

The total extra power that can be re-routed through the indirect paths of a branch (denoted as TC_l in Figure 6) is found by FT by using the breadth first search (BFS) graph traversal scheme. If the indirect paths do not have sufficient capacity to reroute the power flowing through the direct path, it means that the outage of the direct path saturates a cut-set in the network (i.e., $TC_l < f_l$); and the direct path is termed a special asset. If the outage of a branch saturates more than one cut-set, the FT screens out the cut-set that is saturated by the largest margin; this cut-set is known as the limiting critical cut-set, K_{crit}. The FT also provides the transfer margin, T_m (calculated as $T_m = TC_l - f_l$) by which K_{crit} is saturated.

The utility of the proposed algorithm for enhanced situational awareness is explained with a case-study on the IEEE-118 bus test system. Due to a hurricane, let the following transmission asset outages occur one after another: 15-33, 19-34, 37-38, 49-66, and 47-69 (marked O_1 through O_5 in Figure 7). From Figure 7 and Table II, following information is obtained when the FT algorithm is applied as outages manifest:

1) *Base-case:* In the base-case scenario, the asset 26-30 fails the graph theory-based FT and is classified as a special asset. The loss of 26-30 would saturate the limiting critical cut-set K_{crit}^0 by a margin of -77 MW, i.e., T_l^0= -77 MW.
2) *1st Outage*: When 15-33 is lost, no additional special assets are identified.
3) *2nd Outage*: When 19-34 is lost, no additional special assets are identified.
4) *3rd Outage*: When 37-38 is lost, the asset 42-49 fails the FT and is classified as a special asset. The loss of 42-49 would saturate the limiting critical cut-set K_{crit}^3 by a margin of -186 MW, i.e., T_l^3= -186 MW.
5) *4th Outage*: When 49-66 is lost, no additional special assets are identified.

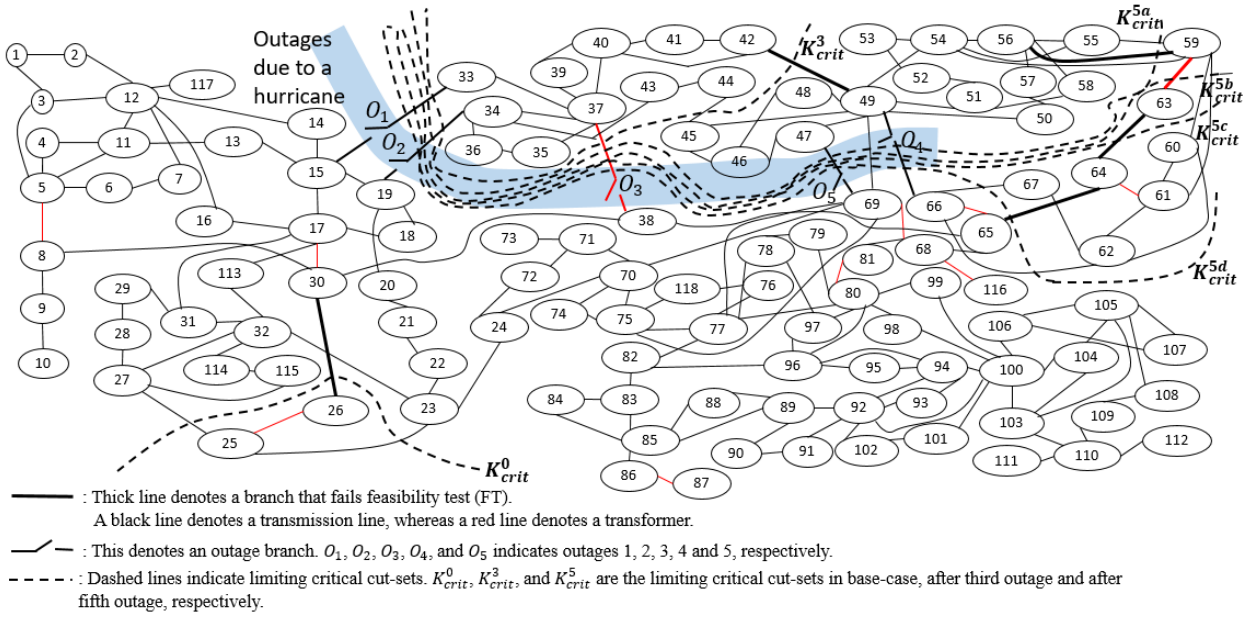

Figure 1.7 Detection of post-contingency cutset saturation on the IEEE 118 bus system

The value of the information obtained above can be realized by considering the following scenario: after the occurrence of the fifth outage, the proposed FT algorithm would inform the power system operator that if any of the four assets identified in the last row, second column of Table II is lost next (as the 6th outage), the corresponding cut-set identified in the third column would be saturated by the margin mentioned in the fourth column. The operator must preemptively reduce the power flowing through the identified cut-set by at least the amount mentioned in the fourth column to avoid the anticipated overload. Thus, the proposed network analysis tool is an enhanced power system connectivity monitoring scheme that improves the power system operators' situational awareness by augmenting their visualization in real-time.

To validate the severity of the contingencies (special assets) identified by the FT algorithm we have performed an independent cascading failure simulation using MATCASC [12]. MATCASC is a software package linked with MATPOWER that facilitates the simulation of cascading failures for any initiating contingency. The amount of load shed at the end of the cascade (shown in the sixth column of Table II) indicates the severity of the contingency. It can be observed that all the critical contingencies identified by the cascading failure analysis (that results in some load shedding) have also been identified by the graph-theory based FT analysis (compare the second and fifth columns of Table II).

Table 1.2 Performance comparison of Feasibility Test (FT) algorithm

Event	Feasibility Test (FT) Analysis			Cascading failure analysis by MATCASC	
	New special asset	Limiting critical cut-set	Transfer margin (MW)	Critical Contingency	Load shed (%)
Base-case	26-30	K_{crit}^{0} = {26-30,25-27,25-23}	T_l^0=-77	26-30	12.20%
Outage 1 (15-33)	-	-	-	-	-
Outage 2 (19-34)	-	-	-	-	-
Outage 3 (37-38)	42-49	K_{crit}^{3}= {42-49,44-45}	T_l^3= -186	42-49	29.87%
Outage 4 (49-66)	-	-	-	-	-
Outage 5 (47-69)	59-56	K_{crit}^{5a}= {59-56,59-54,59-55,69-49}	T_l^{5a}= -64	59-56	25.27%
	63-59	K_{crit}^{5b}= {63-59,61-59,60-59,69-49}	T_l^{5b}= -191	63-59	28.68%
	63-64	K_{crit}^{5c}= {63-64,61-59,60-59,69-49}	T_l^{5c}= -191	63-64	28.26%
	64-65	K_{crit}^{5d}= {64-65,66-62,66-67,69-49}	T_l^{5d}= -219	64-65	28.92%

1.4 Mitigation of Saturated Cut-sets

After detection of saturated cut-sets, their quick alleviation is the next requirement. Towards this end, a two-component methodology is proposed. The first component is named integrated corrective action (iCA) and it combines the proposed feasibility test with RTCA. This helps the system to be secure against post-contingency cut-set saturation and critical branch overloads. The second component is named relaxed corrective action (rCA) and it uses the feasibility test results to secure the system against post contingency cut-set saturation alone. While rCA is more computationally efficient, iCA is more comprehensive. Implementing both in parallel, with rCA being used only when iCA cannot provide a solution before next redispatch occurs, enhances the system security significantly in comparison to only relying on RTCA all the time.

1.4.1 Integrated corrective action (iCA)

Post contingency cut-set saturation as well as critical branch overloads are a big concern for power system operators as they can lead to successive contingencies and compromise system security to a great extent. iCA is designed by combining the results from FT and RTCA. iCA ensures the critical contingencies detected by RTCA do not create post-contingency branch overloads and the special assets identified by FT do not create saturated cut-sets, while also finding a least cost redispatch solution. Generation redispatch and load-shedding are the two actions to mitigate all the identified overloads. Since load-shedding has higher socio-economic costs, it is not a preferred action. However, during multiple contingency scenarios, generation redispatch alone might not be able to mitigate all identified overloads and load-shedding should be used as a last resort. The mathematical formulation and derivation of iCA is explained below.

Consider that the generator at bus $i \in G$ in the system is associated with a quadratic cost curve as shown below:

$$F_i(G_i) = a_i + b_i G_i + c_i G_i^2 \tag{2}$$

where, G_i is the power produced (in MW) by the generator at bus i, and a_i, b_i, and c_i are the fixed cost coefficient (in \$), the linear cost coefficient (in \$/MW), and the quadratic cost coefficient (in \$/MW2), respectively, for the corresponding generator. Let G_i^o and G_i^n denote the power produced before and after the new dispatch. The change in generation cost as a function of change in power generation, $\Delta G_i (= G_i^n - G_i^o)$, is given by,

$$\begin{aligned} \Delta F_i(\Delta G_i) &= \{a_i + b_i G_i^n + c_i (G_i^n)^2\} - \{a_i + b_i G_i^o + c_i (G_i^o)^2\} \\ &= c_i \Delta G_i^2 + d_i \Delta G_i \end{aligned} \tag{3}$$

where, $d_i = (2c_i G_i^o + b_i)$. Now, the cost of shedding the load at bus $j \in L$ can be written as:

$$\Delta F_j(\Delta L_j) = m_j \Delta L_j \tag{4}$$

where, ΔL_j denotes the amount of load-shed, and m_j is the cost coefficient of load-shed (in \$/MW); m_j is chosen to be significantly higher compared to the generator cost coefficients because the goal is to use load-shed only when generation redispatch alone cannot mitigate all violations. The optimization problem that minimizes the total cost of change in generation and load-shed can now be written as:

$$\text{Minimize: } \sum_{i \in \boldsymbol{G}} (c_i \Delta G_i^2 + d_i \Delta G_i) + \sum_{j \in \boldsymbol{L}} (m_j \Delta L_j) \tag{5}$$

While this optimization problem can find the minimum cost for the total change in generation and load-shed, there are some constraints that should be satisfied to ensure practicality. The first constraint is the branch power flow constraint. This ensures the power flow is within the minimum and maximum flow limits. Let the power transfer distribution factor (PTDF), $PTDF_{l,i}$, denote the change in flow in branch e_l for one unit of power added at bus i and one unit of power withdrawn

from the reference bus of the system. Then, the change in flow, Δf_l, through e_l for the change in bus power injections can be obtained as follows:

$$\Delta f_l = \sum_{\forall i \in \boldsymbol{G}} PTDF_{l,i}\, \Delta G_i - \sum_{\forall j \in \boldsymbol{L}} PTDF_{l,j}\, \Delta L_j \tag{6}$$

Defining f_l^o, f_l^{max} and f_l^{min} to be the original power flow, maximum and minimum power flow limits respectively, the branch power flow constraint can be expressed as

$$f_l^{min} - f_l^0 \leq \sum_{\forall i \in \boldsymbol{G}} PTDF_{l,i} \Delta G_i - \sum_{\forall j \in \boldsymbol{L}} PTDF_{l,j} \Delta L_j \leq f_l^{max} - f_l^0, \ \forall e_l \in \boldsymbol{E} \tag{7}$$

In (7), $\boldsymbol{E}$ denotes the set of all edges of the power system network. The maximum and minimum power generation constraint is given as follows:

$$G_i^{min} - G_i^0 \leq \ \Delta G_i \leq G_i^{max} - G_i^0 , \quad \forall i \in \boldsymbol{G} \tag{8}$$

where, G_i^o, G_i^{max}, and G_i^{min} denote the original power produced, maximum power and minimum power that can be produced by the generator at bus i, respectively. Similarly, the constraints for minimum and maximum power demand at a load bus j is given as follows:

$$L_j^{min} - L_j^0 \leq \Delta L_j \leq L_j^{max} - L_j^0 \tag{9}$$

The post-contingency branch flow constraints can be efficiently modeled using the line outage distribution factors (LODFs). Let $\text{LODF}_{l,k}$ represent the change in flow through branch e_k that will appear on branch e_l for an outage of branch e_k. The post-contingency flow, f_l^c, through e_l for a potential outage of e_k is given as follows:

$$f_l^c = f_l^n + LODF_{l,k} f_k^n \tag{10}$$

where, f_l^n and f_k^n denote the new flows corresponding to the new redispatch solution through branches e_l and e_k, respectively. Now, (10) can be re-written as:

$$f_l^c = (f_l^0 + \Delta f_l) + LODF_{l,k}(f_k^0 + \Delta f_k) \tag{11}$$

where, f_l^0 and f_k^0 denote the original flows through branches e_l and e_k, respectively, and Δf_l and Δf_k represent the incremental change in flows through branches e_l and e_k as obtained from the redispatch. Substituting Δf_l and Δf_k from (6) into (11), and using the respective branch flow limits, we obtain the following equations for post-contingency branch flow constraints:

$$f_l^{min} - \left(f_l^0 + LODF_{l,k} f_k^0\right) \leq \sum_{\forall i \in \boldsymbol{G}} \left(PTDF_{l,i} + LODF_{l,k} PTDF_{k,i}\right) \Delta G_i -$$

$$\sum_{\forall j \in \boldsymbol{L}} \left(PTDF_{l,j} + \ LODF_{l,k} PTDF_{k,j}\right) \Delta L_j \leq f_l^{max} -$$

$$-\left(f_l^0 + LODF_{l,k} f_k^0\right), \quad \forall e_k \in \boldsymbol{E}_v \, , \forall e_l \in \boldsymbol{E} \tag{12}$$

where, $\boldsymbol{E}_v$ contains the critical contingencies detected by RTCA. Equation (12) is modeled for all post-contingency overloads for the critical contingencies detected by RTCA.

For the special assets detected by the FT algorithm, (13) is formulated.

$$\sum_{\forall e_l \in \boldsymbol{K}_{crit}} \Delta f_l \leq T_m, \tag{13}$$

Note that (13) reduces the total power transfer across the limiting critical cut-set $\boldsymbol{K}_{crit}$ by the respective transfer margin, T_m. Now substituting Δf_l from (6) into (13), the constraint for cut-set power transfer is obtained as follows:

$$\sum_{\forall i \in \boldsymbol{G}} \left(\sum_{\forall e_l \in \boldsymbol{K}_{crit}} PTDF_{l,i} \right) \Delta G_i -$$

$$\sum_{\forall j \in \boldsymbol{L}} \left(\sum_{\forall e_l \in \boldsymbol{K}_{crit}} PTDF_{l,i} \right) \Delta L_j \leq T_m, \forall\, \boldsymbol{K}_{crit} \in \boldsymbol{\mathcal{K}}_{crit} \tag{14}$$

where, the set $\boldsymbol{\mathcal{K}}_{crit}$ contains the limiting critical cut-sets detected by the FT corresponding to different special assets.

Note that the difference between iCA and the traditional security constrained economic dispatch (SCED) is the additional cut-set power transfer constraint described by (14). Thus, iCA creates a more comprehensive corrective action than SCED as the iCA considers both post-contingency cut-set saturation and post-contingency branch overloads.

1.4.2 Relaxed corrective action (rCA)

A second component is proposed to provide a high-speed corrective action. This utilizes the results from FT to create a relaxed corrective action (rCA) as shown in Figure 8. The rCA solves the same optimization problem (given by (5)), but *without modeling the post-contingency branch flow constraints* (described by (12)). However, the cut-set power transfer constraints, described by (14), are retained in rCA, i.e., the rCA utilizes the results from FT to only secure the system against post-contingency cut-set saturation. Note that the optimization problem given by (5) can reduce to an optimal power flow (OPF) problem if it is solved without modeling any security constraints (neither (12) nor (14)). Therefore, by considering (14), the rCA adds a relaxed criterion of power system security onto an OPF problem.

If the set E_s contains the special assets detected by FT, the number of cut-set power transfer equations modeled by the rCA is $|E_s|$. Now, as the number of cut-set violations identified will be smaller than the total number of branches of a power system, $|E_s| \ll |E|$, and consequently, $|E_s| \ll |E_v| \times |E|$. *This implies that the number of security constraints modeled by the rCA is significantly less compared to the number of security constraints modeled by the iCA (or SCED)*. This is the primary reason for the very high speed of rCA.

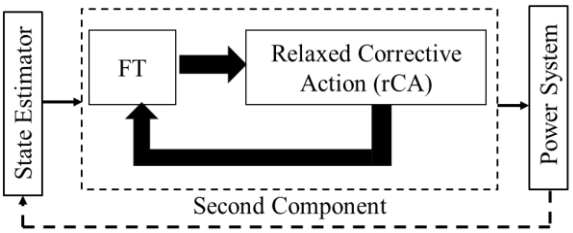

Figure 1.8 The second component - The results from FT are only utilized to create a relaxed corrective action (rCA)

The priority of usage of the first (iCA) and the second (rCA) components can be explained using Figure 9. Let an outage occur at time t_0. Both first and second components should be initiated simultaneously but independently.

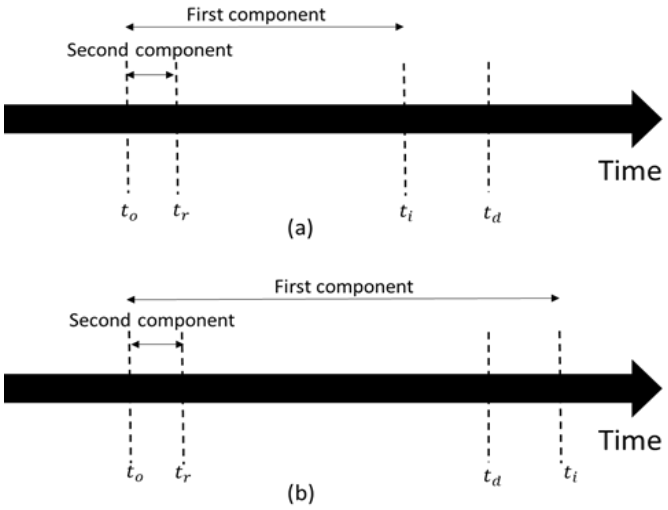

Figure 1.9 (a) If the first component provides a dispatch solution before the scheduled time for the next redispatch, then the solution obtained from the first component should be implemented. (b) If the first component does not provide a dispatch solution before

Let t_i and t_r denote the time taken to compute the first and second component solutions. Moreover, let t_d denote the time at which the redispatch solution is implemented. Then as shown in Figure 9, if $t_i < t_d$, the solution obtained by the first component should be used since it has better quality. If $t_i > t_d$ and $t_r < t_d$, then the solution obtained from the second component should be implemented to at least secure the system against post-contingency cut-set saturation. Even for large power systems, the possibility of $t_r > t_d$ is small. However, if that happens the solution from the first (preferred) or the second component should be implemented in the next redispatch depending upon their availability.

1.4.3 Performance results

The performance of the first and second components are evaluated in this subsection. The performance of the proposed two-component methodology against traditional approaches, such as RTCA-SCED or DC-OPF, is verified using a detailed case-study that involves a sequence of six outages. All simulations were done in MATLAB. GUROBI was used to solve the optimization problems. Table III compares the performance of the first component with the RTCA-SCED framework when six outages manifest successively in the IEEE 118-bus test system. The second column shows the special assets detected by the FT. An outage of any of these special assets (after the outage that has already occurred in the corresponding row of the first column), will create post-contingency cut-set saturation. The third column shows the critical contingencies detected by RTCA that result in post-contingency branch overloads. A two-step procedure was used to determine the entries of the third column. First, PTDFs and line ratings were used to rank the contingencies following every outage. Subsequently, top 30% of the contingencies were evaluated by RTCA to determine the post-contingency branch overloads.

Table 1.3 Comparative analysis of the first component and RTCA-SCED for a sequence of six outages in the IEEE 118-bus test system

Event (branch outages)	**First component (FT-RTCA-iCA)**					**RTCA-SCED**			
	FT	**RTCA**	**MATCASC (before correction)**	**Gen. Cost (k$)**	**MATCASC (after correction)**	**RTCA**	**MATCASC (before correction)**	**Gen. Cost (k$)**	**MATCASC (after correction)**
Outage 1: 15-33	-	-	-	126.2	-	-	-	126.2	-
Outage 2: 19-34	-	5-8	-	126.3	-	5-8		126.3	-
Outage 3: 37-38	42-49	42-49, 5-8, 26-30	42-49	126.5	-	42-49 5-8, 26-30	42-49	126.5	-
Outage 4: 42-49	45-46, 45-49	45-46, 45-49	45-46, 45-49	126.7	-	45-46, 45-49	45-46, 45-49	126.7	-
Outage 5: 49-66	-	5-8	-	126.7	-	5-8	-	126.7	-
Outage 6: 66-67	64-65, 65-66	64-65	64-65, 65-66	127.1	-	64-65	64-65, 65-66	126.9	65-66

An independent cascading simulation analysis was conducted using MATCASC [12], a software package that evaluates the consequence of cascading failures in power systems. Every outage was evaluated by MATCASC to screen out outages that will trigger a cascade and result in unserved power demand. Cascade triggering contingencies detected by MATCASC *before* and *after* the implementation of iCA are shown in columns 4 and 6 of Table III. The fifth column presents the

redispatch solution (generation cost) obtained from the iCA. The solution obtained from iCA does not contain any cascade triggering contingencies is shown in column 6.

The results of the first component and RTCA-SCED are identical for the first five outages. This is because for these outages the FT does not identify additional violations to those already detected by RTCA (compare the second and third columns of Table III). However, after the sixth outage FT detects the special asset 65-66 in addition to the critical contingency 64-65 identified by RTCA (see second and third column of the last row). This becomes the basis for the difference in the redispatch solutions of the first component and RTCA-SCED as seen in the fifth and ninth columns of the last row of Table III. Finally, it is observed that the RTCA-SCED solution contains one cascade triggering contingency (65-66), while the solution obtained from iCA did not have any (compare the sixth and the tenth columns of the last row of Table III). *This observation proves that integrating the results from FT with RTCA enhances the ability of power system security assessment in mitigating the risk of cascade triggering contingencies*.

Now, there could be situations when the first component takes a long time to generate a solution, in which case the second component should be utilized as shown in Figure 9. Table IV presents the application of the second component and compares it with a simple DC-OPF. Note that it is fair to compare the second component with a DC-OPF instead of an AC-OPF because the DC-OPF solves a linearized constrained optimization problem (like rCA used in the second component) while the optimization problem solved in AC-OPF is non-linear. Moreover, the focus here is on high-speed, and it is well-known that for any given system, a DC-OPF problem can be solved much faster than an AC-OPF problem.

The first column of Table IV lists the sequence of events. Columns two through five present the results of the second component. Note that only the FT results are shown in this section as the RTCA results are not considered in the second component. Cascading analysis done after the corrective action indicates that the redispatch obtained from rCA does not contain any cascade triggering contingency for the first five consecutive outages (see fifth column of Table IV). However, after the sixth outage, two cascade triggering contingencies manifest before the corrective action is initiated (see last row, third column of Table IV), of which, only one is addressed by rCA. That is, the solution obtained using the rCA still contains one cascade triggering contingency (see last row, fifth column of Table IV). This happened because the contingency 64-65 triggered cascading failures due to branch overloads, even after the rCA alleviated all post-contingency cut-set saturation.

However, the second component performs significantly better than a DC-OPF (see columns six and seven of Table IV). The sixth column presents the DC-OPF redispatch solution, while the seventh column presents the results of the cascading analysis by MATCASC on the redispatch solution. Since a DC-OPF does not model any security constraints, the number of cascade triggering contingencies in its solution is significantly high in comparison to the one obtained using rCA (in the second component). *This shows that in situations when the first component takes a long time to generate a solution due to heavy computational burden, the second component should*

be used to secure the system against post-contingency cut-set saturation, and thereby reduce the risk of cascading failures.

Table 1.4 Comparative analysis of the second component and DC-OPF for a sequence of six outages in the IEEE 118-bus test system

Event (Branch outages)	**Second component (FT-rCA)**				**DC-OPF**	
	FT	**MATCASC (before correction)**	**Gen. Cost (k$)**	**MATCASC (after correction)**	**Gen. Cost (k$)**	**MATCASC**
Outage 1: 15-33	-	-	126.2	-	125.9	26-30
Outage 2: 19-34	-	-	126.2	-	125.9	26-30
Outage 3: 37-38	42-49	42-49	126.3	-	125.9	26-30, 42-49
Outage 4: 42-49	45-46, 45-49	45-46, 45-49	126.4	-	126.2	26-30, 45-46 42-49
Outage 5: 49-66	-	-	126.4	-	126.2	26-30, 45-46 45-49
Outage 6: 66-67	64-65, 65-66	64-65, 65-66	126.7	64-65	126.2	26-30, 45-46, 45-49, 64-65, 65-66

1.5 Conclusions

A methodology for detecting if a contingency will create a saturated cut-set, namely, the feasibility test (FT) algorithm, is proposed. This is important particularly in the presence of extreme weather events where successive contingencies can take place leading to uncontrolled islanding of the power system. For generating inputs to the FT algorithm, data-driven methods for time synchronized bus power injections using deep neural networks and support vector regression are developed. Results show that even in presence of PMUs at only the highest voltage buses of the network, both the data driven methods can estimate the time synchronized bus power injections of all the buses in the network with sufficient accuracy. Subsequently, a two-component methodology that enhances the scope and speed, respectively, of static power system security assessment during multiple outage scenarios is developed. The first component of the proposed methodology combines the results from the FT algorithm and RTCA to create an integrated corrective action (iCA). The iCA initiates a comprehensive response to the violations detected by FT and RTCA to protect the system against saturated cut-sets as well as critical branch overloads. The second component of the proposed methodology presents an alternative method that complements real-time power system operations during extreme event scenarios, when detailed network analysis tools such as the first component or traditional RTCA-SCED take longer to generate a solution. Under such circumstances, by only employing the FT algorithm, a relaxed corrective action (rCA) is implemented that quickly secures the system against post-contingency cut-set saturation.

2. Real Time Event Identification Based on Modal Analysis of Phasor Measurement Unit Data

2.1 Introduction

Power systems are prone to a variety of events (e.g., line trips and generation loss) and real-time identification of such events is crucial in terms of situational awareness, reliability, and security of the system. 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. This makes real-time identification of events a challenge.

Prior research in the context of real-time identification of events in power grids can be categorized into two main approaches, namely model-based and data-driven. Model-based methods (see e.g., [13]-[15]) involve modeling of power system components and estimation of the system states. The performance of such methods highly depends on the accuracy of dynamic models and estimated states of the system, which in turn limits their implementation for real world problems. Data-driven methods have begun receiving increased attention, mainly due to the growing penetration of phasor measurement units (PMUs) in the electric grid which can help address situational awareness challenges. Such methods can be further classified into two subcategories: (i) well-studied physics-based signal processing methods such as modal analysis for feature extraction that are directly applied to PMU measurements to detect events [16]-[18], and (ii) purely data-driven classification methods using PMU measurements which has begun to recently gain traction [19][20]. In this study, we develop a technique that seeks to take the best of both worlds: we take advantage of knowledge of the physics of the system by characterizing event signatures based on their modal information that can be directly extracted from PMU measurements, and subsequently apply ML techniques to produce a robust classifier from limited but feature-rich training data. The overview of the proposed approach is shown in Fig. 10. It has three main steps:

Step 1) The first step is to identify the precise physics-based method that can help detect a set of delineating features for different events using spatio-temporally correlated PMU data from multiple units; in turn, this will help characterize events based on a set of features obtained from modal analysis of various PMU measurements.
Step 2) determining an optimal subset of features that succinctly describes the system dynamics using all the features extracted from step 1 above, and
Step 3) designing a set of learning models that can be classified using the chosen optimal subset of features from step 2 above.

In step (1), for each set of PMU measurements (positive sequence magnitude and angle of voltages and currents, frequency, and rate of change of frequency) obtained from multiple PMUs, techniques such as multi-signal matrix pencil (MSMP) can be applied to find a single set of modes that best represent the underlying dynamical behavior of the system. Based on this approach, power system events can be described as a set of features, e.g., via angular frequencies, damping values and the corresponding residue magnitudes. 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. Hence in step (2) we will test different data-driven filter methods to choose the best subset of features. Our goal here is to avoid overfitting or underfitting while ensuring that multiple events can be distinguished by the same set of sufficient features. Finally, using the extracted features via steps (1) and (2), in step (3) we investigate the performance of two well-known classification algorithms, namely, logistic regression and support vector machine on two different datasets, one obtained from simulated generation loss and line trip events in Texas 2000-bus synthetic grid using PSSE software and the other is a proprietary dataset with labeled generation loss and line trip events obtained from a large utility in the US involving measurements from nearly 500 PMUs.

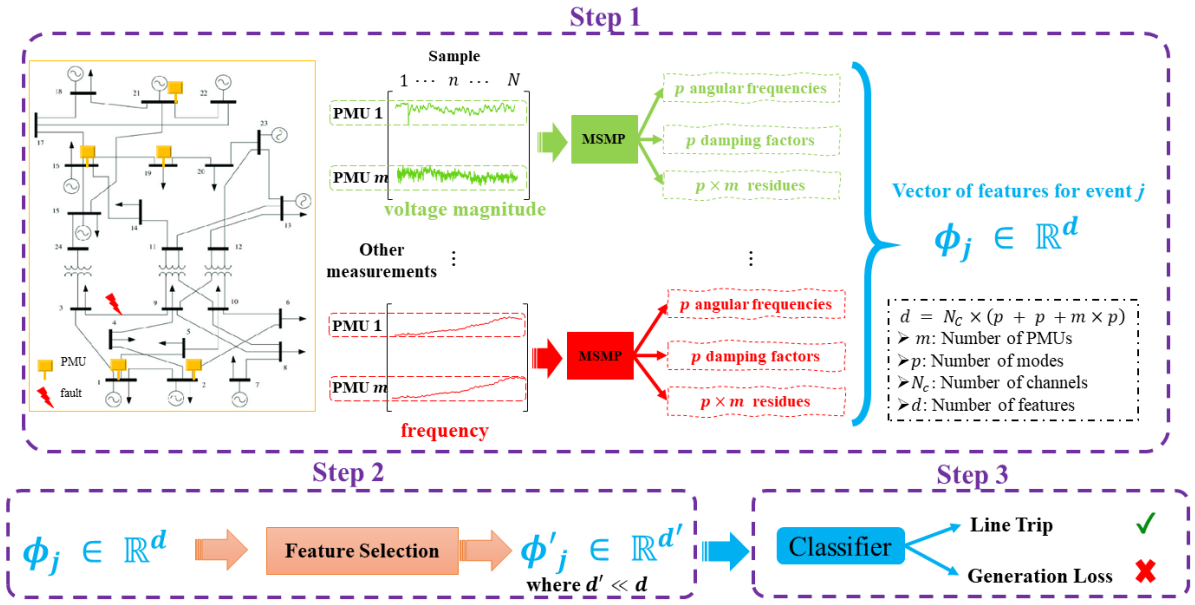

Figure 2.1 Overview of the proposed approach.

2.2 Problem statements

The first step in identifying a system event from PMU data is to extract the relevant features from the data stream. Because the PMU data-rate is so high, particularly if multiple PMUs simultaneously observe the same event, it would be overly naive to plug the raw data into a machine learning model. Rather, it is advantageous to use domain knowledge to identify features in the data streams that are likely to contain information regarding the class of the event. 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. Specifically, we assume that the PMU data streams after an event consist of a superposition of several 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.

Consider a power grid with m installed PMUs. In general, 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., positive sequence voltage magnitude) to describe the modal representation of PMU measurements. Let $y_i(n) \in \mathbb{R}, i = 1,\dots,m$, and $n = 0,\dots,N-1$ denotes the positive sequence voltage magnitude (VPm) measurement obtained from the i^{th} PMU at sample n with a sampling period of T_s. We assume that each PMU measurement signal, $y_i(n)$, after an event consists of a superposition of several dynamic modes. It is well known that in a large electric grid, multiple PMUs can capture the dynamic response of the system after an event [27]. However, in case of individual modal analysis of PMU measurements, we obtain varying modal estimates due to the noise and nonlinear inherent power system dynamics [28]. Hence, we aim to analyze multiple measured signals simultaneously to obtain one optimum set of mode estimates which can accurately represent the underlying dynamic behavior of the system. Thus, each PMU measurement signal, $y_i(n)$, can be represented as a superposition of p damped sinusoidal modes as follows:

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

where

$$Z_k = e^{\lambda_k T_s}, \qquad \lambda_k = \sigma_k \pm j\omega_k \tag{2}$$

and $\epsilon_i(n)$ represents the noise in the i^{th} PMU signal, Z_k is the k^{th} mode associated with each PMU signal and σ_k and ω_k are its corresponding damping factor and angular frequency, respectively. Furthermore, residue $R_k^{(i)}$ corresponding to each mode k and i^{th} PMU signal is defined by its magnitude $|R|_k^{(i)}$ and angle $\theta_k^{(i)}$.

Using estimated modes, $\{Z_k\}_{k=1}^{p}$, the corresponding residues $\{R_k^{(i)}\}_{k=1}^{p}$ for each signal $i = 1,\dots,m$ are obtained by solving

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ Z_1 & Z_2 & \cdots & Z_p \\ \vdots & \vdots & \ddots & \vdots \\ Z_1^{N-1} & Z_2^{N-1} & \cdots & Z_p^{N-1} \end{bmatrix} \begin{bmatrix} R_1^{(i)} \\ R_2^{(i)} \\ \vdots \\ R_p^{(i)} \end{bmatrix} = \begin{bmatrix} y_i(0) \\ y_i(1) \\ \vdots \\ y_i(N-1) \end{bmatrix} \tag{3}$$

Note that the idea here is that the parameters of (1) are the key to capturing signatures of an event, and hence are used as the features to distinguish between various types of events. For instance, using the positive sequence voltage magnitude measurements, we would obtain a set of features, $\boldsymbol{\mathcal{F}}_{\text{VPm}}$, as follows:

$$\begin{aligned} \boldsymbol{\mathcal{F}}_{\text{VPm}} = [&\{\omega_k : k = 1,\dots,p\}, \{\sigma_k : k = 1,\dots,p\}, \\ &\{|R|_k^{(i)} : i = 1,\dots,m, k = 1,\dots,p\}, \{\theta_k^{(i)} : i = 1,\dots,m, k = 1,\dots,p\}] \end{aligned} \tag{4}$$

which consists of p angular frequencies, p damping values and corresponding residues magnitude and angle for each PMU $i = 1, \ldots, m$, and mode $k = 1, \ldots, p$. A crucial point worth noting here is that to make the features comparable across the events, we must sort the features in a meaningful manner. Since the residue coefficients indicate the quantity of each mode present in each PMU data stream, one can sort the modes based on their average residue across all the PMUs. So, in our notation in (4), $k = 1, \ldots, p$, represent the sorted modes based on their average residue in a descending order. For instance, $k = 1$ always represents the mode with the largest average residue. Moreover, residue magnitudes corresponding to each mode k, are the sorted values of $|R|_k^{(i)}$, $i = 1, \ldots, m$ in descending order and we use the same order to sort the corresponding $\theta_k^{(i)}$.

Similarly, using other types of measurement, i.e., positive sequence voltage angle (VPa), positive sequence current magnitude (IPm), and corresponding angle (IPa), frequency (F), and rate of change of frequency (DF), we would obtain the corresponding set of features, denoted as $\boldsymbol{\mathcal{F}}_{\text{VPm}}$, $\boldsymbol{\mathcal{F}}_{\text{VPa}}$, $\boldsymbol{\mathcal{F}}_{\text{IPm}}$, $\boldsymbol{\mathcal{F}}_{\text{IPa}}$, $\boldsymbol{\mathcal{F}}_{\text{F}}$, $\boldsymbol{\mathcal{F}}_{\text{DF}}$, respectively.

2.2.1 Multi Signal Matrix Pencil method

Various modal analysis techniques have been used in power systems to estimate the parameters of (1). It is now well understood that matrix pencil method (MPM) is more robust to noise relative to other similar techniques such as Prony analysis and dynamic mode decomposition (see, for example, [21],[22], and [23]) and, hence, will be the main modal analysis approach we use. A brief overview of the matrix pencil method is presented in this section (we refer readers to [24], [25], and [26] for a comprehensive study of the multi-signal matrix pencil method in the presence of noise in the signal).

Matrix pencil method involves constructing the Hankel matrix over a block of N samples obtained from the i^{th} PMU as

$$\mathcal{H}_i = \underbrace{\begin{bmatrix} y_i(0) & y_i(1) & \cdots & y_i(L) \\ y_i(1) & y_i(2) & \cdots & y_i(L+1) \\ \vdots & \vdots & \ddots & \vdots \\ y_i(N-L-1) & y_i(N-L) & \cdots & y_i(N-1) \end{bmatrix}}_{(N-L)\times(L+1)} \tag{5}$$

where L is the pencil parameter. It is well studied that choosing $L = N/2$ will result in the best performance of the matrix pencil method in a noisy environment (close to Cramer-Rao bound) [12]. We define $\mathcal{H}_i^{(1)}$as the matrix consisting of first L columns of $\mathcal{H}_i$ and $\mathcal{H}_i^{(2)}$ as the matrix comprised of its last L columns. Then the parameters $Z_1, \ldots, Z_p$ are the generalized eigenvalues of $(\mathcal{H}_i^{(1)})^{\dagger}\mathcal{H}_i^{(2)}$ [26].

The MPM method described above may be extended to find a single set of modes which best represent the underlying dynamical behavior of a set of measurements. 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 (this matrix is denoted as **H**) and the same method as the original MPM for a single signal can be applied to the **H** matrix to identify a single set of modes $Z_1, \ldots, Z_p$.

Following the assumption that PMU measurements after an event can be represented as a superposition of p dynamic modes and because only a small number of modes are enough to represent the underlying dynamical behavior of the system ($p \ll L$), one can show that rank(**H**)= p for a noise free signal [26]. This implies that, for noisy PMU measurements, the number of significant singular values obtained from singular value decomposition (SVD) of **H**, is associated with the number of dominant modes in the signal and the remaining singular values are linked to minor noisy variation in the signal. Hence, the best way to approximate p in (1) and remove small variations derived by the noise in the signal is to find the best rank $p$approximation of **H**, denoted as $\mathbf{H}_p$. The rank p approximation error is given by

$$E_p = \frac{\|\mathbf{H} - \mathbf{H}_p\|_F}{\|\mathbf{H}\|_F} \tag{6}$$

where $\|H\|_F$ is the Frobenius norm of the matrix **H**.

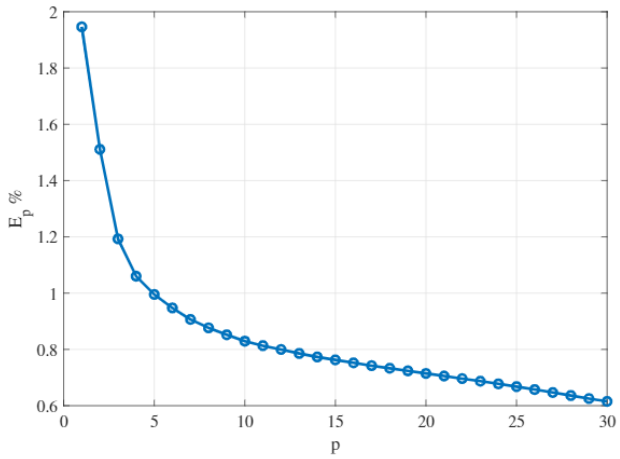

Figure 2.2 Rank p approximation error of the matrix H, for different values of p.

Figure 11 demonstrates the rank p approximation error of the matrix, **H**, which is constructed using the positive sequence voltage magnitude measurements (after a line trip event) from nearly 500 PMUs over a block of $N = 180$ samples with a pencil parameter of $L = 90$. Low rank approximation error of the **H** considering the first 6 largest singular values ($p = 6$), would be less than 1% for all the events in our dataset. Thus, throughout the report, we assume that $p = 6$.

2.3 Feature Engineering of PMU Time Series Data

To characterize the dynamic response of the power system after an event, modal analysis is conducted on each type of PMU measurement (i.e., VPm, VPa, IPm, IPa, F, and DF) obtained from multiple locations across the grid. Each event j can be described as a vector of features as

$$\phi_j = [\boldsymbol{\mathcal{F}}_{\text{VPm}}, \boldsymbol{\mathcal{F}}_{\text{VPa}}, \boldsymbol{\mathcal{F}}_{\text{IPm}}, \boldsymbol{\mathcal{F}}_{\text{IPa}}, \boldsymbol{\mathcal{F}}_{\text{F}}, \boldsymbol{\mathcal{F}}_{\text{DF}}]^{\text{T}} \tag{7}$$

where each $\boldsymbol{\mathcal{F}}_s$, $s \in$ {VPm, VPa, IPm, IPa, F, DF} consists of the modal analysis results corresponding to the selected type of PMU measurement (e.g., see (4) for $\boldsymbol{\mathcal{F}}_{\text{VPm}}$). However, using all the estimated parameters of (1) obtained from different types of measurement, will result in a high dimensional vector of features, ϕ_j. To avoid overfitting or underfitting while ensuring that multiple events can be distinguished by the same set of sufficient features, we propose a two-step approach to reduce features into a more manageable number. In the first step, we select a subset of features based on our domain knowledge of the system under study. The second step is to select the most informative and relevant features using a feature selection technique called filter method. The details are provided in the following subsections.

2.3.1 Feature selection based on the domain knowledge

As discussed in section 2.2, parameter p represents the number of dominant modes in the signal and can be obtained by finding the best rank p approximation of **H**. We also showed that $p = 6$ results in $E_p < 1\%$ for all the events in our dataset. In general, these modes are complex conjugate pairs (i.e., 3 complex conjugate pairs and 6 modes in total). However, for a given type of measurement, s, in order to remove redundant modal information present in the complex conjugate modes, we only include one of them in the corresponding vector of features $\boldsymbol{\mathcal{F}}_s$, $s \in$ {VPm, VPa, IPm, IPa, F, DF}. However, the problem which arises here is that for small portion of the events, the modal analysis may result in different combinations of real and complex conjugate modes and 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 stream, one can sort the modes based on their average residue across all the PMUs. Then, we choose $p' = 3$ modes with the largest average residues across all the PMUs where p' is the number of distinct (complex conjugate or real) modes that will be used in the vector of features for each event.

Moreover, among all the m installed PMUs in the grid, only a small portion of them ($m' < m$) with the largest residue magnitudes will be affected by the event, thereby capturing dynamic behavior of the system after a disturbance. Hence, just the residues of those PMUs are included in the vector of features.

2.3.2 Feature selection using filter method

As detailed above, each mode is captured by the angular frequency and damping factor and thus, for p' modes, there are $2 \times p'$ such variables; further, every PMU measurement when expressed via the modes has p' residues and thus for a total of m' out of m PMUs, the total number of residues (including their magnitude and angle) is $2 \times m'p'$. 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 $\phi_j = [\varphi_1, \ldots, \varphi_d]^{\mathrm{T}} \in \mathbb{R}^d$, where $d = 2 \times n_{ch} \times (p' + m'p')$and a label $\xi_j$which describes the type of an event (i.e., line trips and generation loss events are labeled as 0 and 1, respectively). We define our dataset as $\boldsymbol{\Phi} = [\phi_1, \ldots, \phi_j, \ldots, \phi_{N_e}]^{\mathrm{T}} \in \mathbb{R}^{N_e \times d}$ and $\boldsymbol{\Xi}$=$[\xi_1, \ldots, \xi_j, \ldots, \xi_{N_e}]^{\mathrm{T}} \in \{0,1\}^{N_e}$ where N_e is the total number of labeled events.

In general, a designer of the feature extraction methodology has multiple hyperparameters at their disposal including: (a) number of modes p' that suffices to delineate events; (b) number of PMUs m' out of a total of m in the system to generate features via modal analysis; and (c) number of channels n_{ch} per PMU to use. For instance, for $m' = 20$, $p' = 3$, and using $n_{ch} = 6$ channels at a PMU, we obtain a total of $d = 756$ features. While this may not seem many features, when the number of labeled events is small (e.g., 70 labeled events in our dataset) which is typically the case in practice, a nearly 750-dimensional feature set can be extremely large.

Moreover, high dimensionality of the feature space compared to the number of labeled samples available for training makes it impossible to learn the relative contribution of each feature precisely [29]. So, a necessary pre-processing step before using any classification algorithm is to select relevant and most informative features.

Common statistical approaches for selecting features include filter methods, wrapper methods and embedded methods [30]. Because filter methods, unlike wrapper and embedded methods, are independent from classification algorithms, they are computationally inexpensive and are more efficient for real time applications [31] and will be the main tool for features selection in this study which will be further explained in the following paragraphs.

Filter methods employ some measure of dependence between a feature and the type of event in the training dataset to rank the features and retain only top ranked features. Various statistical tests, including one-way analysis of variance (ANOVA) F-value test (F) [32], sure independence screening (S) [33], mutual information (M) [34], Pearson correlation (P) [29], and Kendall correlation (K) [35] are used to quantify the correlation between features and target variables.

Based on the correlation of each feature φ_i, $i = 1, \ldots, d$ and target variable $\boldsymbol{\Xi}$, we sort the features and then remove all the features except d' with highest correlation with the target variable. However, due to the small number of samples, we will rely on a well-known approach in machine learning: bootstrapping: this is a technique of sampling with replacement to create multiple datasets from the original dataset thereby attempting to select the most informative features with some statistical degree of confidence. Let us define $\pi_{\varphi_i}^{(b)}$ as the correlation measure of feature φ_i and target variable $\boldsymbol{\Xi}$ over the b^{th} bootstrap samples ($b = 1, \ldots, B$), i.e. $\pi_{\varphi_i}^{(b)} = \mathbf{F}_x(\boldsymbol{\varphi_i}, \boldsymbol{\Xi})$where x represents the type of the statistical test that is used to quantify the correlation. The details are shown in Fig. 12. We sort the features based on different correlation measures, $\pi_{\varphi_i}^{(b)} = \mathbf{F}_x(\boldsymbol{\varphi_i}, \boldsymbol{\Xi})$, where x represents the type of the statistical test (i.e., F-value test, sure independence screening, mutual information, Pearson correlation, and Kendall correlation), and $\pi_{\varphi_i}^{(b)}$ is the correlation measure of feature φ_i and target variable $\boldsymbol{\Xi}$ over the b^{th} bootstrap samples. We select d' feature with the highest average score over all the bootstrap samples.

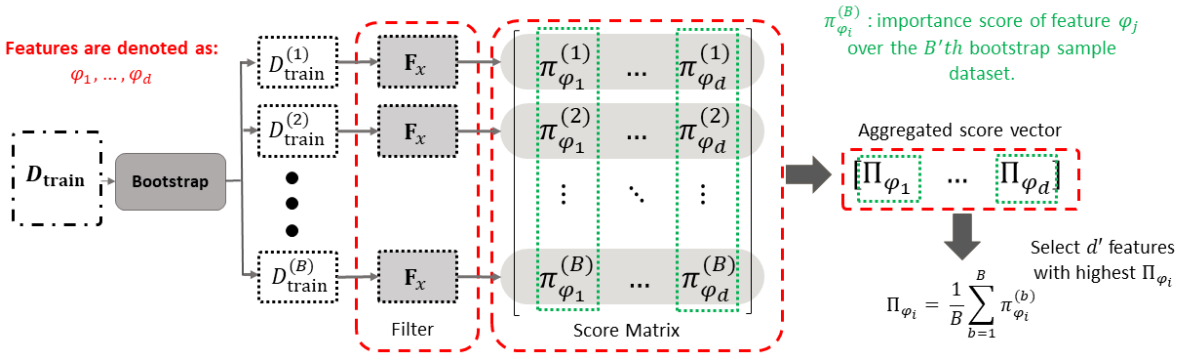

Figure 2.3 Overview of the feature selection step using filter method. D_{train}represents the training dataset.

2.4 Model Validation

The final step in our proposed framework for event identification is to use the optimal subset of features to learn a classification model by finding decision boundaries between various types of events in the feature space. A crucial point worth mentioning here is the choice of classifier that is best suited for the event identification task. With any machine learning model, there is a tradeoff inherent in the choice of complexity of the classification model. A simpler model may be more easily interpreted and are 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. Hence, we investigate the performance of two well-known classification algorithms: logistic regression and support vector machine on two different datasets, one obtained from simulated generation loss and line trip events in Texas 2000-bus synthetic grid using PSSE software and the other is a proprietary dataset with labeled generation loss and line trip events obtained from a large utility in the US involving measurements from nearly 500 PMUs.

The key steps to evaluate the performance of each classification algorithm is shown in Fig. 13. In order to validate the performance of each classification algorithm, we split the data set into a training and a test data and then all the feature engineering techniques are implemented on the training dataset to find the most relevant and informative subset of features. Due to the limited number of real labeled generation loss and line trip events, and for a fair evaluation of each classification algorithm, we generate ***B*** bootstrapped reduced order training datasets (with only an optimal subset of features) to learn a classification model. To evaluate the performance of the classification, we use the area under curve (AUC) of receiver operator characteristic (ROC) which illustrates the accuracy of a binary classifier based on various discrimination threshold. ROC plot shows the relation between true positive rate (also known as recall) against the false positive rate at various threshold settings. ROC AUC value is bounded between 0 and 1. The closer AUC to 1, the classifier has a better ability to classify the positive and negative samples whereas an uninformative classifier yields a 0.5 AUC score [36]. To quantify the accuracy of the learned classifier on the test dataset, we compute the average AUC, and 5%, and 95 % confidence interval of the AUC values over all the bootstrap samples.

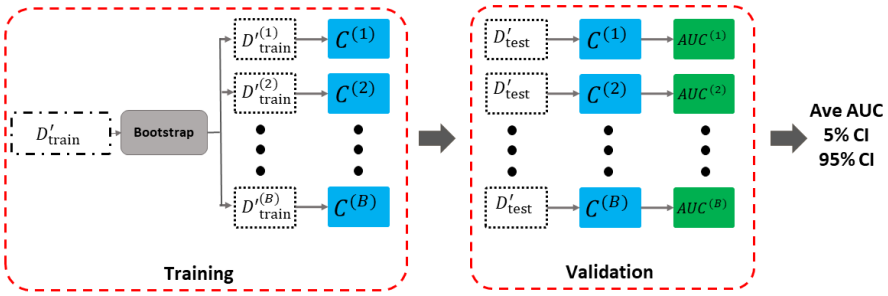

Figure 2.4 Evaluation of classification algorithms. For each chosen classifier, we will learn a model on each bootstrap dataset with an optimal subset of features and then use the learned model to predict target variables in the corresponding reduced order test *d*

2.5 Simulation results

Case study 1: identifying generation loss and line trip events using the generated synthetic event scenarios.

As the real data with event labels is often scarce, we find it better, as is the case with recent other studies, to generate synthetic data for training and testing of machine learning models. Synthetic data can never fully replace real data, but it can be a powerful tool in a variety of circumstances. To this end, we perform dynamic simulations of power system events using PSSE for the Texas 2000-bus synthetic grid to generate synthetic data streams from 95 PMUs across the grid. Total number of synthetic generation loss and line trip event scenarios are:

- Generation Loss: 400 (i.e., 200 in normal loading and 200 in 80% of normal loading)
- Line trip: 400 (i.e., 200 in normal loading and 200 in 80% of normal loading)
- 600 samples are used for training and 200 for test

Figure 14 shows the performance of the logistic regression classifier w.r.t. number of selected features in the generated synthetic dataset. We choose $d' = 10$ as the number of desired features obtained from the features selection step. Moreover, the average AUC score of the two classifiers (i.e., logistic regression and SVM with RBF kernel) as well its corresponding 5% and 95% confidence intervals are shown in Fig. 15. SVM with radial basis function (RBF) kernel has a slightly better performance than logistic regression in identifying the two types of the events in our synthetic dataset. In terms of performance of the models with respect to the type of filter method, apart from the sure independence screening method, using other measures of the correlations to find the optimal features result in a similar accuracy of the classifiers.

Case study (2): identifying generation loss and line trip events using a proprietary dataset with labeled generation loss and line trip events obtained from a large utility in the US involving measurements from nearly 500 PMUs.

Total number of 70 labeled events (23 generation loss and 47-line trip events) are used in this study. Total number of 59 events are included in the training dataset to obtain the most informative and relevant features and learn classification models from each bootstrapped dataset using only the selected features. The average AUC score of the two classifiers (i.e., logistic regression and

SVM with RBF kernel) as well its corresponding 5% and 95% confidence intervals are shown in Fig. 16. The main reason for a lower accuracy of classification algorithms in identifying the events in the real dataset compared with the generated synthetic dataset is the limited number of events (i.e., 70 labeled events over a period of three years).

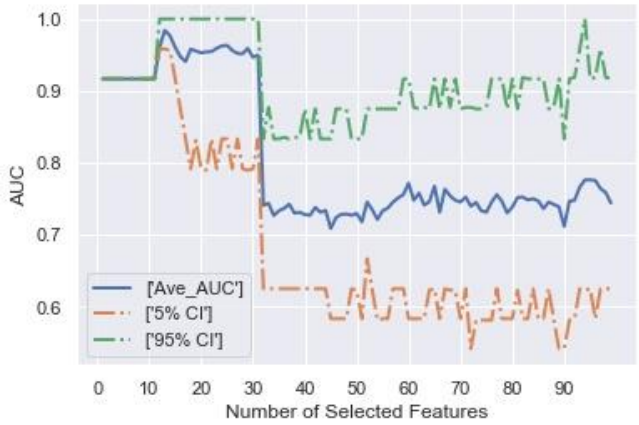

Figure 2.5 Performance of the logistic regression algorithm considering different number of selected features

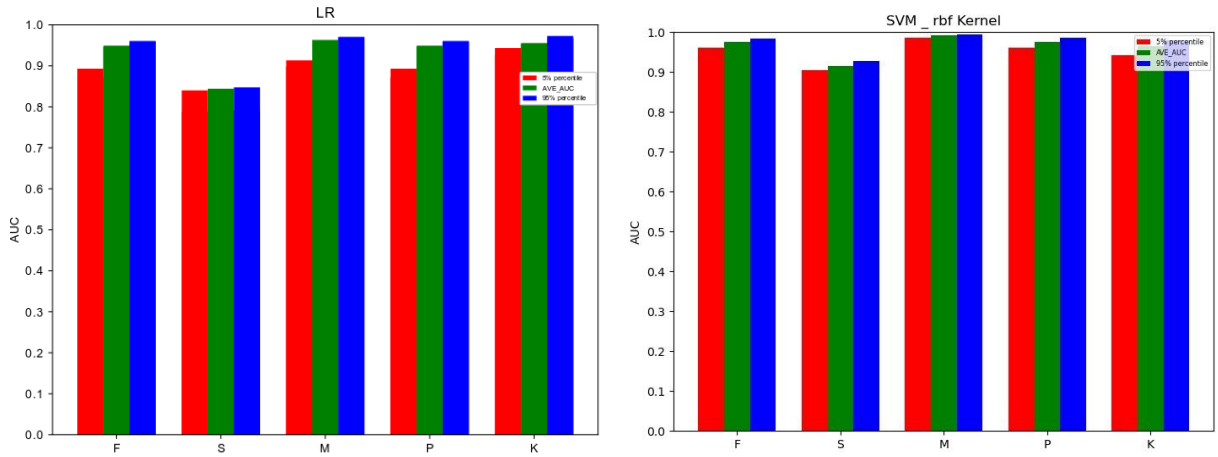

Figure 2.6 Performance of classification algorithms logistic regression (left), and SVM with rbf kernel (right) using the synthetic dataset with simulated generation loss and line trip events considering an optimal subset of 10 features. Note that F, S, M, P, and K represent one-way ANOVA F-value, Sure Independence Screening, Mutual Information, Pearson correlation, and Kendall correlation.

Performance of classification algorithms logistic regression (left), and SVM with rbf kernel (right) using the synthetic dataset with simulated generation loss and line trip events considering an optimal subset of 10 features. Note that F, S, M, P, and K represent one-way ANOVA F-value, Sure Independence Screening, Mutual Information, Pearson correlation, and Kendall correlation.

An interesting observation in both case studies is that angular frequency and first few residue magnitudes corresponding to the first mode of VPm, VPa and F signal are included in the set of optimal features obtained from various filter methods.

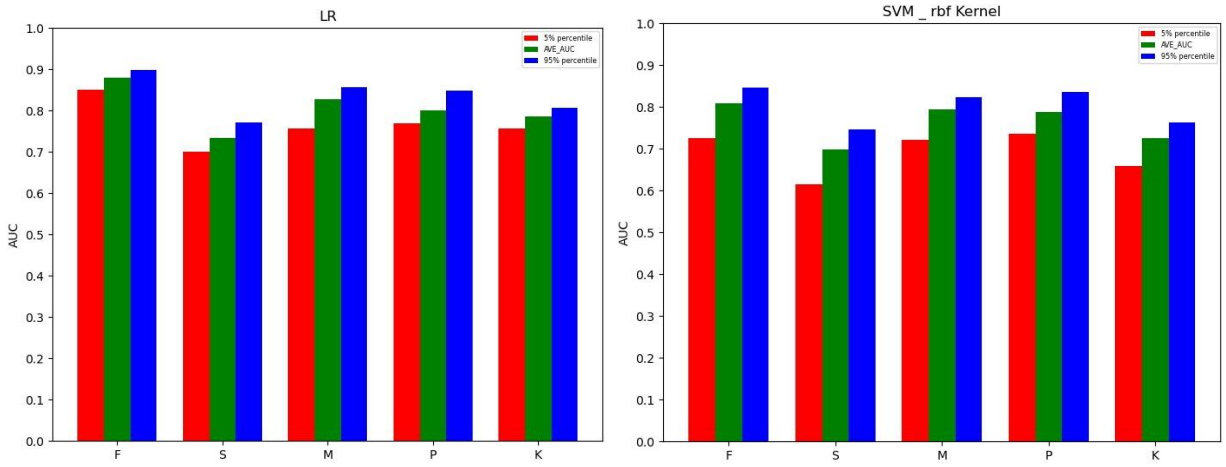

Figure 2.7 Performance of classification algorithms logistic regression (left), and SVM with rbf kernel (right) using the real PMU dataset with labeled generation loss and line trip events considering an optimal subset of 10 features. Note that F, S, M, P, and K represent one-way ANOVA F-value, Sure Independence Screening, Mutual Information, Pearson correlation, and Kendall correlation.

2.6 Conclusion

In this study, we proposed a novel framework for real time identification of events in the electric grid based on the modal analysis of various types of PMU measurements. Power system events are characterized by a set of features obtained from the modal analysis result of PMU measurements (positive sequence magnitude and angle of voltages and currents, frequency, and rate of change of frequency) from multiple locations in the grid where the features are the angular frequencies, damping values and the corresponding residue magnitudes. However, extracting features using all channels of PMU measurements across multiple PMUs will inevitably lead to a high-dimensional feature set, and thus, we use different data-driven filter methods to choose the best subset of features. Our key goal here is to avoid overfitting or underfitting while ensuring that multiple events can be distinguished by the same set of sufficient features. Finally, using the extracted features, the performance of two well-known classification algorithms, namely, logistic regression and support vector machine on two different datasets (one based on the synthetic events scenarios and the other from a proprietary dataset with labeled generation loss and line trip historical events) are compared. Based on the simulation result, both classifiers achieve similar classification accuracy. SVM with RBF Kernel slightly performs better in identifying the events in the synthetic real dataset whereas logistic regression is more accurate in identifying the events in the real dataset. Future work will focus on localizing events based on the PMU data by leveraging signal processing techniques based on spatial correlation of features.

References

[1] F. Galvan, S. Mandal, and M. Thomas, "Phasor Measurement Units (PMU) instrumental in detecting and managing the electrical island created in the aftermath of hurricane Gustav," in *Proc. IEEE Power Syst. Conf. Expos.* Seattle, WA, pp. 1-4, Mar. 2009.

[2] P. M. Chakalian, L. C. Kurtz, and D. M. Hondula, "After the lights go out: household resilience to electrical grid failure following hurricane Irma," *Nat. Hazards Rev.*, vol. 20, no. 4, pp. 1-14, Nov. 2019.

[3] S. Huang, and V. Dinavahi, "Real-time contingency analysis on massively parallel architectures with compensation method," *IEEE Access*, vol. 6, pp. 44519-44530, 2018.

[4] A. Abedi, L. Gaudard, and F. Romerio, "Review of major approaches to analyze vulnerability in power systems," *Reliability Eng. & System Safety*, vol. 183, pp. 153-172, Nov. 2018.

[5] R. Sen Biswas, A. Pal, T. Werho, and V. Vittal, "Fast identification of saturated cut-sets using graph search techniques," in *Proc. IEEE Power Eng. Soc. General Meeting*, Montreal, Canada, pp. 1-5, Aug. 2020.

[6] R. Sen Biswas, A. Pal, T. Werho, and V. Vittal, "A graph theoretic approach to power system vulnerability identification," *IEEE Trans. Power Syst.*, vol. 36, no. 2, pp. 923-935, Mar. 2021.

[7] R. Sen Biswas, A. Pal, T. Werho, and V. Vittal, "Mitigation of Saturated Cut-sets During Multiple Outages to Enhance Power System Security." *IEEE Transactions on Power Systems* (2021).

[8] A. Beiranvand, and P. Cuffe, "A Topological Sorting Approach to Identify Coherent Cut-Sets Within Power Grids," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 721-730, Jan. 2020.

[9] T. Werho, V. Vittal, S. Kolluri and S. M. Wong, "Power System Connectivity Monitoring Using a Graph Theory Network Flow Algorithm," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4945-4952, Nov. 2016.

[10] M. Ghamsari-Yazdel, M. Esmaili, F. Aminifar, P. Gupta, A. Pal, and H. A. Shayanfar, "Incorporation of controlled islanding scenarios and complex substations in optimal WAMS design," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3408-3416, Sep. 2019.

[11] L. Hanyue, et al. "Load modeling in synthetic electric grids," in Proc. 2018 IEEE Texas Power and Energy Conf. (TPEC)., College Station, TX, 2018, pp. 1-6.

[12] Y. Koç, T. Verma, N. A. M. Araujo, and M. Warnier, "MATCASC: A tool to analyse cascading line outages in power grids," in *Proc. IEEE Intl. Workshop Intelligent Energy Syst. (IWIES)*, Vienna, Austria, pp. 143-148, Nov. 2013.

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

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

[15] H. Zhu and G. B. Giannakis, "Sparse Overcomplete Representations for Efficient Identification of Power Line Outages," *in IEEE Transactions on Power Systems,* vol. 27, no. 4, pp. 2215-2224, Nov. 2012.

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

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

[18] M. Cui, J. Wang, J. Tan, A. R. Florita and Y. Zhang, "A Novel Event Detection Method Using PMU Data With High Precision," *in IEEE Transactions on Power Systems,* vol. 34, no. 1, pp. 454-466, Jan. 2019.

[19] K. Venugopal, P. Madhusudan and A. Amrutha, "Artificial neural network based fault prediction framework for transformers in power systems," *2017 International Conference on Computing Methodologies and Communication (ICCMC),* 2017, pp. 520-523.

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

[21] L. L. Grant and M. L. Crow, "Comparison of Matrix Pencil and Prony methods for power system modal analysis of noisy signals," *2011 North American Power Symposium*, 2011, pp. 1-7.

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

[23] C. M. HWEE, "Matrix pencil method as a signal processing technique performance and application on power systems signals," Ph.D. dissertation, 2013.

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

[25] 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," *in IEEE Transactions on Power Delivery*, vol. 32, no. 5, pp. 2179-2188, Oct. 2017.

[26] 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.

[27] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*. McGraw-hill New York,1994, vol. 7.

[28] D. J. Trudnowski, J. M. Johnson and J. F. Hauer, “Making Prony analysis more accurate using multiple signals,” *in IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 226-231, Feb. 1999, doi: 10.1109/59.744537.

[29] L. Yu and H. Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution,” *in Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 856–863.

[30] T. van der Ploeg and E. W. Steyerberg, “Feature selection and validated predictive performance in the domain of legionella pneumophila: a comparative study,” *BMC Research Notes*, vol. 9, no. 1, pp. 1–7,2016.

[31] J. Li and H. Liu, "Challenges of Feature Selection for Big Data Analytics," *in IEEE Intelligent Systems*, vol. 32, no. 2, pp. 9-15, Apr. 2017.

[32] M. Sheikhan, M. Bejani, and D. Gharavian, “Modular neural-svm scheme for speech emotion recognition using anova feature selection method,” *Neural Computing and Applications*, vol. 23, no. 1, pp. 215–227, 2013.

[33] 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.

[34] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys, “Robust biomarker identification for cancer diagnosis with ensemble feature selection methods,” *Bioinformatics*, vol. 26, no. 3, pp. 392–398,2010.

[35] J. Van Hulse, T. M. Khoshgoftaar, A. Napolitano, and R. Wald, “Threshold-based feature selection tech-niques for high-dimensional bioinformatics data,” *Network modeling analysis in health informatics and bioinformatics*, vol. 1, no. 1-2, pp. 47–61, 2012.

[36] T. Fawcett, “An introduction to ROC analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874,2006.