Pacific Northwest NATIONAL LABORATORY

PNNL-30452

Machine Learning for Synchrophasor Analysis

Final Project Report

September 2020

Huiying Ren
Zhangshuan Hou
Heng Wang
Pavel Etingov

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <https://www.ntis.gov/about>
Online ordering: http://www.ntis.gov

PNNL-30452

Machine Learning for Synchrophasor Analysis

Final Project Report

September 2020

Huiying Ren
Zhangshuan Hou
Heng Wang
Pavel Etingov

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

The report presents results from the development of a cloud-based, Big Data analysis framework for power systems. The computational pipeline uses the Apache Spark framework running in an OpenStack cloud infrastructure. A real-world phasor measurement unit (PMU) dataset has been used to carry out the analysis. Several Machine Learning (ML) methods have been developed and implemented for event and anomaly detection and classification. Actual examples of power system events detection and analysis using synchrophasor data are presented. It has been shown that applications of the cloud-based computing environment and the Apache Spark framework enable a significant increase in the computational efficiency of large-scale PMU data analysis.

Summary

Rising deployments of phasor measurement units (PMUs), smart meters, digital fault recorders (DFRs), and other contemporary measurement devices dramatically increase the size of data collected by electrical utilities. This digital information is frequently unstructured, has different time scales, and is stored on different servers and databases. The size of the collected datasets is growing rapidly, which complicates data processing and analysis. However, because the collected information contains many insights about the power system's state and its dynamic behavior, extracting this knowledge can significantly increase situational awareness, detect system-wide or local anomalies (e.g., under- frequency or voltage events), validate system models, and discover/predict equipment malfunctions.

This report presents results of synchrophasor information analysis conducted on a cloud-based Hadoop and Spark Big Data infrastructure. The framework is based on the Pacific Northwest National Laboratory's (PNNL) Institutional Cloud Computing OpenStack installation. The Hadoop Distributed File System (HDFS) is used to store the raw PMU information, and then Spark is used for data analysis and ML. Analysis results presented here are based on the real synchrophasor data provided by the Bonneville Power Administration (BPA). Several statistical and ML methods were developed and applied to this synchrophasor dataset to detect different types of events (e.g., frequency or voltage) and abnormalities. The aim of this work is to develop technologies and techniques that improve power system situational awareness and reliability.

We apply a set of signal processing and machine learning approaches aiming at deciphering the characteristic behaviors of multiple PMU attributes (e.g., voltage, frequency, rate of change of frequency, phase angle), including their auto-correlation, cross-dependence, similarities and discrepancies across units and temporal scales, and distributions of anomalies and their linkages to potential external factors such as weather events. The PMU measurements, recorded events, and weather extremes are all from real-world datasets. The findings from the study can help understanding the system dynamics. The derived metrics can be directly used for adjusting or filtering simulated PMU data used for advanced algorithm development.

We adopted the Long Short-Term Memory (LSTM) based Deep Neural Network (DNN) models to predict multiple steps ahead and detect abnormal events by address both the spatial and temporal variations in PMUs. Different model configurations were evaluated to yield an optimal model parameter set for the high resolution, complex, and dynamic PMU dataset. The decent relative error is obtained at each testing point which can be used for the abnormal event detections.

We proposed and developed framework features a scoring system for the anomaly detection. We also evaluated the effect of an alternative scoring system on our detection framework. Compared with the additive scoring system, the alternative multiplicative scoring system is much stricter in that the score for each MRA scale at each unit is multiplied to obtain scores of either 0 or 1. Using this multiplicative scoring system, both false alarm rate and detection rate were reduced.

We developed and evaluated a deep learning Convolutional Neural Network (CNN) model to identify locations and predict types of various faults in the power system. Faults in different spatial zones and locations was simulated with four distinct types and used for CNN training and testing. The CNN is composed of a number of layers, including convolutional, pooling, dropout and dense layer, which are designed to adaptively learn spatial hierarchies of features.

Hyperparameter search were performed to determine the optimal model configuration and the final model for fault classification and prediction.

In future work, we plan to continue both a mathematical and software enhancement of this framework’s functionality by adding new analytical modules and additional data sources, like supervisory control and data acquisition (SCADA) data, and also weather information. We are also going to shift from PNNL institutional cloud to Amazon AWS or Microsoft Azure cloud platform to improve computational performance of the developed analytical framework.

Acknowledgments

The authors are thankful for the support provided by Sandra Jenkins of the U.S. Department of Energy, Office of Electricity, and Joe Eto (Lawrence Berkeley National Laboratory). The authors also acknowledge our colleagues Marcelo Elizondo, Bharat Vyakaranam, Dimitri Zarzhitsky, Jeff Dagle and Eric Andersen at PNNL for their support, comments, and review of this work.

Acronyms and Abbreviations

ACF	Auto-correlation function
API	Application programming interface
ASOS	Automated Surface Observing Systems
BPA	Bonneville Power Administration
CDF	Cumulative distribution function
CNN	Convolutional Neural Network
DFR	Digital fault recorder
DNN	Deep Neural Network
DOE	(U.S.) Department of Energy
EIOC	Electricity Infrastructure Operations Center
HDFS	Hadoop Distributed File System
HPC	High-performance-computing
IaaS	Infrastructure-as-a-Service
LSTM	Long Short-Term Memory
ML	Machine Learning
MRA	Multi-resolution analysis
NOAA	National Oceanic and Atmospheric Administration
PACF	Partial autocorrelation function
PCA	Principal component analysis
PMU	Phasor Measurement Units
PNNL	Pacific Northwest National Laboratory
ROCOF	Rate of change of frequency
RMSE	root mean square error
SQL	Structured query language
SNR	Signal to noise ratio
WECC	Western Electricity Coordinating Council
WT	Wavelet Transform (also called wavelet decomposition)

Contents

Figures

Tables

3.0 Introduction

Rising deployments of phasor measurement units (PMUs), smart meters, digital fault recorders (DFRs), and other contemporary measurement devices dramatically increase the size of data collected by electrical utilities (Akhavan-Hejazi and Mohsenian-Rad 2018, Kezunovic et al. 2020, Wang et al. 2016). This digital information is frequently unstructured, has different time scales, and is stored on different servers and databases. The size of the collected datasets is growing rapidly, which complicates data processing and analysis. However, because the collected information contains many insights about the power system's state and its dynamic behavior, extracting this knowledge can significantly increase situational awareness, detect system-wide or local anomalies (e.g., under- frequency or voltage events), validate system models, and discover/predict equipment malfunctions.

For the past decade, technologies for Big Data analytics, cloud computing and machine learning (ML) have been developing very rapidly, and have been applied in many different engineering areas, including power system studies (Zhou et al. 2016, Vajjala 2016). New methods and computer frameworks for Big Data collection and analysis are based on distributed storage and parallel processing of information. Many of the Big Data analytical frameworks are open-source software projects, making it possible to apply this technology to an organization's existing commodity computing infrastructure without incurring new licensing costs. The per-unit cost of hardware components (e.g., central processing units, memory, and storage) has decreased dramatically as a function of computational performance. Combined with the lack of licensing costs for the state-of-the-art analytic solutions for Big Data analysis, it allows for use of either on-premise or Infrastructure-as-a-Service (IaaS) computer clusters by a broader community of researchers and industrial customers. In addition, popular commercial cloud services offered by multiple providers (e.g., Amazon Web Services, Google Cloud, Microsoft Azure, etc.) take this abstraction even further with the Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) business models, which feature a high-degree of customization and a variety of pre-packaged solutions for both data management, analytics, visualization, long-term secure storage, and many other operational and mission-impacting concerns.

The Apache Hadoop (Apache 2017) framework has been successfully used as a foundation by many software solutions for distributed data analysis. An active community of Hadoop developers produced a thriving open-source software ecosystem for high-performance data analysis. It distributes (i.e., partitions) large datasets across multiple storage and computation nodes within a computer network. Using many common Hadoop-inspired technologies, Apache Spark is a popular and widely-used open source framework for Big Data analysis and ML (Apache 2020). ML is a method of data analysis that automates analytical model-building. ML techniques could build general analytical models based on the data analysis and find hidden insights without being explicitly programmed for each specific problem. Moreover, an ML engine can continuously improve its model from new data. Spark is based on a high-performance distributed memory architecture and it achieves exceptional performance in parallel data processing. Together, Spark and Hadoop have been used in different areas including power system applications (Zhou et al. 2015, Šutić and Varga 2017).

This report presents results of synchrophasor information analysis conducted on a cloud-based Hadoop and Spark Big Data infrastructure. The framework is based on the Pacific Northwest National Laboratory's (PNNL) Institutional Cloud Computing OpenStack installation. The Hadoop Distributed File System (HDFS) is used to store the raw PMU information, and then Spark is used for data analysis and ML. Analysis results presented here are based on the real synchrophasor data provided by the Bonneville Power Administration (BPA). Several statistical and ML methods were developed and applied to this synchrophasor dataset to detect different types of events (e.g., frequency or voltage) and abnormalities. The aim of this work is to develop technologies and techniques that improve power system situational awareness and reliability.

4.0 Synchrophasor Data Analytical Framework

4.5 Computer cluster

PNNL's Institutional Cloud Computing system is based on the OpenStack open-source platform (OpenStack 2019) for cloud computing, and the Hadoop and Spark computational environment is provided by the Cloudera Express distribution (Cloudera 2020). The computer cluster network topology used in this study is shown in Figure 2. It consists of 20 nodes including one master head node. Each node is equipped with eight core processors, 32 GB of RAM, and 100 GB of disk storage space. Apache Spark for Big Data analysis and ML (Apache 2020), and the Apache Hive based structured query language (SQL) interface for data storage, management are configured through the Cloudera Manager.

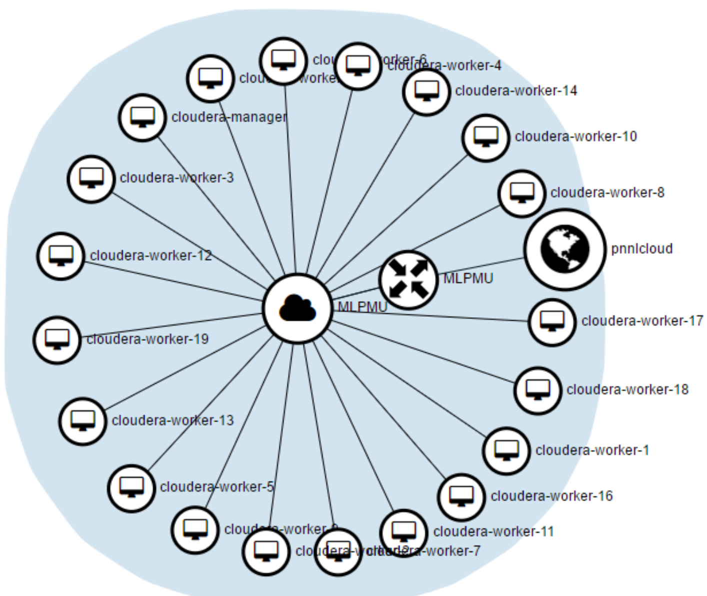

Figure 1. Computer cluster network configuration.

Our system's main functional components are diagrammed in Figure 2. PNNL receives the synchrophasor measurements as a real-time data stream from BPA, storing it at the PNNL's Electricity Infrastructure Operations Center (EIOC) (PNNL 2020) as a set of PDAT-formatted files. The PDAT format was developed by the BPA, and is used to capture PMU measurements from multiple devices in binary files (the format is based on the IEEE Standard C37.118.2-2011 data frames) (Faris , IEEE 2011). Each file contains one minute of PMU data, collected at the 60 samples per second rate. An approximate size of the dataset as a function of time is given in Table 1.

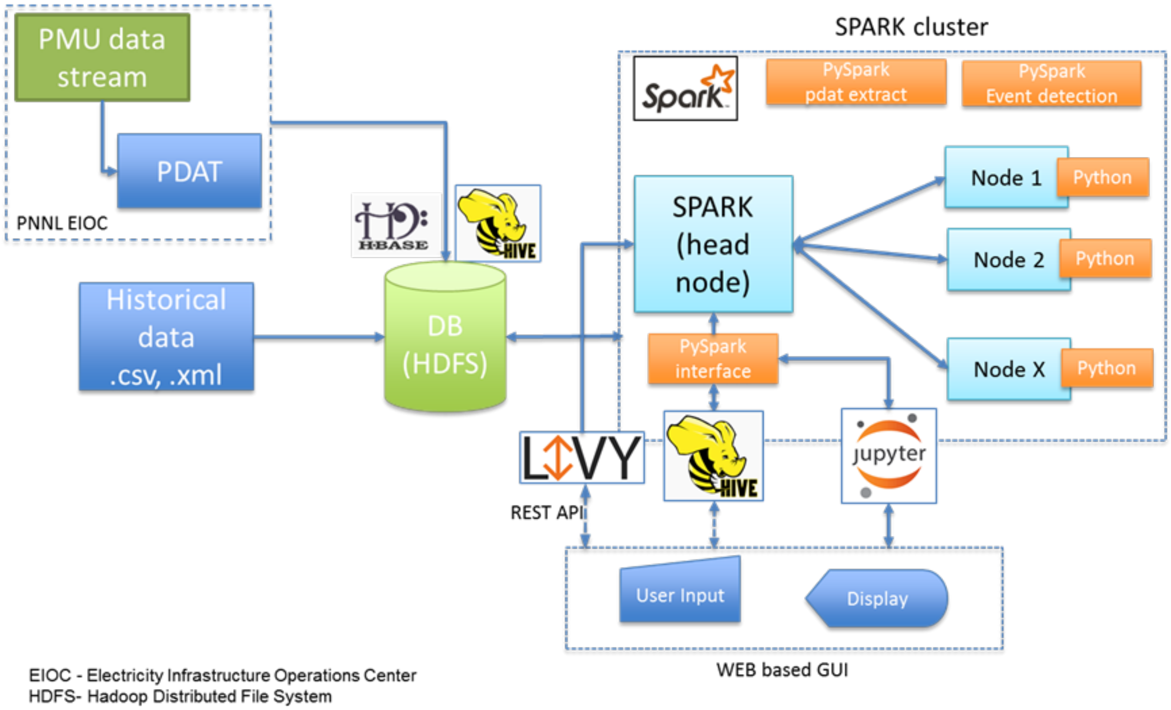

Figure 2. Suite of open source tools

Table 1. Approximate dataset size

1 minute	1hour	1 day	1 month	1 year
5 MB	300 MB	7.2 GB	216 GB	2.6 TB

4.6 Data extraction

All the PDAT files with the synchrophasor data are stored and distributed among the cluster nodes via the Hadoop Distributed File System (HDFS). The Python programming language, because of its wide use by the Data Science community and the availability of a large number of open-source data-processing modules, was selected as the programming environment for our data processing pipeline. The pipeline itself is split into several stages, and the interaction with the Spark execution engine is implemented using the PySpark binding. The first processing stage reads data from the HDFS hosted PDAT binary files and creates Spark data frames (Apache 2020). Here, the use of Spark enables significantly increased speed of data extraction (extraction of a one-hour dataset takes only 10-12 seconds compared to the 3-5 minutes required on a single personal computer).

As part of the second stage of data analysis, the Spark-processed data frames are saved as Hive tables in order to enable the use of Spark SQL application programming interface (API). Our design enables external modules, such as MS Windows standalone applications or web-based graphical user interfaces, to interact with Hive directly, further increasing the number of analytic and visualization options that can benefit from the cloud-based system architecture.

5.0 ML Methodology

5.5 Spatiotemporal pattern recognition in PMU signals

We apply a set of signal processing and machine learning approaches aiming at deciphering the characteristic behaviors of multiple PMU attributes (e.g., voltage, frequency, rate of change of frequency, phase angle), including their auto-correlation, cross-dependence, similarities and discrepancies across units and temporal scales, and distributions of anomalies and their linkages to potential external factors such as weather events. Data analytics are applied to PMUs from the U.S. Western Electricity Coordinating Council (WECC) system. The PMU measurements, recorded events, and weather extremes are all from real-world datasets. The findings from the study and mechanistic understanding of the PMU dynamics help provide guidance on system control or preventing blackouts. The derived metrics can be directly used for adjusting or filtering simulated PMU data used for advanced algorithm development.

5.5.2 Time series pattern recognition

Auto-correlation analysis is used to look at temporal continuity and periodicity in the PMU data series. Auto-correlogram, also called serial correlation, is the correlation of a time series sequence with a delayed copy of itself as a function of lag or delay. It measures the similarity between observations as a function of the time lag between them. The autocorrelation analysis extracts repeating patterns, such as the presence of a periodic signal obscured by noise or identifies the missing fundamental frequencies. The partial autocorrelation function (PACF) gives correlation of a stationary time series with its own lagged values, regressed the values of the series at shorter lags and helps determine the appropriate lags in the autoregressive patterns.

Time-series auto-correlation and partial auto-correlation analyses are summarized for the multivariate PMU attributes across different months. Both the auto-correlation function (ACF) and partial ACF (PACF) quantify the strength of a relationship with an observation in a time series with observations at prior time steps with different time lags; but the latter tries to remove the indirect correlations. They are used together to determine the strength of temporal continuity and existence of periodicity in the PMU signals. The generality and transferability of these continuity characteristics are further evaluated with respect to temporal factors such as hour of the day or seasons. These information help provide guidance on short-term anomaly detection and/or mid- to long-term prediction. Figure 3 shows that the two hours on the same day can have quite different temporal continuities. Here we define a measure of such continuity called correlation range, which is the time lag beyond which the auto-correlation values drop below a threshold (i.e., 0.05). For the two example hours, such correlation ranges are 520 seconds and 260 seconds, respectively. The second hour has much weaker temporal continuity.

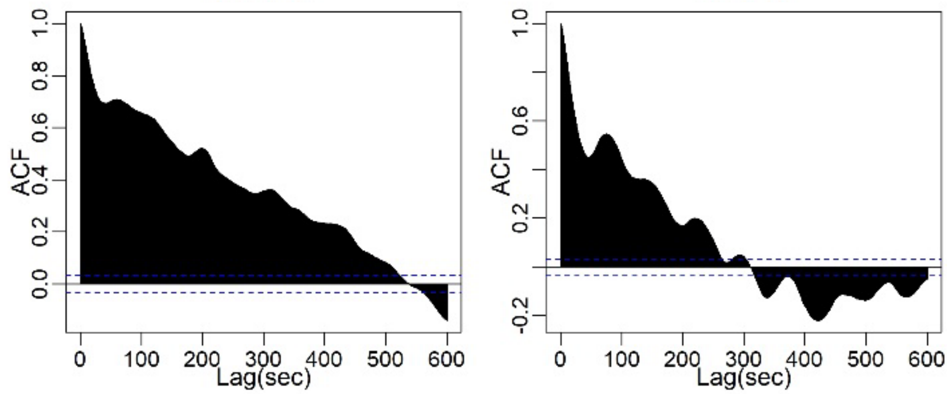

Figure 3. ACF for two randomly selected hours on the same day with different temporal continuity. The blue deshed line is the threshold 0.05.

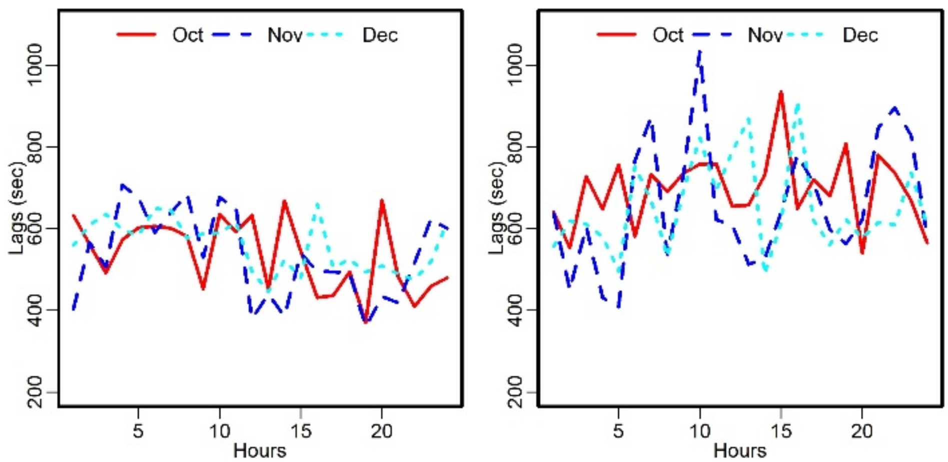

Figure 4. Continuity – correlation ranges of PMU frequency in left panel and voltage in right panel, based on ACF analysis.

Figure 4 summarizes the correlation ranges of PMU attributes as a function of hour of a day during different months. The daily pattern in PMU frequency is obvious with larger correlation ranges (around 600 seconds), that is, stronger continuity in the morning but 400-500 seconds correlation range (weaker continuity) in the afternoon. This is related to the relatively larger power system variability in the afternoons during the months of study. However, the other PMU attribute, voltage, is having a different but also consistent patterns across months. It has stronger continuity during the middle of the day (correlation range 600-1000 seconds), but weaker continuity at nights (correlation range 400-700 seconds). But note that the correlation range metric is varying from hour to hour for the PMU voltage. PACF provides guidance on the order of auto-regressive models for prediction purposes. Here PACF indicates that these PMU attributes is following an autoregressive order 1 (AR(1)) model. These correlation ranges and AR model order should be among the metrics for guiding anomaly detection, classifying and labeling anomalies, verifying extracted temporal patterns, or evaluating and filtering simulated PMU data.

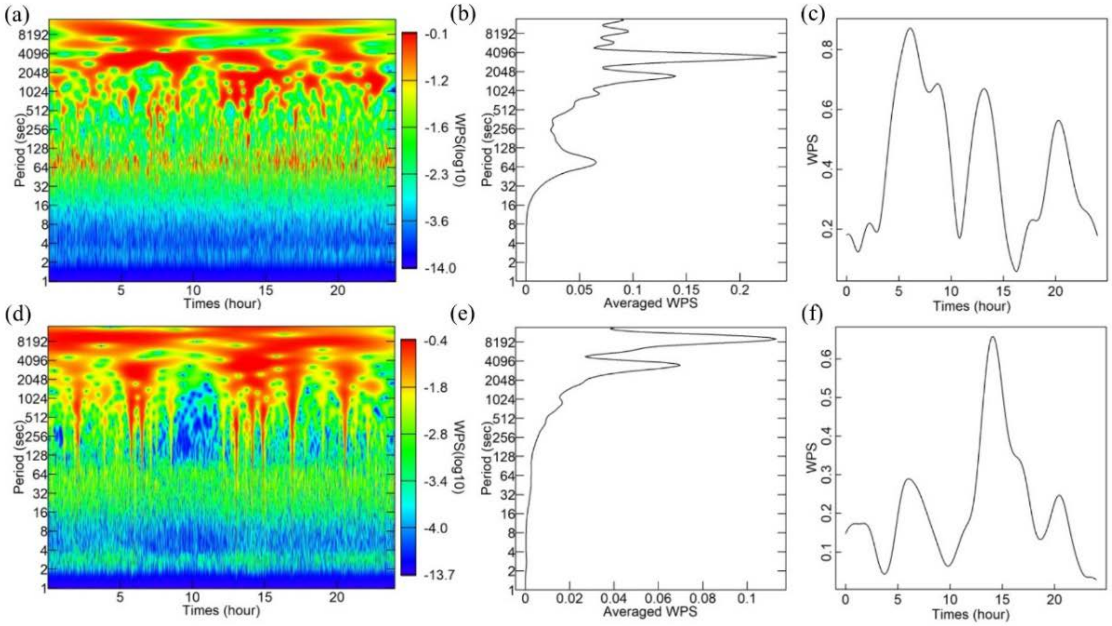

Figure 5. The wavelet spectra of PMU attributes at unit #5. (a) the decomposed coefficients at various temporal scales for PMU frequency for the day September 14; (b) the corresponding wavelet power distribution; (c) the extracted dominant hourly component; (d) the decomposed coefficients at various temporal scales for PMU voltage for the day September 14; (e) the corresponding wavelet power distribution; and (f) the extracted dominant hourly component.

Wavelet decomposition (WT) has widespread use in speech and image processing as well as time series analysis and is suited to nonstationary signals. The forward transformation separates the original signals into multiple components. At each step, the signal is decomposed simultaneously using a high-pass filter h and a low-pass filter g, resulting in detail coefficients and approximation coefficients (the remaining mixed signal), respectively. WT extracts dominant component in a mixture signal and localizes anomalies at multiple temporal scales (Daubechies 1992). In addition, signal to noise ratio (SNR) analysis are conducted to extract and summarize the SNR attributes to complement the wavelet spectral information in the frequency domain.

The ACF analysis shows that the correlation ranges are within one hour, which means that for detecting short-duration events, one does not need to look beyond one hour of data which would involve unnecessary computational burden; but in order to understand mid- to long-term behaviors, wavelet spectra analysis (see Fig. 5b and 5e) tells us that hourly, daily, even weekly signals should be examined. Figure 5e and Figure 5f also show that the hourly component is not just sinusoidal, but rather having higher peaks during mid-day time periods. These components are attributed to potential external drivers such as weather attributes. Both PMU and weather attributes can be wavelet-decomposed to help evaluate their associations at matching temporal scales.

In addition, we perform signal to noise ratio (SNR) analysis and extracted/summarized the SNR attributes as a complementary metric to the wavelet spectra. The autoregressive AR(1) low-frequency noise is generated with the maximum power spectrum 0.08 and applied to compute the SNR of PMU frequency and voltage signal for hours of a day in Figure 6. The ratios of SNR are relatively stable for frequency signals which vary within 1dB, but still have the two higher SNR peaks during the earlier morning and later mid-day and lower SNR at mid-day. For the voltage attribute, both the values and variations of SNR ratios are much higher and subject to larger ranges.

The SNR ratios in both frequency and voltage attributes vary with months but are consistent across the units. SNR is another useful and commonly used measure particularly for characterizing the level of noises in the data. The measure can be directly used for adding noises to PMU data generated by physics-based data simulators, with the purpose to augment the existing PMU databases to be adequate for reliable artificial intelligence model development and evaluations.

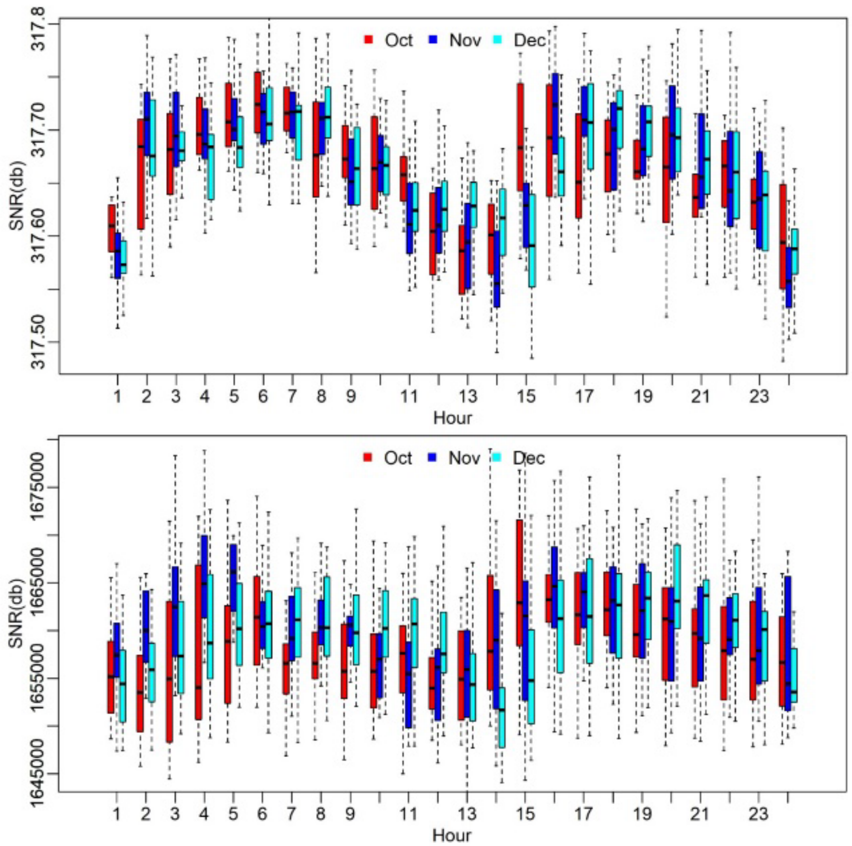

Figure 6. The diuanal rythem of SNR obtained from frequency and voltage PMU attributes for different months.

5.5.3 Similarities and differences across units

PCA is used to evaluate the similarities and differences across the PMU units, that is, the spatial cross-dependence. A scree plot is used to determine the number of principal components needed to capture the major variability in the data matrix. A biplot may show clusters of samples based on their similarity, and with loading projections to show how strongly each characteristic influences a principal component. Figure 7 is a summary of block-wise (5-min time window-based) PCA results. It is an example of tracking the real-time changes of PMU signal behavior, in terms of the system complexity (number of critical principal components and their relative contributions) and similarity (clustering). From Figure 7Figure 5, PMU frequency is generally dominated by the first two or three components, and the units can be grouped into 3-4 groups based on their similarities, and such groups are spatially coherent as geographically-adjacent units tend to be clustered together. These findings have two implications: (1) although the units have strong cross-dependence given the nature of power system, there are still local/regional behaviors of PMUs that subject to local/regional external factors such as weather impacts; (2) for simulation-based data augmentation, the computational effort can be reduced by simulating cluster behaviors then superimposing within-cluster variabilities.

Figure 7 also shows that the relative changes and contribution/dominance of the first four principal components are changing but very slightly with time, and such changes are more related to the hour to hour variations in the original signals, but not much affected by the occurrences of a number of recorded events each of which lasts about 20 seconds. Therefore, although there are many practices using PCA for anomaly detection, one can use PCA for anomalies at the corresponding dominant temporal scales, but not for real-time PMU pulse event detection.

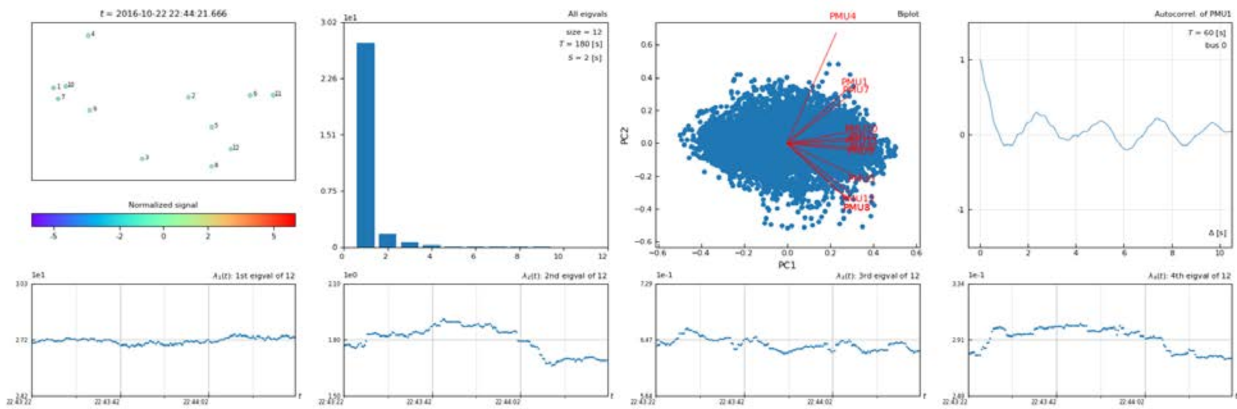

Figure 7. Block-wise PCA. Top row shows the rough spatial locations of the units, scree plot, biplot, auto-correlation, respectively. The bottom row shows the changes of the first four principal components with respect to time. A 5-min moving window is used to conduct PCA.

5.5.4 Similarities/discrepancies between days/months

Next, we look at the deviations of time series under different conditions (e.g., days, seasons), and use Taylor Diagram to integrate/compare differences in phase angle and magnitude, and combines Pearson correlation, root- mean-square error, and individual standard deviations. The time series are grouped in clusters based on distances (combined similarity measure) on the diagram. Figure 8 shows that angle differences change dramatically from day to day, around -10 on Nov 1 but around -25 on Nov 5, and there seems to be a gradual shift day by day and beyond a week. This suggests the lack of mid-term continuity and periodicity in the PMU data. The long-term behaviors are expected to be attributed to factors at the corresponding scales associated with weather attributes. The between-day variability is bigger than within-day variation ranges in magnitudes but not in phases (patterns), although Nov 6 behaves differently representing a change of system status. Taylor Diagram also shows that when cross-correlation coefficient goes up to 0.6 in a cluster, their deviations in terms of root mean squared errors are not necessarily small.

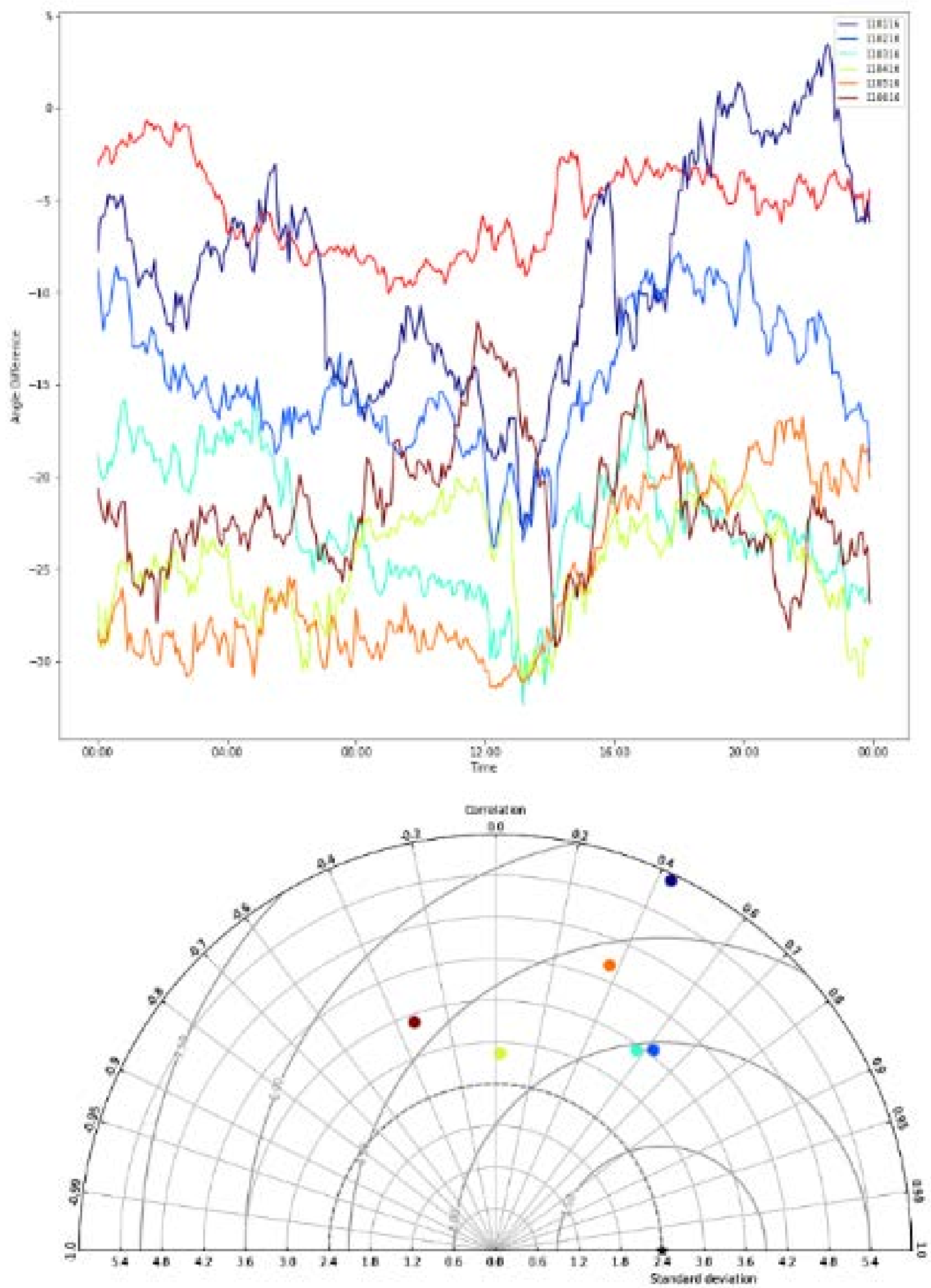

Figure 8. PMU angle difference time series during several adjacent days, and the corresponding Taylor Diagram.

5.5.5 Companion of Weather and PMU Anomalies

Next, we identify and localize anomalies using Multiresolution Analysis (MRA)-based anomaly detection (Ren et al. 2018), given potentially strong impacts of weather extremes on the PMU signal anomalies at various temporal scales.

Weather data at adjacent weather stations are downloaded from Automated Surface Observing Systems (ASOS) under National Oceanic and Atmospheric Administration (NOAA). The weather attributes including precipitation, temperature, wind speed, gust, humidity, pressure, and so on, have been analyzed using a measure of outliers' likelihood calculated over a range of the observations. In this study, although wind gust has a clear and easy-to-interpret linkage to the identified PMU anomalies (with deviations and durations passing pre-set anomaly thresholds), other attributes, particularly their extracted anomalies, have good matches with PMU anomalies as well, as shown in Figure 9.

Figure 9 shows the recorded actual historical PMU events (red lines) and the weather events (green triangles) and durations of weather events (blue dashed lines by lumping the adjacent weather anomalies). As the recorded PMU events have a high chance of occurring with weather anomalies and usually occur at certain durations into the identified weather events, the identified weather anomalies can serve as an early awareness indicator. Quantitatively, the conditional probabilities of PMU event occurrences can be derived and integrated with anomalies classified from weather forecast for anomaly prediction purpose or for adding pulse event signals to the normal condition time series generated in PMU data simulation efforts, such that the simulated data contains appropriate timing and realistic probability of pulse inputs (non-noises) to the signals. It is expected that anomalies added this way are similar within a cluster with similar weather factor(s) or climate types.

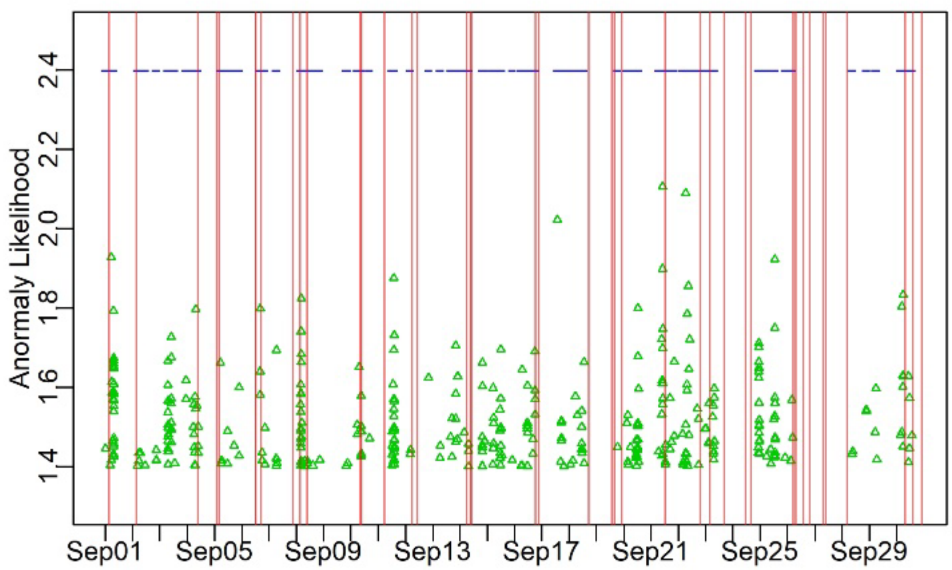

Figure 9. Correspondence of localized anomalies with extreme weather events, the green triangle dot is the extreme weather event obtian from the ASOS monitoring, the blue dashed line shows the timing and duration of the weather anomalies by lumping the adjacent anomalies identified.

5.6 Offline anomaly detection

The wavelet transform separates 1-D signals into 2-D components overlapping in time-frequency domain. The wavelet techniques have been widely used because of its multiple time-frequency resolution (Mallat 1999, Lounsbery, DeRose, and Warren 1997). Wavelet transforms has been proven to be very efficient (Benedetto and Li 1998) in signal analysis with the reduction of coefficients numbers as the scaling factor increases. Discrete wavelet transform (DWT) is sufficient to decompose and reconstruct most power quality problems, which can provide information adequately and efficiently (Gaouda et al. 1999). Wavelet-based anomaly detection has been successfully applied for detecting network anomalies (Alarcon-Aquino and Barria 2001, Lu and Ghorbani 2009, Aradhye et al. 2004, Wang et al. 2013, Bhuyan, Bhattacharyya, and Kalita 2014) for various systems and problems. Wavelet-based multi-resolution analysis (MRA) uses wavelet function and scaling function to decompose and construct the signal at different resolution levels. The anomaly phenomena can be detected and localized at each resolution level.

5.6.2 Multi-resolution analysis (MRA)

Wavelet process using DWT can filter the input signal with lowpass and highpass filters. Lowpass filter is defined by scale function, that is,

$$\varphi_{j,k}(x) = 2^{\frac{j}{2}}\varphi\left(2^j x - k\right) \qquad (1)$$

where k is the translation coefficient and j is the scale factor (Mallat 1989). The expansion function of any subspace can be built from double-resolution copies of themselves, so the scaling function can be transformed to (Burrus, Gopinath, and Guo 1997)

$$\varphi(x) = \sum_n h_\varphi(n)\sqrt{2}\varphi(2x - n) \qquad (2)$$

where h_φ is the scaling function coefficients. The highpass filter is defined by wavelet function, that is

$$\psi_{j,k}(x) = 2^{\frac{j}{2}}\psi\left(2^j x - k\right) \qquad (3)$$

Then it can be expanded to

$$\psi(x) = \sum_n h_\psi(n)\sqrt{2}\psi(2x - n) \qquad (4)$$

where h_ψ is the wavelet function coefficients. The DWT decomposes signals into its approximation (A) and detail (D) components, respectively. Approximation of the signal at resolutions $2^{-j}, j = 0, 1, 2$ ... can be obtained at decreasing levels of details. A detailed theory of MRA has been developed by Meyer (Meyer 1985), and can be mathematically presented as

$$f(x) = \sum_n C_o(n)\varphi(x - n) + \sum_n \sum_{j=0}^{j-1} D_j(n) 2^{\frac{j}{2}}\psi\left(2^j x - n\right) \qquad (5)$$

where C_o is the 0 level scaling coefficient and D_j is the wavelet coefficient at scale j. The scaled and translated wavelet $\psi\left(2^j x - n\right)$ in MRA is decomposed signals in time-frequency domain. Orthogonal wavelets expanded by 2^j carry signal variations at the resolution 2^{-j}. A number of wavelet families have been developed with different characteristics, and a well-known family is Daubechies (db) (Avdaković and Čišija 2015). In our paper, Haar (db1) wavelet is employed in the MRA. Haar's wavelet has 1 moment of wavelet function which has linear phase and complete localization in time domain (Avdaković and Čišija 2015) (Daubechies 1992).

5.6.3 Wavelet-based anomaly detection framework

We used a 30-day real-world PMU datasets to test our framework. 32 historical events were recorded during the 30-day testing time period. We analyzed four attributes including the voltage, angle variation, frequency and ROCOF for each PMU dataset. In general, the ROCOF attribute has nosier signals than the rest attributes and the detected candidates have weak spatiotemporal correlations. As observed in (Konakalla and de Callafon 2016), ROCOF has large variance which reduces the event detection accuracy due to sensor and grid dynamics. The angle variation attribute contains similar and relative redundant information to the frequency attribute because the frequency is derived from the angel variation. Therefore, it is warranted to focus on the voltage and frequency attributes for event detection in this study. The framework of wavelet-based anomaly detection and classification is illustrated in Figure 10.

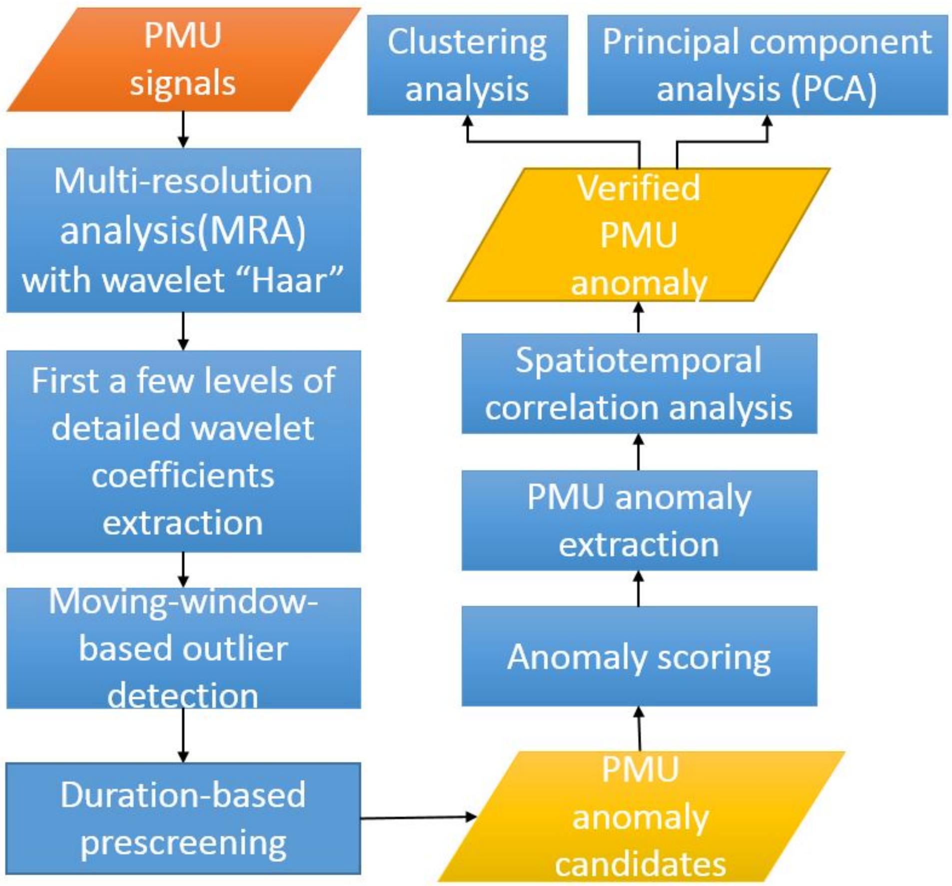

Figure 10. Wavelet-based PMU anomaly detection and classification framework.

5.6.4 Anomaly detection and classification

The developed detection framework was applied to the actual western interconnection synchrophasor data. The raw PMU signals were down-sampled to 1Hz, so the time resolution for the MRA procedure was 1 second, and the first three levels of MRA, D1, D2 and D3, have resolutions of 2-, 4- and 8-seconds, respectively. The 2-D time-frequency representation of real-world PMUs has significant benefits compared to the regular 1-D data in time domain for event detection perspective. Figure 11 illustrates MRA and moving-window outlier detection results for the first unit of PMUs (PMU1) using the frequency attribute. Given the observations of the signal and detailed wavelet coefficients, the events are more evidenced from the D3, D2, and D1 coefficients than from the original, therefore, it helps increase event localization accuracy. This increase with detail coefficients is the key-component of our event detection algorithm. The recorded historical events were marked in red in Figure 11(a), which shows that the anomalous candidates have been identified at each resolution level using our proposed detection framework.

In this study, we used a 30-day real-world PMU datasets to test our framework. 32 historical events were recorded during the 30-day testing time period. We analyzed four attributes including the voltage, angle variation, frequency and ROCOF for each PMU dataset. In general, the ROCOF attribute has nosier signals than the rest attributes and the detected candidates have weak spatiotemporal correlations. As observed in (Konakalla and de Callafon 2016), ROCOF has large variance which reduces the event detection accuracy due to sensor and grid dynamics. The angle variation attribute contains similar and relative redundant information to the frequency attribute because the frequency is derived from the angel variation. Therefore, it is warranted to focus on the voltage and frequency attributes for event detection in this study.

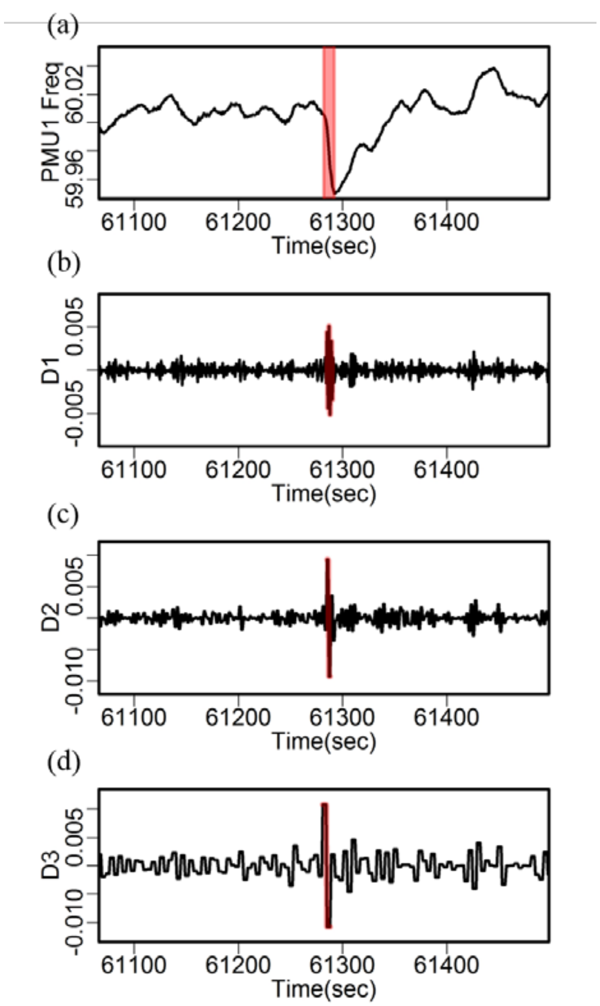

Figure 11. An MRA example using PMU 1 frequency attribute. The detected events at each scale are marked in red.

In our study, all recorded events have been detected with high scores. For example, Figure 12(a) illustrates one event occurred at all 12 units, which is detected based on the frequency attribute. At the time of event, the frequency amplitude jumped from 59.96 to 60.01 Hz within 10 seconds. The 12 units behave consistently at the event resulting in strong spatial correlations. This indicates systematic behavior and area-wide situation to be aware of. The same event was also detected from the voltage attribute with strong correlations across all units as shown in Figure 12(b). The amplitude of voltage for each unit increased by over 1000 Volts within 10 seconds. After the event, the voltage gets stabilized, but not necessarily return to the voltage level prior to the event.

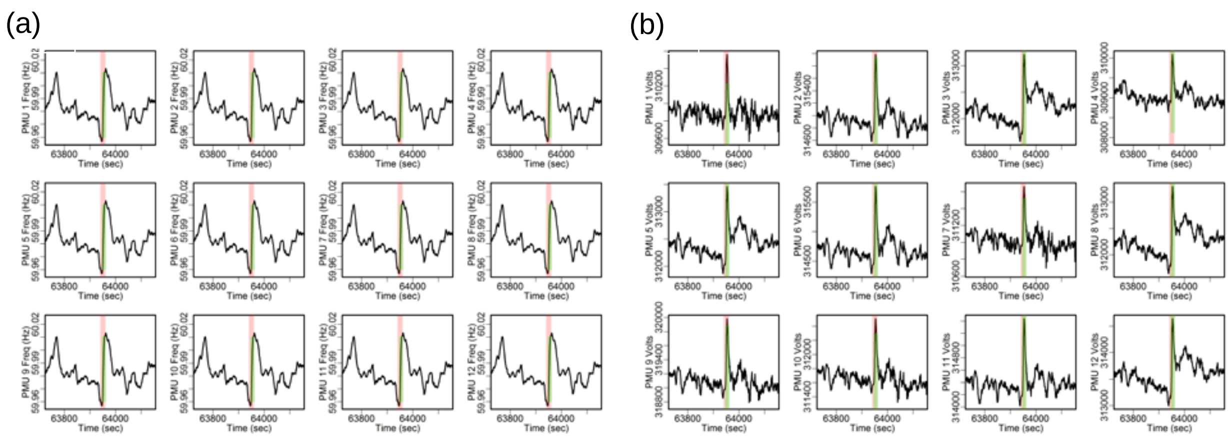

Figure 12. An example of abnormal event occurred across all units for (a) frequency attribute; (b) voltage attribute. The detected events for each unit are marked in red. The recorded historical events are marked in green.

PCA were applied to evaluate and classify the identified events. The PCA result is illustrated in Figure 13. The left panel shows the first two principal components of three attributes (voltage, angle variation and frequency). The events can be grouped easily based on the Biplot, for example, anomalies #83, #108, #109 and #142 have outstanding differences in both frequency and angle variation compared to other anomalies; while anomalies #11 and #141 have outstanding voltages among the anomalies. Another unsurprising observation is that angle variation has strong correlation with frequency, and the two have redundant information with very similar behaviors contributing mainly to the first principal component. The variability of voltage is the major contributor to the second component. By removing the redundant angle variation in PCA, the voltage and frequency are nearly orthogonal factors as shown in the right panel in Figure 13. PCA helps classify the identified events to be either frequency-related or voltage-related. However, there are a few exceptions, for example, events #2, #85, #113 and #116 are clearly identifiable using both frequency and voltage factors. And three of them were actually the historical recorded events which were detected using both voltage and frequency attributes.

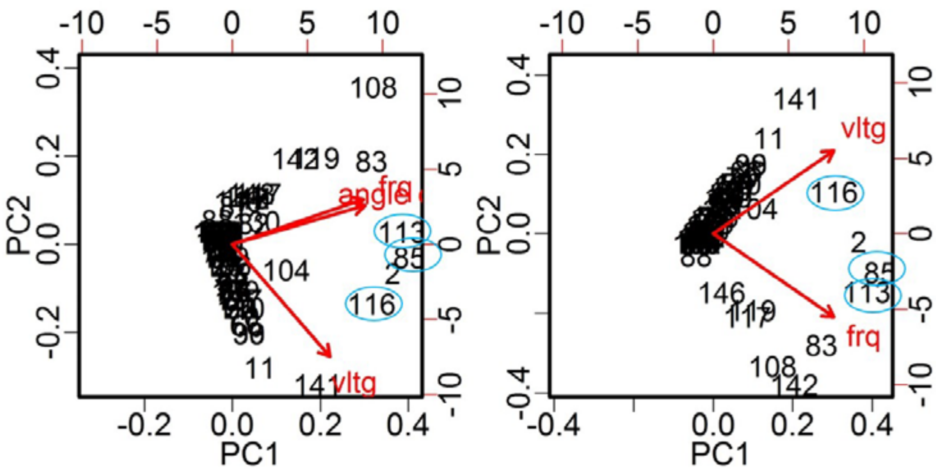

Figure 13. PCA Biplots of detected events using different PMU attributes. The historical recorded events are circled in blue.

5.7 Online anomaly detection

Dynamical machine learning solutions including state space model and Kalman filter are presented in this study to learn the nonlinear and nonstationary PMU measurements and accurately predict system behaviors in real-time. The anomalies can be detected within seconds by comparing the predicted system behaviors with the real system observations. The method proposed in this framework uses PMU data with a given time window (e.g., 5 seconds) using a dynamic nonlinear model, and then predicts system behaviors during the following time window. High prediction accuracy is achieved by applying the dynamic nonlinear model to the real-world PMU measurements – the anomalies detected are successfully validated given the recorded real-world events. High-performance-computing (HPC) techniques are utilized to further reduce computational time to provide real time power system situational awareness.

5.7.2 State space model

General dynamic linear models are a particular class of state space model which can be formulated by observation and model equations.

$y_t = F_t\theta_t + \vartheta_t,\ \vartheta_t \sim N(0, V_t)$yt=Ftθt+ϑt,ϑt~N(0,Vt) (6)

$\theta_t = G_t\theta_{t-1} + w_t, w_t \sim N(0, W_t)$xt=Gtxt-1+wt,wt~N(0,wt) (7)

In most applications, y_t are the time series observations, θ_t is the state vector, and F_t is the regression vector at time t in the observation equation (1). G_t is the state matrix at time t, and θ_{t-1} is the state vector at time t-1 in the system equation (2). The state vector θ_t changes with time, which is an important feature to model nonstationary time series. The associated errors are assumed to follow normal distribution with mean zero and varianceV_t. W_t is the time-dependent state evolution covariance matrix for θ_t; it captures the evolutionary changes in the regression parameters. The formulation of the above dynamic linear models are flexible for obtaining main features of the training time series.

In practice, $W_t$the state space model parameters are estimated using maximum likelihood estimation techniques. Marginal likelihood function $p(y_{1:t}|\theta)$ can be obtained sequentially by Kalman filter$p(y_{1:t}|\theta)$. In dynamic linear models, the Kalman filter passed the likelihood function $p(y_{1:t}|\theta_t)$ at the current time t, to $p(y_{1:t+k}|\theta_{t+k})$ at the next $t+k$ time steps, with updated inference on the state vector. It is the prediction step via state estimation.

An R Package DLM (Petris 2010) is adopted in our paper and the second order polynomial dynamic regression model is chosen considering the nonlinearity and nonstationarity of the PMU measurements. Polynomial dynamic linear models can well describe trend-nonstationarity of a time series (Petris, Petrone, and Campagnoli 2009).The second order polynomial model has a two-dimensional state space, and can be described by the matrices

$$F = c(1,0)$$

$$G = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad (8)$$

$$W = diag(W_1, W_2) \qquad (9)$$

5.7.3 Online anomaly detection framework

In the proposed framework, the second order polynomial dynamic regression model is built sequentially for PMU measurements of subsequent 5-minute time windows, where Kalman filter is applied to compute filtered values of the state vectors, together with their covariance matrices. The training errors are the differences between values fitted by dynamic regression model with Kalman filter and the actual PMU measurements within 5-minutes time window, and the prediction errors are defined as the differences between the expected values of future observations via prediction and actual system observations for the following 5 seconds. For the short-term predictions, we assume that the prediction errors and the training errors follow the same distributions. The cumulative probability distribution (CDF) of prediction errors is approximated to be normal and characterized by the mean and variance of the training errors. The CDF is defined as

$$F(x) = P(X \le x) = \sum_{t \le x} f(t) \qquad (10)$$

The exceedance probability of a prediction error is then computed as

$$P_i(X \le x) = \max\left(P_i(X \le x), 1 - P_i(X \le x)\right) \qquad (11)$$

A threshold of P_i can be used to screen the anomaly candidate points in the PMU data, based on whether its corresponding exceedance probability is greater than the threshold. Another threshold for anomaly candidate screening is the duration of the anomalies. Considering that the recorded anomaly events usually last between 5 and 20 seconds, the anomaly candidates are further screened accordingly; specifically, the candidates pass the screening with durations longer than the duration threshold. With both exceedance probability and duration thresholds, anomalies can be confirmed within 5 seconds of occurrence.

The developed online anomaly detection framework was applied to the 28-day actual WECC synchrophasor data with 25 historical recorded events. The raw PMU data contains 12 units representing signals at 12 different locations in the WECC system. There are four attributes including the voltage, angle variation, frequency and the rate of changes of frequency (ROCOF) at each PMU. The PMU data is stored in PDATA format (Faris) based on IEEE Std. C37.118.2-2011 data frames(IEEE 2011). PMU measurements were re-sampled at 1Hz.

Given the tremendous amount of data, the framework can be implemented on HPC clusters to facilitate real-time implementation. The framework is tested and compatible with the Constance cluster supported by the Pacific Northwest National Laboratory (PNNL) Institutional Computing (PIC) program. The Constance cluster has 520-node; each node contains 24 cores running at 2.3GHz, with 64 GB of 2300 MHz memory. Analyses of PMU signals can be done in parallel to accelerate the dynamic regression model development and Kalman filter calculation.

The flow chart of online anomaly detection framework is shown in Figure 14. In the next section, we demonstrate the training and prediction evaluation of the dynamic regression model, and evaluate the anomaly detection results by comparing with the recorded actual events.

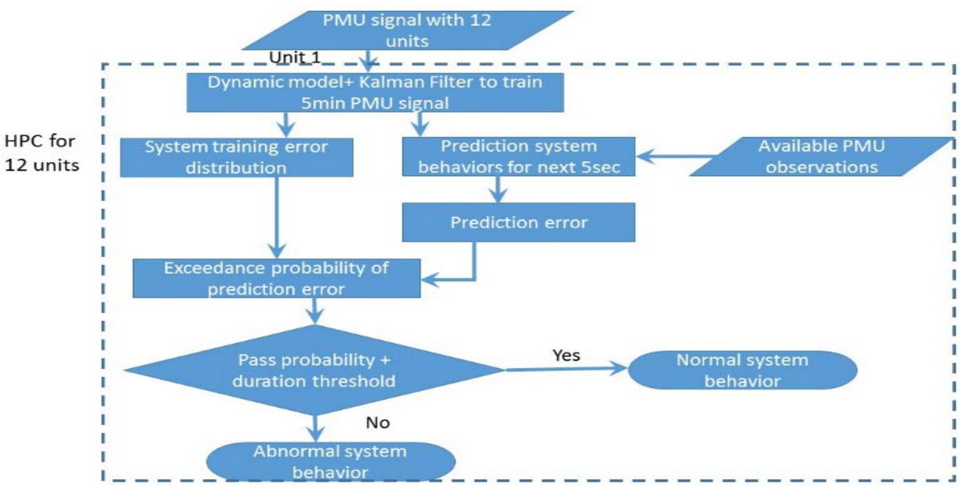

Figure 14. Flow chart of the online detection framework for PMU measurements.

The proposed framework was applied on the frequency attribute of the WECC PMU signals. The detected events were validated against frequency event database maintained by North American Electric Reliability Corporation (NERC) Resource.

First, the developed dynamic linear model was evaluated. The root mean square errors (RMSE) were computed as a measure of both training and prediction errors. The averaged RMSEs across 12 PMUs over a 1-day training time period are shown in Figure 15, demonstrating satisfactory goodness of fit of the dynamic regression models. Specifically, the averaged RMSEs are generally under 0.12% for the non-events time period.

5.7.4 Online anomaly detection results

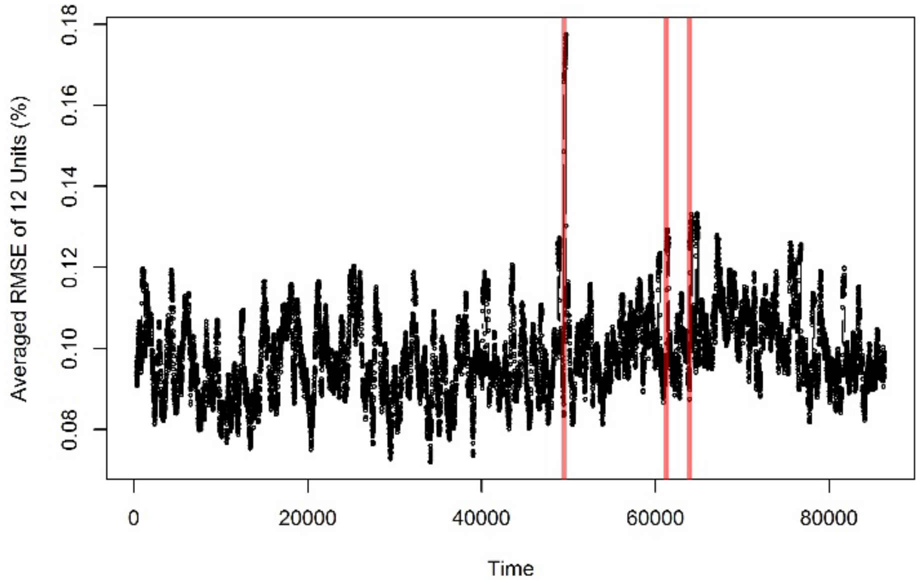

Figure 15. Averaged RMSE across the 12 units between observations and dynamic model fitting in sequential 5-minute training periods. The red vertical lines show temporal locations of recorded events.

When actual events occurred in the system (near the three red vertical lines), the RMSEs increase slightly. The averaged RMSEs are also calculated for the prediction time period, and are shown in Figure 16. The prediction RMSEs are higher than training RMSEs but the averaged prediction RMSEs across 12 units still be managed in a relative low range. The 99.9% quantile of averaged prediction PMSEs across 12 units is 1% which illustrates the accurate predictions of the dynamic model. The periods with RMSEs over 1.5% are highly likely to have some abnormal system behaviors. All three historical recorded events shown in Figure 3 have the relatively high RMSEs which are over 2%.

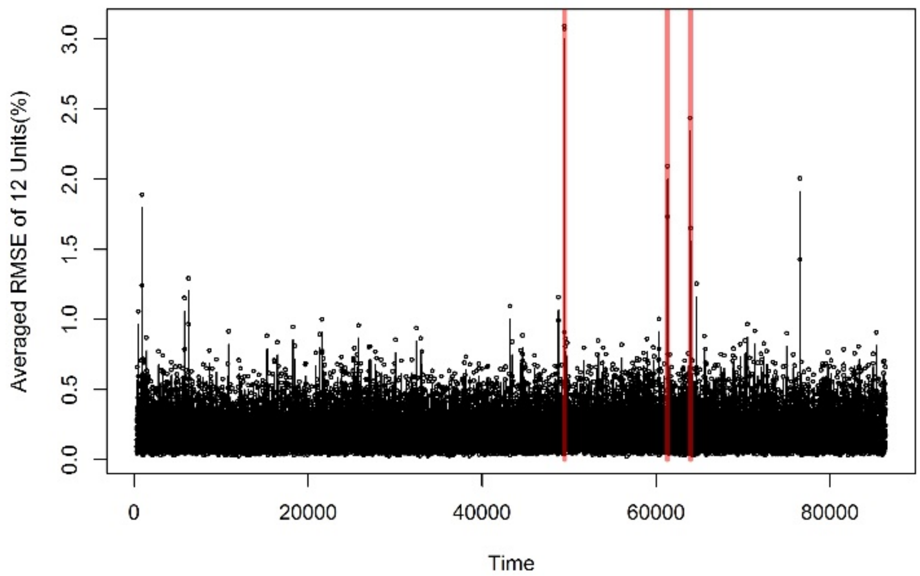

Figure 16. Averaged RMSE across the 12 units between observations and 5-second predictions using the fitted dynamic model. The red vertical lines show temporal locations of recorded events.

Figure 17 illustrates the one-step-ahead predictions of the PMU frequency measurements at Unit 1 using the second order polynomial dynamic model during a 5-min time period, during which no events occurred. We can see that the fitted model well matches historical observations. This is a typical time period with 'normal' variations that can be described and predicted by the second order polynomial dynamic model; when an event occurs, however, the prediction errors will be large and the deviations between the model predictions and observations can be used to quantify the likelihood of an abnormal event, as illustrated in Figure 18. For such an actual event, the deviations or relative errors increase with the time into the events, as shown in Figure 18, where the vertical line corresponds to the event starting time and the blue line denotes observations during the event. The exceedance probability of the relative errors and the duration are compared to the thresholds to confirm anomalies. The proposed framework detected and confirmed this event 4 seconds after it occurs.

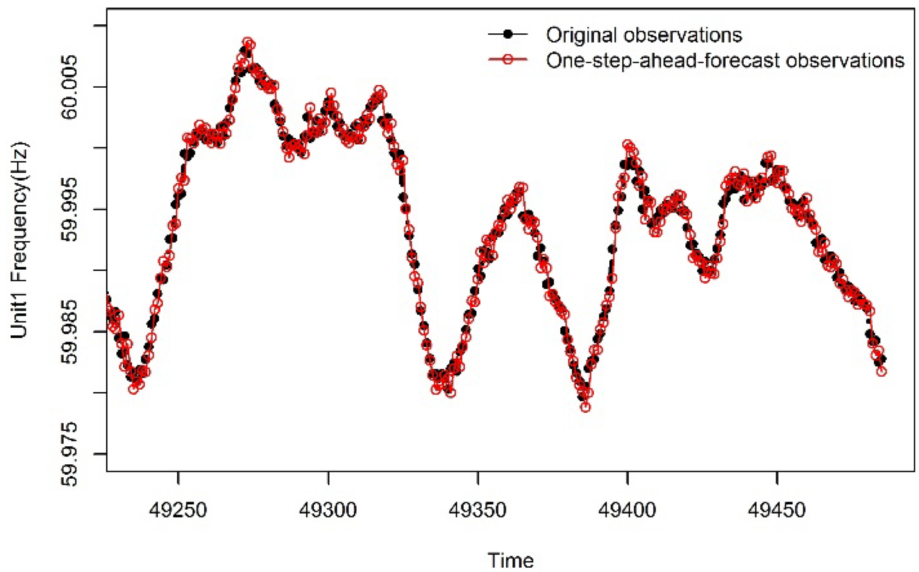

Figure 17. Original PMU frequency measurement at Unit 1 and the predictions with second order polynomial dynamic model

The detection rates of historical recorded events with different combinations of probability and duration thresholds on 28-day PMU data are shown in Figure 19. The optimal thresholds setting seems to exist as follows: The optimal exceedance probability threshold corresponds to 3.5σ (i.e., the prediction error is beyond 3.5 times of the corresponding standard deviation σ). The optimal duration threshold is set to be 5-points (i.e., seconds), which means at least 5 sequential points need to pass the screening in order to confirm an event. A total of 25 historical recorded events during the tested time period were all detected across the 12 Units by the proposed framework using the optimal threshold setting (see the black line in Figure 19).

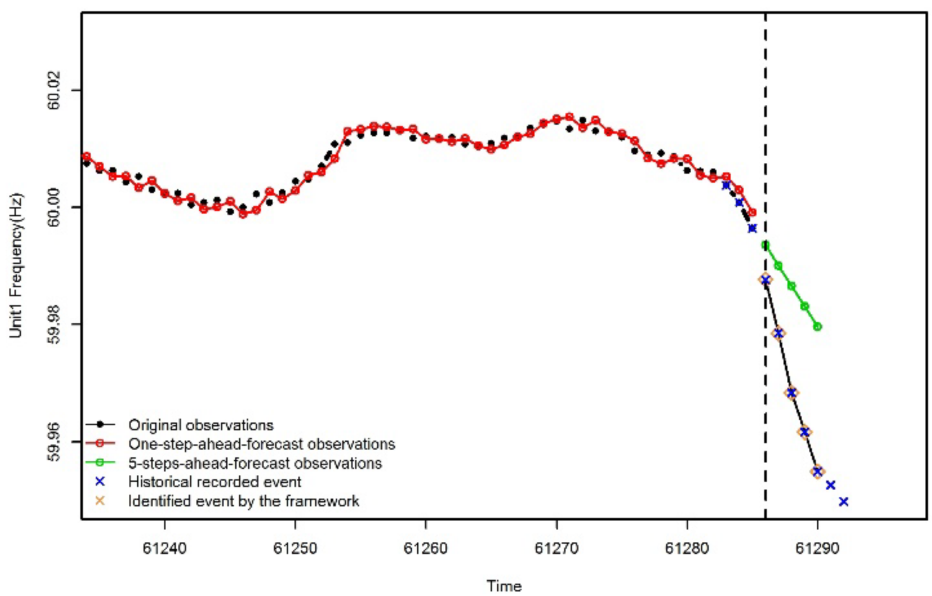

Figure 18. Historical recorded event and anomaly event detected by the framework

This optimum threshold setup allows all of the historical events to be detected (i.e., 100% detection rate or zero false negative rate) with a limited number of total detected event for each day. When the duration threshold is increased to 6 points/seconds, while keeping the probability threshold to 3.5σ, the confidence to confirm two weak events decreases (see the green line in Figure 19). The two events (event #9 and #12) can still be seen at 11 units. The red line in Figure 6 shows the detection results when we keep the duration threshold to be 5 points/seconds but increase the probability threshold to 4.5σ. The overall number of detected events is reduced (~15 per day), which would yield a lower false positive rate. However, in this case, three events (events #9, #12 and #14) can be detected only at 11 units, and events #6 and #25 can be detected only at 10 units. The more rigorous threshold setup lowers the confidence in the confirmed events per unit, but is still adequate to achieve a high detection rate by considering all units for anomaly detection.

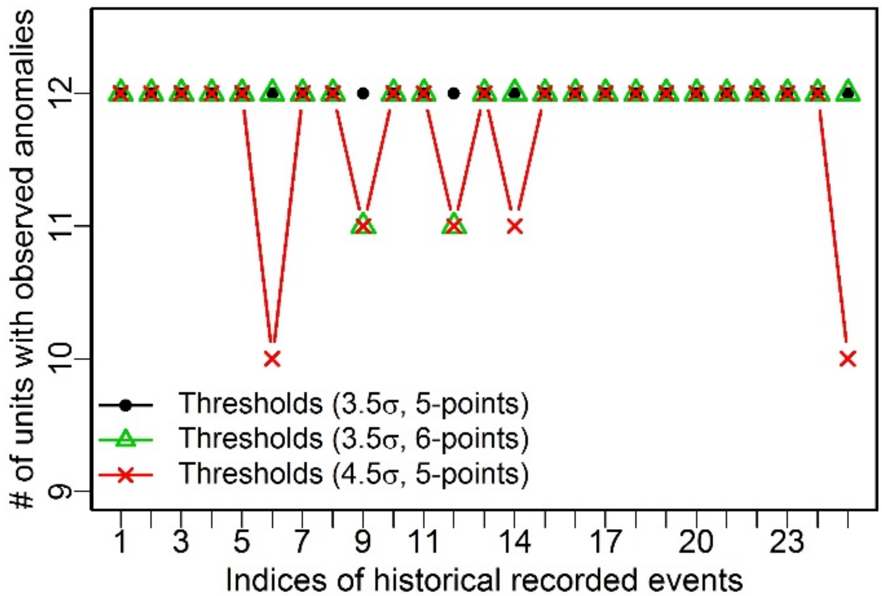

Figure 19. Indices of 25 historical recorded events during the study time period against the number of units where the anomaly event is confirmed, given particular probability and duration threshold settings for anomaly detection.

In this section, a framework for online detection of PMU anomalies with HPC techniques was proposed and developed by implementing dynamic regression models and Kalman filter. The second order of polynomial model was adopted to capture the nonlinear and nonstationary features of the PMU measurements. The model is straightforward and yields adequately low training and prediction errors, which is required for effective and efficient real time anomaly detection. The anomaly detection thresholds can be tuned adaptively to yield optimal anomaly detection and false alarm rates. Overall, the framework is accurate and effective for real-time PMU anomaly detection using the frequency attribute. Abnormal system behaviors can be identified within a few seconds.

5.8 Long Short-Term Memory (LSTM)-based deep neural network (DNN)

Various statistical methods have been developed to forecast PMU time series, such as state space model, regression tree, and support vector machine (Leonardi and Ajjarapu 2010, Gomez et al. 2010, Zheng, Malbasa, and Kezunovic 2012, Liu and Venkatasubramanian 2008, Ren, Hou, and Etingov 2018). The approaches can generally produce accurate short-term (e.g., several time steps ahead) predictions. But for predictions of larger time steps ahead, neural network has been shown in different research areas to be superior to state space models and hidden Markov model, since it enables to capture long-term nonlinear patterns in both spatial and temporal components in the distributed time series (Längkvist, Karlsson, and Loutfi 2014). The flexibility of neural network is that users can develop adaptive and unique models with respect to different datasets and problems from numerous categories of neural network architectures. For instance, Recurrent neural network (RNN) (Connor, Martin, and Atlas 1994) is specially designed for time series analysis by taking temporal sequences as input and output. In practices, an RNN might be difficult to train due to the problem of vanishing gradients which is limited by time windows (Hochreiter et al. 2001). To address this issue, Long Short-Term Memory (LSTM) is designed which can learn to bridge at a minimum 1000 discrete time lags between relevant events by using memory cells to retain information (Hochreiter et al. 2001). LSTM has been successfully used in language modeling (Sundermeyer, Schlüter, and Ney 2012), pattern recognition and image analysis (Chen 2015) involving various time series data. In this section, a Long Short-Term Memory (LSTM)-based deep neural network (DNN) is adopted and evaluated to identify the most appropriate models for event detection and longer-term anomalous pattern extraction. The proposed DNN model show the potential on long-term predictions with the ability to capture nonlinear and nonstationary mixture complex patterns in PMU datasets.

5.8.2 Model architecture

We then design the neural network architecture to train models of different configurations to evaluate the model performances with respect to the configuration parameters such as number of layers, dropout rate, number of units in LSTM layer, batch size, and so on. The architecture of our model is shown in Figure 1, which contains LSTM layers, followed by dropout layers, convolutional layers, and a final output dense layer. This model architecture is flexible by adding layers for each component, for example, one can add multiple LSTM layers to create a stacked LSTM model, where the LSTM layers are "stacked" on top of each other. In this study, each input and predicted output contain the PMU measurements from 12 units, allowing the model to generalize nonlinear connections among the units. Stacked LSTM layers can also take the advantage of the temporal correlations of the measurements to improve model performance. The output dense layer is a neural network where every input neuron is connected to every output neuron with a weight matrix and bias vector. Dropout is a regularization technique that randomly disables a select fraction of neurons during training to enhance robust model performance and prevent overfitting (Hinton et al. 2012b). A 1D Convolutional layer is added next due to its effectiveness when you expect to derive interesting features from shorter (fixed-length) segments of the overall segment is not of high relevance.

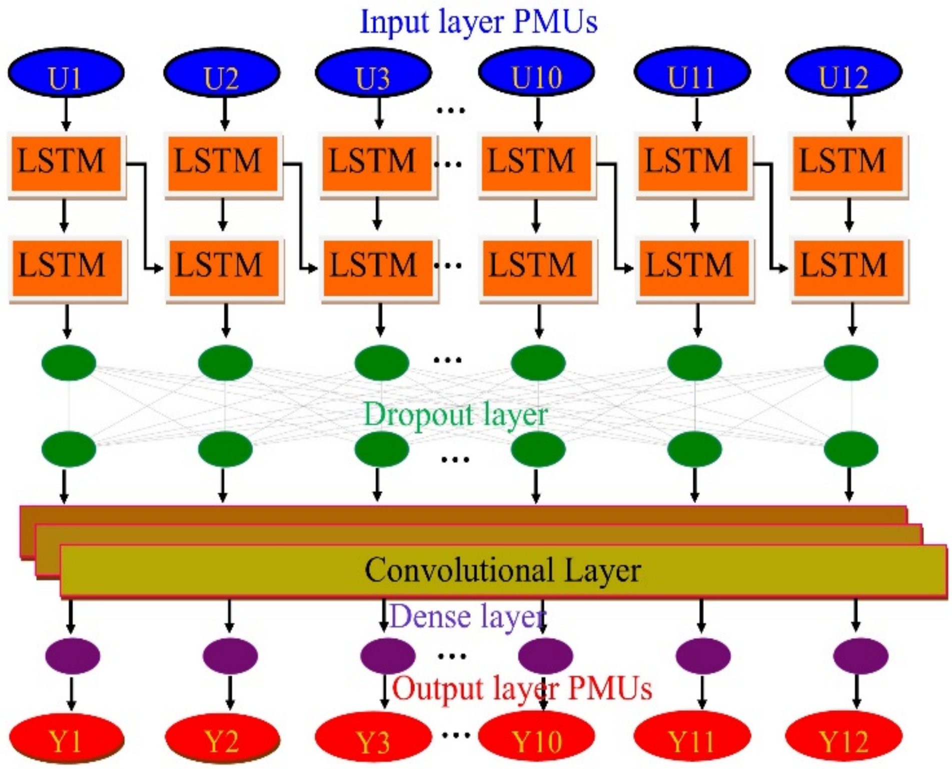

Figure 20. Architecture of the DNN with LSTM, dropout, convolutional and dense layers.

Figure 20 illustrates the standard LSTM cell using three gates including forget, input and output gates, to control the flow of information from one memory cell to another and learn long-term dependencies (Hochreiter and Schmidhuber 1997). LSTM networks have the chain form of repeating modules of neural network same as RNN but contain four neural network layers for each LSTM memory cell. A sigmoid neural net layer with output value between 0 and 1 as well as a multiplication operation is composed to each gate. The cell state is the memory of the LSTM cell which is the key feature to LSTM and the regular RNNs do not have. The horizontal top line across LSTM cell in Figure 21 represents the cell state and the connected three gates are used to protect and control it. A forget gate (*Ft*) layer decides what information to throw away from the previous cell and the decision is made by the sigmoid function composed, where 0 represents completely get rid of the information and 1 represents completely keep it. To determine the information that needs to be updated in the current cell, an input gate (*It*) layer is needed. Combined with *It* layer, a tanh layer through pointwise multiplication generates information to be added to the current cell state. Finally, an output gate (*Ot*) decides the output information based on the input and previous memory state. The sigmoid output layer decides what parts of the cell state will be output, while the tanh layer scales the current cell state by pushing the values to between -1 and 1. The sigmoid output layers multiplied with the tanh layer leads to the output of current cell (Olah 2015, Ma et al. 2015, Malhotra et al. 2015). The output of an LSTM cell is the hidden state, together with the cell input which control what to do with memory.

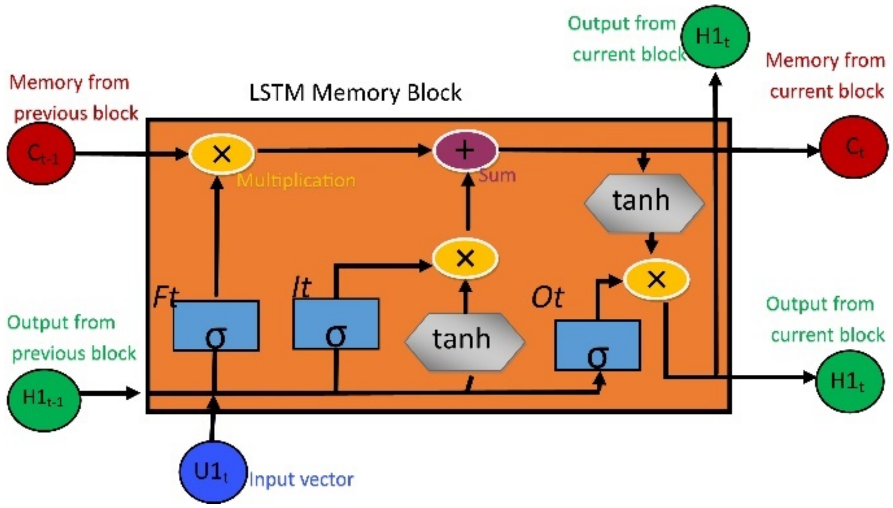

Figure 21. The diagram of LSTM memory cell with the forget (Ft), input (It), and output gate (Ot).

5.8.3 Training and testing models

To build and train the proposed DNN model, Keras (Chollet 2015) is adopted given its highly customizable interface. It also provides an additional layer of DNN primitives with Theano (Al-Rfou et al. 2016) and TensorFlow (Abadi et al. 2016) as its back-end. PMUs time series are split to training, validation, and testing datasets for the 12 units. Training data for the LSTM-based neural network models are created by separating the temporal segments of input and output. The PMUs datasets are preprocessed by normalizing all measurements to be between 0 and 1 with scaling. During the model fitting process, an LSTM model is trained for 10 epochs over the training dataset. The optimizer with adaptive learning rates is chosen to be Adam (Kingma and Ba 2014), the loss function is defined as mean square error (MSE), and activations functions which reduce gradient issues is set to ReLu (Nair and Hinton 2010). The model configuration contains input and output vector length, the number of units in the neural network, layers of stacked LSTM and dropout rate. Given a model configuration, the model is evaluated to predict multiple steps ahead and compared with testing dataset with the sliding window. The accuracy of the model is calculated by mean absolute error (MAE) at each PMU between predicted values and true observations for each prediction length:

$$MAE = \frac{1}{n}\sum_{1}^{n}|Prediction - Observation| \qquad (12)$$

Where n is the testing data length.

5.8.4 LSTM model evaluation

The first 18 hours of one-day PMU frequency attribute is selected for deploying the proposed DNN models. The time period contains three recorded historical events. The first 70% of the data is training set, the next 15% data is used to validate, and the last 15% data is for testing. As a result, there are two historical recorded events in the testing time period. As an example, the frequency time series at Unit#1 is shown in Figure 3a with the three events marked as red vertical lines. The subset testing data is shown in Figure 22b, which clearly shows that the PMU signals are a mixture of variations at multiple temporal scales. The nonstationarity and lack of continuity in the time series make it challenging to perform accurate long-term predictions.

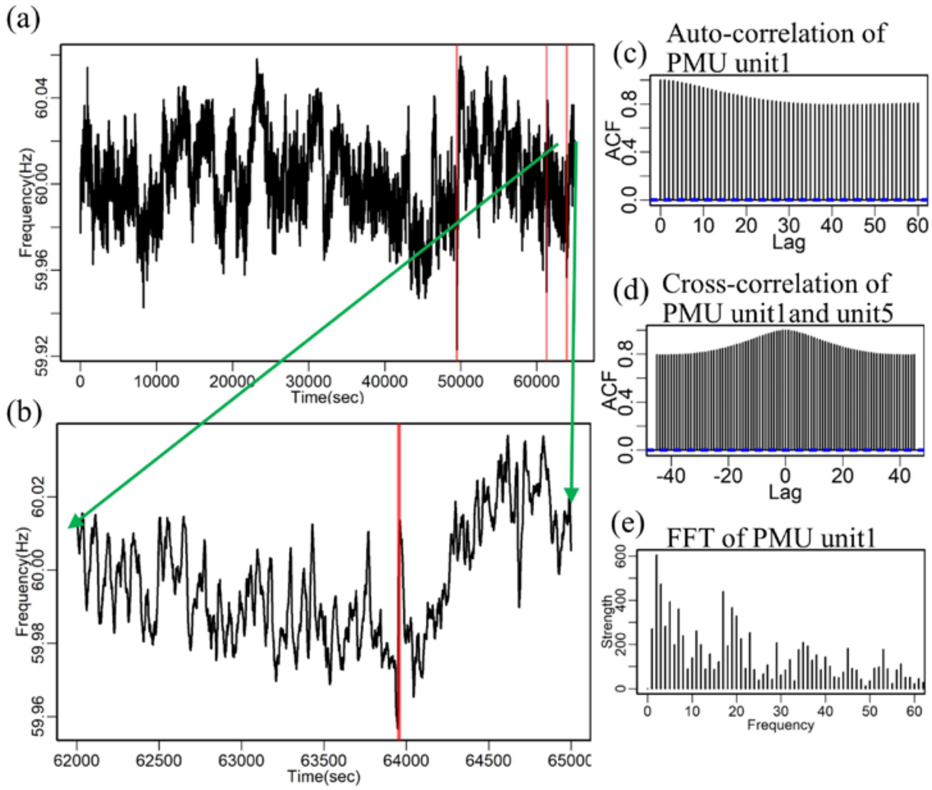

Figure 22. Time series of PMUs frequency and EDA results at Unit #1. The recorded historical events are marked as red lines on the time series.

Time series have been performed on PMU frequency attributes across the units including auto-correlation, cross correlation and Fast Fourier Transform (FFT). All the units yield consistent results, and example results at Unit#1 are shown in Figure 22c-e. It is obvious that strong temporal continuity exists in the PMU dataset which indicates that the time series forecasting is applicable. Approaches from the state space model family, e.g., autoregressive integrated moving average (ARIMA), may work for predictions, but cannot take into account the long-term patterns, e.g., as shown by the second peak in the auto-correlogram indicating non-negligible periodicities in the time series. In fact, the energy spectrum from FFT shows multiple peaks, indicating multiple periodicities at low and high frequencies. The design of the “remembering” previous events in LSTM is suitable for this type of time series. To evaluate the ability of long-term predictions, different DNN model configurations are tested to avoid over fitting and improve the prediction accuracy.

Table 2 summarizes the model evaluation results for each model configurations with a batch size of 30, the number of units of LSTM is 512 and the dropout rate is 0.3 with two-stacked

LSTM layers. The durations of the recorded actual events in the WECC system last between 5 and 20 seconds; therefore, the lengths of our output vectors are set to be within 20 seconds to match the events durations for different DNN model configurations. The model is quite accurate when the output vector length is five-time steps which could serve as the bench mark when different model configurations are compared. It is not surprising that both training and validation error are growing as the output vector length increases. Longer input vector length does not help reduce the training errors but is beneficial for reducing the validation errors when the output vector length is 10. Also, it is noticeable that when the output vector length is 15, the loss errors are ~30% higher than the training errors indicating the model might be overfitting. The model performances are also evaluated with model configurations by fixing the dropout rate to be 0.8, and the number of units to be 128 and 256, for the same input and output vector length as shown in Table 1 to address underfitting and overfitting issues. There seem to be little improvements based on both the training and validation errors by varying the dropout rate and the number of units, but the enhancements are trivial. Thus, the optimal model configuration is having an input vector length of 240 and output vector length of 10 with the dropout rate of 0.3 and containing 512 units in the LSTM neural network. With this optimal model configuration, we can evaluate the accuracy of predictions at different time steps ahead as well as the model architecture.

Table 2. DNN model training and validation against model configuration parameters

Input vector length	Output vector length	Training loss (MSE%)	Validation loss (MSE%)
30	5	0.21	0.24
60	5	0.20	0.24
120	5	0.20	0.23
180	5	0.20	0.25
240	5	0.21	0.24
300	5	0.21	0.25
30	10	0.73	0.93
60	10	0.68	0.83
120	10	0.68	0.90
180	10	0.68	0.89
240	10	0.68	0.86
300	10	0.68	1.06
30	15	1.31	1.42
60	15	1.17	1.47
120	15	1.16	1.46
180	15	1.15	1.51
240	15	1.15	1.56
300	15	1.15	1.59

Three different stacked LSTM layers are evaluated, and the prediction window length is set from 5 to 60 steps ahead. The MAE of each PMU is calculated for each unit and the mean of MAE is obtained among the 12 units as shown in Figure 23. When the prediction time window is shorter than 20 steps, the 3-stacked and 2-stacked LSTM model has the best and worst performances, respectively. All three types of LSTM models have similar level of MAE when the prediction length is around 20 to 30 steps. However, if the prediction window is increasing to longer than 30 steps, the mean MAE raises dramatically for the 3-stacked LSTM. In this case, both the 1-layer and 2-stacked LSTM models still have good performances with MAEs within 1% for up to 60-steps-ahead predictions. For the 2-stacked LSTM model in particular, the MAE is 0.86% for 60-steps-ahead predictions, which is satisfactory for PMU-based mid- and long-term power system planning. The model performances beat the state space models we developed previously (Ren, Hou, and Etingov 2018), particularly for longer-term time series forecasting purpose. For long-term PMUs predictions, we found that the 2-stacked LSTM model is the choice with desired levels of accuracy cross all the PMU units.

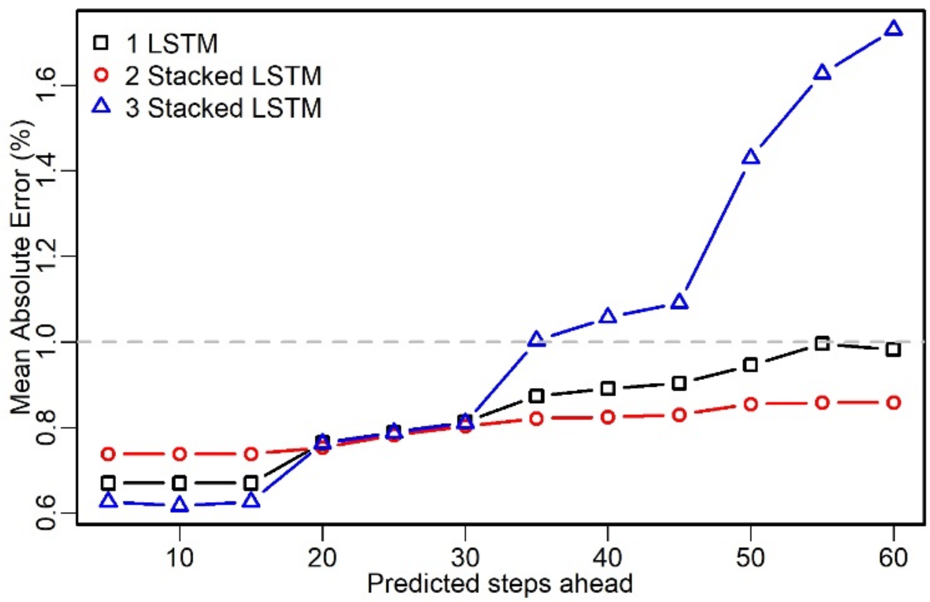

Figure 23. The prediction performance for the three different stacked LSTM models. The grey line corresponds to the error level (MAE) of 1%.

In addition to the statistical errors evaluated for the predictions, the dynamic patterns of the dataset in terms of high nonlinearities and periodicities are also examined. The 30-step-ahead predictions with the 2-stacked LSTM model for PMU Unit #1 is illustrated in Figure 24 with true observations and relative errors as a function of time. The relative errors are calculated using formula (predations–observations)/observations to present the accuracy for each data point in the testing set. It can be seen that the proposed DNN model has the ability to handle the non-stationarity during the relatively long-term prediction time window. The relative errors are within the range of ±0.05% for almost the entire testing time period except for a short time interval around the historical recorded events, which means the events can be detected successfully, even with the relatively long prediction time window.

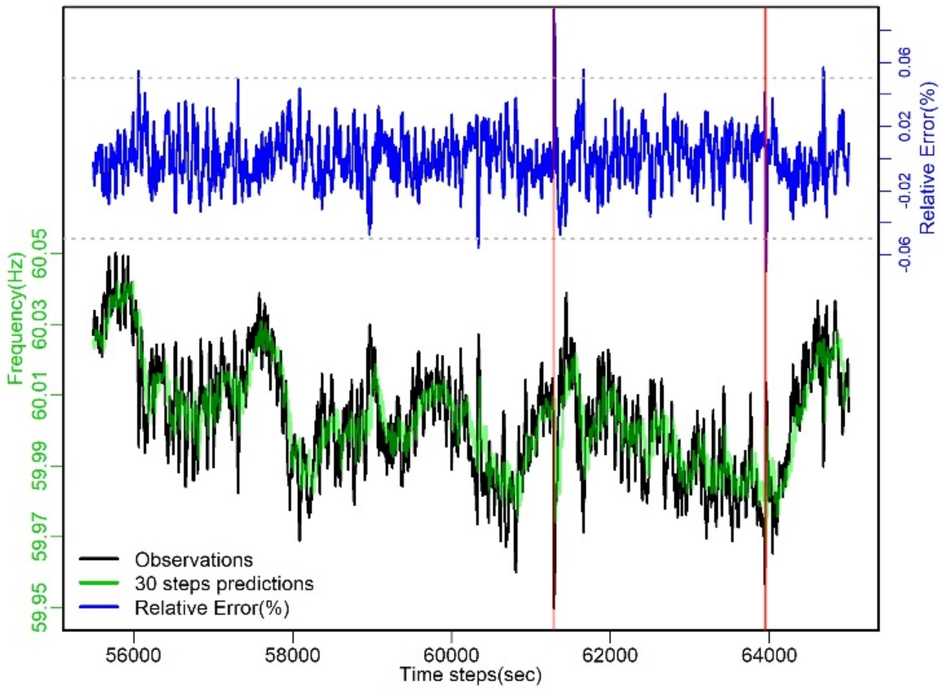

Figure 24. The model prediction of PMU frequency at Unit #1 for 30-step-ahead using the optimal model configuration with 2-stacked LSTM layers. The black line is the frequency observations; the green line is the model predictions; the blue line is the relative errors (%) between the predictions and observations. The historical recorded events are marked as red vertical lines.

In this section, a methodology for frequency predictions for the mid- and long-term time window and detection of abnormal events has been developed. In this study, we adopted the LSTM-based DNN models to predict multiple steps ahead and detect events by address both the spatial and temporal variations in PMU time series. Different model configurations were evaluated to yield an optimal model parameter set for the high resolution, complex, and dynamic PMU dataset. The prediction ability was also examined with various forecasting time window sizes. The results showed that satisfactorily low prediction errors can be achieved in the prediction up to 60 steps ahead. The decent relative error is obtained at each testing point which can be used for the abnormal event detections.

Due to the computational demand, we developed and tested the LSTM-based DNN framework using one-day data in the current study. For the future work, the model transferability will be tested on PMU dataset across dates and seasons, by integrating more data in the training model. We also fully evaluate the different DNN model configurations for predicting with even longer time windows, and to explore the upper limit of predictability of PMU data with deep learning and the most advanced computational resources involving graphical processing units (GPUs).

5.9 Events classification and localization through Convolutional Neural Network (CNN)

Observation-based event detection, classification, and localization using real world data are usually challenging due to lack of labelled data. Such data inadequacy cannot support training of deep neural networks. One solution is by way of data augmentation by generating ensemble simulation-based training dataset. In this study, we evaluate the feasibility of using ensemble simulation-based data with various types of faults, this will provide adequate labelled data for supervised learning. Dynamic simulations have been performed on a Polish 3120-bus system with four types of faults in five geophysical zones. The multi-channel time series of machine speed data are extracted and encoded to images for evaluating the feasibility of CNN models for classification (fault types) and localization (occurred zones). Time series stacking is straightforward, which requires the least computational resources—it serves as the benchmark dataset for our proposed CNN model. Three other time series encoding approaches including time-domain stacking, wavelet decomposition-based frequency-domain stacking and polar coordinate system-based GAF stacking, are adopted to evaluate and compare the CNN model performances. VGG model architecture is adopted in our CNN to take the advantage of pushing model depth towards a high accuracy (Simonyan and Zisserman 2014).

5.9.2 Polish system testbed and data preparation

Simulation tests are performed on the 3120-bus Polish system (“case3120sp.m”) (MATPOWER 2008). Dynamic data have been developed in PSS/E format that includes generator, governor, stabilizer and exciter models for generator dynamics. Several protection models are prepared in PSS/E format; this includes protection models including distance relays, generator-based voltage and frequency relays, and under frequency and under voltage load-shedding relays and are added to the existing relay protection models in the Polish case. The Polish system has 3120 buses, 3487 branches, 2314 loads, ,505 generators and consists of 5 zones. The zonal model for total generations of each zone and the connections between zones is illustrated on Figure 25. Zone 3 is the largest zone connected with zone 1, 2 and 4. While the zone 5 is the smallest zone with only 841MW which has the connection with zone 1 and 4 only.

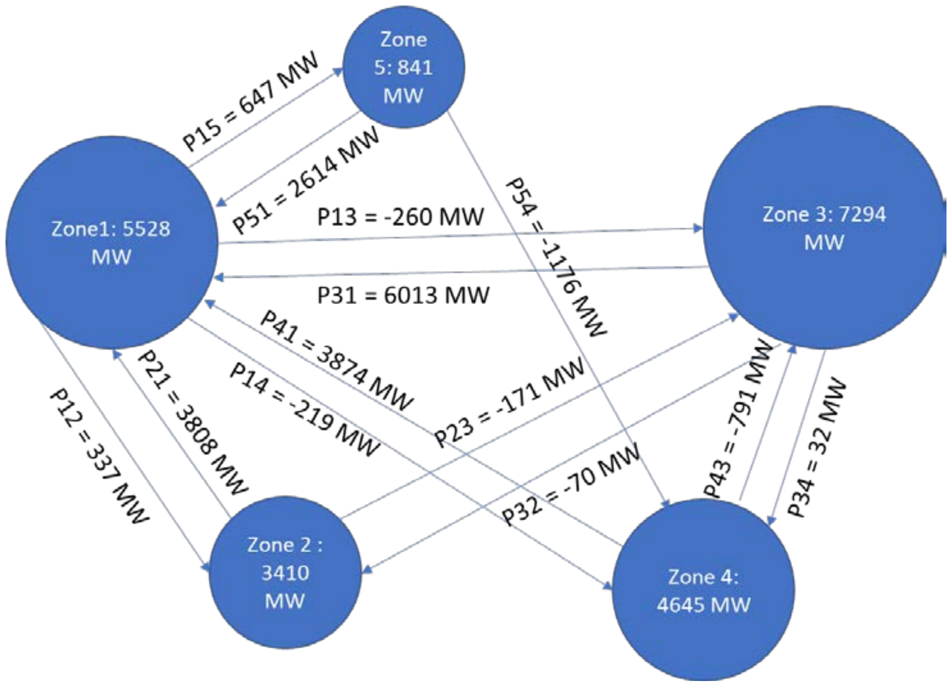

Figure 25. Zonal model of Polish system.

5.9.3 Fault types and implementation

Various power system faults were simulated using the PSS/E software. These faults can be categorized into four types and the outages characteristics of each type have been summarized in Table 3. The Fault 1 is the single generator and multiple line fault. In this scenario, each generator and consequently with two transmission lines were tripped in the system. A total of 298 contingencies were simulated for Fault 1. For Fault 2, a three-phase fault at a bus during dynamic simulations was applied to polish system with 3120 contingencies. Different numbers of generations were set to out-of-service during dynamic simulation for the system as Fault 3 and 4 with 298 contingencies for each fault type. The former scenario set each generator out-of-service and the latter one has the two neighboring generators out-of-service.

Table 3. Fault types simulated in polish system

Types	Outage of single generator	Outage of two generators	Three-phase bus fault	Multiple line fault	Total contingencies
Fault 1	✓			✓	298
Fault 2			✓		3120
Fault 3	✓				298
Fault 4		✓			298

5.9.4 CNN model development

We design the CNN architecture to train models for faulty type prediction and zonal classification for the Polish 3120-bus system. The inputs to the CNN model training are the image sets produced by three different data encoding schemes. Model training and testing are done for each data encoding scheme separately and the corresponding model performances are then compared. Each image data set has been divided into three independent subsets: training, validation, and testing. Training and validation are used during model development for determining the optimized model configuration and hyperparameters. The training data also include the multi-class labels data which has been determined during the ensemble simulation setup. The multi-class labels are fault types for fault classification and can be fault locations or zones for training CNN models for approximating fault locations.

The CNN model architecture conducted in this study is based on Visual Geometry Group 11 (VGG11) (Simonyan and Zisserman 2014) illustrated in Figure 26. The input to the first 2D convolutional layer is fixed size 128 x 128 RGB image sets. Each image is passed through a series of blocks of convolutional layers (orange layers). Totally three sets of convolutional layers are adopted, which contain 2, 3, and 3 consecutive convolutional layers respectively. Different numbers of neural nodes are used for convolutional blocks and the receptive field is consistent for each convolutional layer. Spatial pooling is carried out by three max-pooling layers, following each convolutional block. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation demand in the network. Max-pooling is performed over a 2×2-pixel window, with a stride of 2. A drop out layer follows each max-pooling layer. Dropout is a regularization technique that randomly disables a selected fraction of neurons during training to enhance robust model performance and prevent overfitting (Hinton et al. 2012a). The output from the three blocks of convolutional layers is converted into 1D feature array by flattening each layer to feed the next layers. Finally, three fully connected (FC) layers, also known as dense layers, are added followed by the soft-max activation to yield multi-class predictions in the end. The soft-max layer outputs the values between 0 and 1 quantifying the probability and confidence of which class each image belongs to. ReLU (i.e., rectified linear unit) activation is one of the most commonly used activation functions in neural networks especially in CNNs with its linear property for positive values and zero for negative inputs. In our CNN model architecture, ReLU activation function is deployed in each layer except for the last dense layer. The optimizer is Stochastic Gradient Descent (SGD) which estimates the error gradient for the current state of the model using the training dataset, then updates the weights of the model with backpropagation. The categorical cross-entropy class is chosen for the multi-label classification problems. It computes the cross-entropy loss between the labels and model predictions, and calculation of the loss function requires that the last dense layer is configurated with the total number of classes which allows soft-max activation to predict the probability for each class.

Figure 26. The CNN model architecture.

5.9.5 CNN model evaluation

To evaluate the CNN model performance with respect to zonal and faulty classification, confusion matrix is introduced for multinomial classification. The confusion matrix provides the numbers of the target class values that are assigned to the positive and the negative class. Four types of events are counted for multi-class confusion matrix including: (1) True Positive (TP) which is the cells identified by the row and column for the positive class and correctly classified as such. The TP cells are located at top left corner of the confusion matrix; (2) False Negative (FN) is recognized as the row for the positive class and columns for the negative class. It belongs to the positive class and incorrectly classified as negative which is located at the top right cells of the confusion matrix; (3) False Positive (FP) is determined by rows for the negative class and the column for the positive class. It belongs to the negative class and incorrectly classified as positive which is placed in the lower left on the confusion matrix; (4) True Negatives (TN) is known with the cells outside the row and column for the positive class. It is belonging to the negative class and correctly classified as such. It is placed at the lower right of the confusion matrix. Using the four counts in the confusion matrix, the class statistics metrics can be calculated to quantify the model performance. Sensitivity measures how proper the model is to detecting events in the positive class. Specificity measures how exact the assignment to the positive class is. Precision measures how good the model is at assigning positive events to the positive classes. Accuracy represents the percentage of correctly classified applications compared with the total number of applications.

The well-trained CNN model has been applied on the third independent testing dataset for zonal classification using time-stacking images. The class statistics are calculated and illustrated in Figure 27. The overall averaged accuracy reaches 91% with the straightforward time series stacking approach. The zone 1 and zone 3 have the lowest and highest accuracy which are 88% and 96%. The specificity is high for all zones with averaged value 97% which means the model has satisfied performance on assigning the extract the positive class. For sensitivity and precision measurements, the averaged values are about 86% for both and zone 3 stands out among five zones. All the measured metrics show the high performance level which means the polish system has well defined zonal structures which makes the CNN model can accurately predict the fault locations. Generally, the fault added on other zones outside of the target zone can impact the power flows in the target zones if the zones are physically connected with each other. The physical connections and impacts between zones will slightly increase the FT and FN in model predictions. However, if the zones are not connected, the model can provide accurate TN predictions. Zone 1 has the worst prediction results especially for sensitivity and accuracy since the model tends to predict a small portion of zone 1 events to the rest zones, especially to zone 2 and zone 5. One of the reasons might be that zone 1 has the connections with all the rest zones which means the flows within the zone 1 is complex and can be impact by the faults added to other zones. As the largest zone, zone 3 has no transmission lines going to zone 5 and our model results has shown that the none of the predictions of zone 3 fall in zone 5. Similar pattern can be noticed for the model predictions for zone 5 that there is no prediction goes to zone 2 or 3 because zone 5 does not have the physical connections with these two zones.

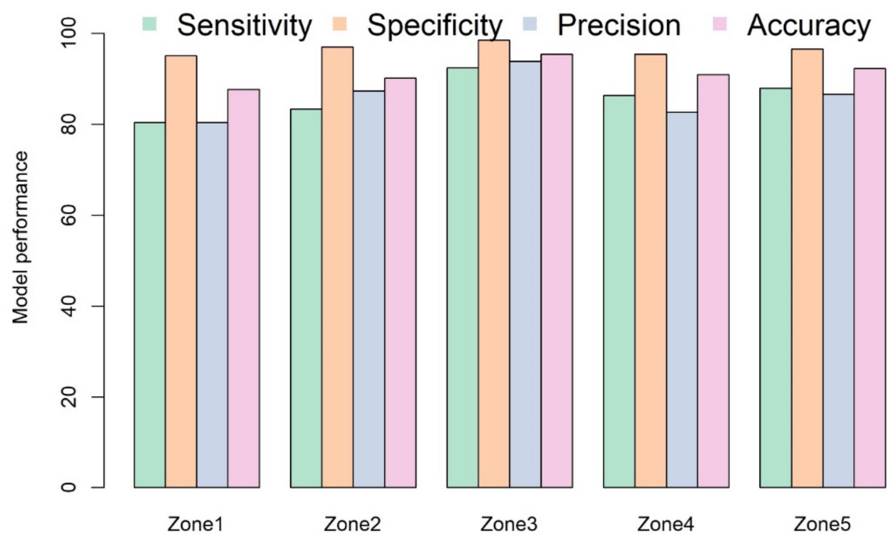

Figure 27. Zonal CNN model performance confusion matrix using time-domain stacking encoding approach.

6.0 Conclusions

The framework for PMU data analysis based on the Apache Spark technology has been developed and tested using real system synchrophasor measurements. Software modules to efficiently read large volumes of PMU information from BPA PDAT files and pre-process raw data for event detection have been implemented using the Python programming language and the PySpark interface. We showed that application of a cloud-based platform and Apache Spark significantly increases the computational throughput of the application. Analysis of several months of PMU data using the 20 node computer cluster with commodity hardware takes up-to several hours. By comparison, a similar job executed on a single personal computer can take several days to complete.

The team developed and adopted multiple metrics to evaluate and understand the data-to-day variations and similarities of multiple PMU attributes. Such metrics included cross-correlations, Euclidean-distanced clusters, and Taylor Diagrams (it integrates/compares differences in phase angle and magnitude, and combines Pearson correlation, root-mean-square error, and individual standard deviations). Performed comprehensive spatiotemporal analyses can help for PMU data reconstruction or simulation, which could enable development of next-generation PMU-focused machine learning algorithms.

We adopted the LSTM-based DNN models to predict multiple steps ahead and detect abnormal events by address both the spatial and temporal variations in PMUs. Different model configurations were evaluated to yield an optimal model parameter set for the high resolution, complex, and dynamic PMU dataset. The decent relative error is obtained at each testing point which can be used for the abnormal event detections.

We proposed and developed framework features a scoring system for the anomaly detection. We also evaluated the effect of an alternative scoring system on our detection framework. Compared with the additive scoring system, the alternative multiplicative scoring system is much stricter in that the score for each MRA scale at each unit is multiplied to obtain scores of either 0 or 1. Using this multiplicative scoring system, both false alarm rate and detection rate were reduced.

We developed and evaluated a deep learning CNN model to identify locations and predict types of various faults in the Polish 3120-bus test system. The Polish system in different spatial zones and locations was simulated with four distinct types of faults, and the outputs provided adequate and balanced data for CNN training and testing. Our CNN is composed of a number of layers, including convolutional, pooling, dropout and dense layer, which are designed to adaptively learn spatial hierarchies of features. Hyperparameter search were performed to determine the optimal model configuration and the final model for fault classification and prediction.

In future work, we plan to continue both a mathematical and software enhancement of this framework’s functionality by adding new analytical modules and additional data sources, like supervisory control and data acquisition (SCADA) data. We are also going to shift from PNNL institutional cloud to Amazon AWS or Microsoft Azure cloud platform to improve computational performance of the developed analytical framework.

7.0 References

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard. 2016. "Tensorflow: A system for large-scale machine learning." 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16).

Akhavan-Hejazi, Hossein, and Hamed Mohsenian-Rad. 2018. "Power systems big data analytics: An assessment of paradigm shift barriers and prospects." *Energy Reports* 4:91-100. doi: https://doi.org/10.1016/j.egyr.2017.11.002.

Al-Rfou, Rami, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, and Alexander %J arXiv preprint arXiv:.02688 Belopolsky. 2016. "Theano: A Python framework for fast computation of mathematical expressions."

Alarcon-Aquino, Vicente, and Javier A Barria. 2001. "Anomaly detection in communication networks using wavelets." *IEE Proceedings-Communications* 148 (6):355-362.

Apache Hadoop Documentation. 3.0. Apache Software Foundation, Wakefield, MA.

Apache Spark Documentation 2.4. Apache Software Foundation, Wakefield, MA.

Aradhye, Hrishikesh B, Bhavik R Bakshi, James F Davis, and Stanley C Ahalt. 2004. "Clustering in wavelet domain: A multiresolution art network for anomaly detection." *AIChE journal* 50 (10):2455-2466.

Avdaković, Samir, and Nejra Čišija. 2015. "Wavelets as a tool for power system dynamic events analysis–State-of-the-art and future applications." *Journal of Electrical Systems and Information Technology* 2 (1):47-57.

Benedetto, John J, and Shidong Li. 1998. "The theory of multiresolution analysis frames and applications to filter banks." *Applied and Computational Harmonic Analysis* 5 (4):389-427.

Bhuyan, Monowar H, Dhruba Kumar Bhattacharyya, and Jugal K Kalita. 2014. "Network anomaly detection: methods, systems and tools." *Ieee communications surveys & tutorials* 16 (1):303-336.

Burrus, C Sidney, Ramesh A Gopinath, and Haitao Guo. 1997. "Introduction to wavelets and wavelet transforms: a primer."

Chen, Chi-hau. 2015. *Handbook of pattern recognition and computer vision*: World Scientific.

Chollet, François. 2015. Keras.

Cloudera User Guide. Cloudera, Inc, Palo Alto, CA.

Connor, Jerome T, R Douglas Martin, and Les E %J IEEE transactions on neural networks Atlas. 1994. "Recurrent neural networks and robust time series prediction." 5 (2):240-254.

Daubechies, Ingrid. 1992. *Ten lectures on wavelets*: SIAM.

Faris, Tony. "BPA Synchrophasor Lab Tools." *WECC JSIS meeting, September, 2016* [Online] Available: https://www.wecc.biz/Administrative/18%20BPA%20SP%20LabTools-2016_09.pdf.

Gaouda, AM, MMA Salama, MR Sultan, and AY Chikhani. 1999. "Power quality detection and classification using wavelet-multiresolution signal decomposition." *IEEE Transactions on power delivery* 14 (4):1469-1476.

Gomez, Francisco R, Athula D Rajapakse, Udaya D Annakkage, and Ioni T %J IEEE Transactions on Power Systems Fernando. 2010. "Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements." 26 (3):1474-1483.

Hinton, Geoffrey E, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012a. "Improving neural networks by preventing co-adaptation of feature detectors." *arXiv preprint arXiv:1207.0580*.

Hinton, Geoffrey E, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R %J arXiv preprint arXiv:. Salakhutdinov. 2012b. "Improving neural networks by preventing co-adaptation of feature detectors."

Hochreiter, Sepp, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. A field guide to dynamical recurrent neural networks. IEEE Press.

Hochreiter, Sepp, and Jürgen %J Neural computation Schmidhuber. 1997. "Long short-term memory." 9 (8):1735-1780.

IEEE. 2011. "IEEE Standard for Synchrophasor Data Transfer for Power Systems." *IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*:1-53. doi: 10.1109/IEEESTD.2011.6111222.

Kezunovic, Mladen, Pierre Pinson, Zoran Obradovic, Santiago Grijalva, Tao Hong, and Ricardo Bessa. 2020. "Big data analytics for future electricity grids." *Electric Power Systems Research* 189:106788. doi: https://doi.org/10.1016/j.epsr.2020.106788.

Kingma, Diederik P, and Jimmy %J arXiv preprint arXiv:. Ba. 2014. "Adam: A method for stochastic optimization."

Konakalla, Sai Akhil R, and Raymond de Callafon. 2016. "Optimal filtering for grid event detection from real-time synchrophasor data." *Procedia Computer Science* 80:931-940.

Längkvist, Martin, Lars Karlsson, and Amy %J Pattern Recognition Letters Loutfi. 2014. "A review of unsupervised feature learning and deep learning for time-series modeling." 42:11-24.

Leonardi, Bruno, and Venkataramana %J IEEE Transactions on Power Systems Ajjarapu. 2010. "Development of multilinear regression models for online voltage stability margin estimation." 26 (1):374-383.

Liu, Guoping, and Vaithianathan Venkatasubramanian. 2008. "Oscillation monitoring from ambient PMU measurements by frequency domain decomposition." 2008 IEEE International Symposium on Circuits and Systems.

Lounsbery, Michael, Tony D DeRose, and Joe Warren. 1997. "Multiresolution analysis for surfaces of arbitrary topological type." *ACM Transactions on Graphics (TOG)* 16 (1):34-73.

Lu, Wei, and Ali A Ghorbani. 2009. "Network anomaly detection based on wavelet analysis." *EURASIP Journal on Advances in Signal Processing* 2009:4.

Ma, Xiaolei, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng %J Transportation Research Part C: Emerging Technologies Wang. 2015. "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data." 54:187-197.

Malhotra, Pankaj, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. "Long short term memory networks for anomaly detection in time series." Proceedings.

Mallat, Stéphane. 1999. *A wavelet tour of signal processing*: Academic press.

Mallat, Stephane G. 1989. "A theory for multiresolution signal decomposition: the wavelet representation." *IEEE transactions on pattern analysis and machine intelligence* 11 (7):674-693.

MATPOWER. 2008. "MATPOWER case archive – “Case3120sp.m” – 2008 summer peak Polish Case ".

Meyer, Yves. 1985. "Principe d'incertitude, bases hilbertiennes et algebres d'operateurs." *Séminaire Bourbaki* 28:209-223.

Nair, Vinod, and Geoffrey E Hinton. 2010. "Rectified linear units improve restricted boltzmann machines." Proceedings of the 27th international conference on machine learning (ICML-10).

Olah, Christopher. 2015. "Understanding lstm networks."
OpenStack User Guide. OpenStack Foundation.
Petris, G. 2010. "An R Package for Dynamic Linear Models." *Journal of Statistical Software* 36 (12):1-16.
Petris, Giovanni, Sonia Petrone, and Patrizia Campagnoli. 2009. "Dynamic linear models." *Dynamic Linear Models with R*:31-84.
Phadke, Arun G, and John Samuel Thorp. 2008. *Synchronized phasor measurements and their applications*: Springer Science & Business Media.
PNNL. 2020. Electricity Infrastructure Operations Center. Richland, WA.
Ren, Huiying, Zhangshuan Hou, and Pavel Etingov. 2018. "Online Anomaly Detection Using Machine Learning and HPC for Power System Synchrophasor Measurements." 2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS).
Ren, Huiying, Zhangshuan Hou, Heng Wang, Dimitri Zarzhitsky, and Pavel Etingov. 2018. "Pattern Mining and Anomaly Detection based on Power System Synchrophasor Measurements." Proceedings of the 51st Hawaii International Conference on System Sciences (HICSS).
Simonyan, Karen, and Andrew Zisserman. 2014. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556*.
Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. 2012. "LSTM neural networks for language modeling." Thirteenth annual conference of the international speech communication association.
Šutić, D., and E. Varga. 2017. "Apache spark as distributed middleware for power system analysis." 2017 25th Telecommunication Forum (TELFOR), 21-22 Nov. 2017.
Vajjala, V. A. H. 2016. "A novel solution to use Big Data technologies and improve demand response program in aggregated residential houses." 2016 IEEE Conference on Technologies for Sustainability (SusTech), 9-11 Oct. 2016.
Wang, B., B. Fang, Y. Wang, H. Liu, and Y. Liu. 2016. "Power System Transient Stability Assessment Based on Big Data and the Core Vector Machine." *IEEE Transactions on Smart Grid* 7 (5):2561-2570.
Wang, Wei, Dunqiang Lu, Xin Zhou, Baoju Zhang, and Jiasong Mu. 2013. "Statistical wavelet-based anomaly detection in big data with compressive sensing." *EURASIP Journal on Wireless Communications and Networking* 2013 (1):269.
Zheng, Ce, Vuk Malbasa, and Mladen %J IEEE Transactions on Power Systems Kezunovic. 2012. "Regression tree for stability margin prediction using synchrophasor measurements." 28 (2):1978-1987.
Zhou, D., J. Guo, Y. Zhang, J. Chai, H. Liu, Y. Liu, C. Huang, X. Gui, and Y. Liu. 2016. "Distributed Data Analytics Platform for Wide-Area Synchrophasor Measurement Systems." *IEEE Transactions on Smart Grid* 7 (5):2397-2405.
Zhou, G., D. Zhao, K. Zou, W. Xu, X. Lv, Q. Wang, and W. Yin. 2015. "The static security analysis in power system based on Spark Cloud Computing platform." 2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA), 3-6 Nov. 2015.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354
1-888-375-PNNL (7665)

www.pnnl.gov