8-2021

Advanced Wide-Area Monitoring System Design, Implementation, and Application

Weikang Wang
wwang72@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Weikang Wang entitled "Advanced Wide-Area Monitoring System Design, Implementation, and Application." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Engineering.

Yilu Liu, Major Professor

We have read this dissertation and recommend its acceptance:

Fangxing Li, Hairong Qi, Wenxuan Yao, Yilu Liu

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Advanced Wide-Area Monitoring System Design, Implementation, and Application

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Weikang Wang

August 2021

Whatever you are, be a good one.

ACKNOWLEDGEMENTS

I deeply appreciate those who has helped me towards getting the Ph. D. in the University of Tennessee and anywhere else in this lovely world.

I would like to give my deepest thanks to my advisor, Dr. Yilu Liu for her careful, patient, and insightful guidance and heart-felt support during my pursuit of the Ph. D. degree. Thank you. In the meantime, I would like to thank my dissertation committee members, Dr. Fangxing Li, Dr. Hairong Qi, and Dr. Wenxuan Yao. I appreciate your constructive suggestions on composing my research and this dissertation.

I would like to give my thanks to members of Power IT Laboratory, Dr. Dao Zhou, Dr. Ling Wu, Dr. Jiecheng Zhao, Dr. Shutang You, Dr. Wenpeng Yu, Dr. Lin Zhu, Dr. He Yin, Dr. Kaiqi Sun, Dr. Fuhua Li, Dr. Hesen Liu, Dr. Yi Cui, Dr Feifei Bai, Dr. Yao Zhang, Dr. Micah J. Till, Abbigail Till, Chujie Zeng, Xuemeng Zhang, Yu Su, Yingfeng Zhao, and Wei Qiu.

I also would like to thank Warren Wu, Darrell Moore, Bill Graham, Brent Kane, Ara Johns, Bob Cummings, Howard Gugel, Dr. Ryan Quint and Dr. James for your supports on my research and projects.

In addition, I would like to thank all my friends from Christian Churches for your friendships. I would like to give special thanks to Pastor Dr. Chang-wen Chen, Connie Chen, Tom Fethe, and Becky Fethe for their kindness and supports.

Finally, I would like to give my deepest love to my Wife Candice (Chang) Chen, my father Wenquan Wang, my mother Juan Zhang, my grandfather Yunjie Wang, and my Grandmother Shufang Xiao. I could never achieve all the accomplishments without your love and supports.

ABSTRACT

Wide-area monitoring systems (WAMSs) provide an unprecedented way to collect, store and analyze ultra-high-resolution synchrophasor measurements to improve the dynamic observability in power grids. This dissertation focuses on designing and implementing a wide-area monitoring system and a series of applications to assist grid operators with various functionalities. The contributions of this dissertation are below:

First, a synchrophasor data collection system is developed to collect, store, and forward GPS-synchronized, high-resolution, rich-type, and massive-volume synchrophasor data. a distributed data storage system is developed to store the synchrophasor data. A memory-based cache system is discussed to improve the efficiency of real-time situation awareness. In addition, a synchronization system is developed to synchronize the configurations among the cloud nodes. Reliability and Fault-Tolerance of the developed system are discussed.

Second, a novel lossy synchrophasor data compression approach is proposed. This section first introduces the synchrophasor data compression problem, then proposes a methodology for lossy data compression, and finally presents the evaluation results. The feasibility of the proposed approach is discussed.

Third, a novel intelligent system, SynchroService, is developed to provide critical functionalities for a synchrophasor system. Functionalities including data query, event query, device management, and system authentication are discussed. Finally, the resiliency and the security of the developed system are evaluated.

Fourth, a series of synchrophasor-based applications are developed to utilize the high-resolution synchrophasor data to assist power system engineers to monitor the performance of the grid as well as investigate the root cause of large power system disturbances.

Lastly, a deep learning-based event detection and verification system is developed to provide accurate event detection functionality. This section introduces the data preprocessing, model design, and performance evaluation. Lastly, the implementation of the developed system is discussed.

PREFACE

Before I started my PhD study in engineering, there were some arguments on the what would such a PhD degree program look like. There are some people holding a belief that an engineering PhD degree program should only graduate those who can deduct complex functions, publish tons of papers, and advance an area from the theoretical realm. On the opposite side, some would suggest PhD students focus on practical things and be down-to-earth people. Through the past years of my PhD study, I never find any of these beliefs works alone. I have seen many PhD students lose their passion of research by forcing themselves to study complex theoretical problems and publish papers. As my PhD supervisor says, *you don't want to do research to publish papers*, *or you will eventually hate yourself*. Surprisingly, as opposed to that, there have been equally same amount of people who lose their passions as well by doing similar engineering projects again and again.

To me, the nature of my research is solely driven by the passion and the desire to create something meaningful to the industry. Most of my research topics are rooted from practical projects, while proposing complex algorithms and publishing papers are always the least considerations. Surprisingly, I always find there are things I can, or I should, do to improve the existing technologies and they are usually complex as well. Trying to address these issues are enjoyable because it is likely that I will apply what I develop to move the industry one step further.

I love what I have been doing because they might benefit hundreds of thousands of people out there on this planet. This is why I gave up the computer science master offer from Columbia University several years ago and chose to become a PhD student at the University of Tennessee.

Life is too short, why not dream big.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE
INTRODUCTION OF THE MODERN WIDE AREA MONITORING SYSTEM

1.1 Phasor measurement unit

Phasor measurement units (PMU) provide high-resolution, rich-type and large-volume measurements in modern power systems to support situation awareness, analytics, control, and protection. The first PMU was invented by the Virginia Polytechnic Institute and State University in 1990s [1]. Since then, many have improved the original prototype by proposing better phasor estimation algorithms and synchronization methods. Nowadays, PMUs utilize different phasor measurement techniques and the Global Positioning System (GPS) to provide synchronized phasor measurements (synchrophasors). Due to the wide recognition of PMUs, in recent years, they have been increasingly deployed in the wide area measurement systems (WAMS) [2]. Due to the high cost of conventional PMUs, in recent years, some micro-PMUs are developed to provide equally high-resolution and highly accurate synchrophasor measurements but in much lower costs. In the North American grids, there are several types of widely used micro-PMUs, including frequency disturbance recorder (FDR) [3], universal grid analyzer (UGA) [4], and microPMU (μPMU) [5]. At present, the PMUs and the micro-PMUs are widely installed in the worldwide power systems. Figure 1-1 shows a map of deployed micro-PMUs in the distribution-level synchrophasor system, FNET/GridEye.

Figure 1-1 Deployment map of FNET/GridEye in North America

1.2 Phasor data concentrator

A phasor data concentrator (PDC) is a combination of hardware and software system that collects and processes synchrophasor measurement data. According to the IEEE **C37.247-2019 [6]** protocol, a PDC provides functionalities including data collection, data forwarding, data processing, data analytics, and data storage. In modern WAMSs, PDCs can be configured via multiple communication schemes [7]. A PDC can directly collects measurement data from PMUs or other PDCs depending on its configuration level. With the advances in computer technology, both academia [8], [9] and industry have proposed and implemented some PDC systems [10], [11].

Due to the increasing deployment of PMUs and micro-PMUs, the performance of PDCs has been challenged in recent years. First, PMUs from difference manufacturers can introduce different communication protocols. As the result, it is required that the PDC should handle multiple communication protocols as the receiving end. Second, as the reporting rates of PMUs grow increasingly higher, the PDC is expected to provide a large enough throughput for a large-volume synchrophasor data. Third, the large volume synchrophasor data should be efficiently stored so that it does not consume many resources. Finally, the PDC must provide an efficient data distribution system so that it supports online/offline power system applications. In the past several years, many have explored to address the first three challenges. Nowadays, commercial PDCs can support major PMU communication protocols [12], provide enough throughput for the large-volume data collection, and compress the data for efficient storage [13].

However, there are some issues in the modern PDCs. First, although efficient synchrophasor data compression draws much attention of the academia, the proposed compression methods may be vulnerable to large power system disturbances. For example, due to the complex characteristics of large frequency disturbance, the reconstruction performance of many state-of-the-art compression methods can be significantly deteriorated. Second, the reliability of the data storage draws much less attention of the academia. Synchrophasor data can be extremely valuable especially when it depicts the detailed dynamics of large disturbances. Such details may help engineers understand the cause of the disturbances and prevent them from future occurrence. Therefore, a data loss can be unaffordable due to its value. To reliably store the synchrophasor data should receive much more attention from the academia. Third, the challenge of data distribution system is still a vacant research topic in both academia and industry. The state-of-the-art commercial PDCs [10]-[12] are not capable of efficiently distributing the synchrophasor data. In fact, the data distribution system is an important component that can support online/offline synchrophasor applications, which provide critical functionalities to the monitoring and control of the power system. Therefore, to study the efficient distribution of synchrophasor data should receive more attention as well.

1.3 Synchrophasor applications

Synchrophasor applications utilize the field-collected synchrophasor measurements to perform tasks including disturbance detection, control, and protection. Synchrophasor applications can roughly be categorized into real-time and offline ones. Real-time synchrophasor applications are usually implemented as modules in the PDC and

they receive the real-time raw measurement data, which is usually pushed from the PDC, and process the real-time raw measurement data to generate meaningful analytical results, calculations, visualizations, etc. As opposed to the real-time ones, offline applications are usually implemented as standalone software that pull the data from the data storage system to generate post event analysis, statistical analysis, etc. Synchrophasor applications can target at solving problems at generation, transmission, and distribution levels. Nowadays, many synchrophasor applications are developed for frequency disturbances detection, voltage stability assessment, state estimation, load control, etc.

Among the synchrophasor applications, many target at assessing and ensuring the reliability of the power grids.

1.4 Motivations and objectives

Due to the outstanding challenges brought by the advances of synchrophasor technology, it is an urgent need to develop a series of efficient, reliable, and secure software systems that satisfy the ever-growing needs in the modern WAMSs and support the day-to-day operations of utilities, balancing authorities (BA), regional coordinators (RC) and electricity reliability organizations (ERO).

This dissertation develops a series of software systems that cover critical functionalities targeting at the WAMS industry including synchrophasor data collection, storage, distribution, and analytics. The dissertation utilizes the distribution-level WAMS, FNET/GridEye [3], as the basis and demonstrates multiple developments based on it. It tries to fill the vacancy in the modern PDC research and introduce some useful tools that can support the field power system operation.

Chapter II demonstrates the development of a novel PDC system, SynchroConnect. Chapter III introduced the development of a novel synchrophasor data storage system, SynchroStorage. Chapter IV discusses a novel lossy data compression to efficiently store the large volume synchrophasor data. Chapter V demonstrates the development of a micro-service, SynchroConnect, that facilitates the data exchange between different entities. Chapter VI summaries a few advanced synchrophasor applications that are developed based on the SynchroStorage and SychroConnect systems and their use cases in the power industry.

CHAPTER TWO
DEVELOPMENT OF THE SYNCHROCONNECT SYSTEM

2.1 Data Collection

The SynchroConnect system collects GPS time synchronized, real-time measurement data from micro-PMUs including frequency disturbance recorders (FDR) and universal grid analyzers (UGA). Micro-PMUs stream measurement data to the SynchroConnect system through internet connections via transport control protocol (TCP). The SynchroConnect system adopts the client-server model. Micro-PMUs act as TCP clients, who request to establish connections to the SynchroConnect server, while the SynchroConnect server accepts the requests and sends acknowledge back to micro-PMUs. When a connects is established, the micro-PMU starts to stream measurement data. In this dissertation, the UGAs use the IEEE C37.118-2011 protocol [26], and FDRs use the FNET/GridEye data protocol to format the measurement data. For generality, the SynchroConnect system is designed to receive both formats. At the server side, there are two components. A data collection module writes the newly received data into a buffer and notifies adapters to consume it. The data collection module also includes a timestamp checking function, so that the bad timestamps are filtered out before further operations. The adapters can forward or try storing the data to remote hosts. The data collection module and the adapters are designed to run asynchronously to improve the efficiency of the system. Moreover, to increase the redundancy, the SynchroConnect system is deployed in two servers to operate independently. Figure 2-1 demonstrates the architecture of the data collection module in the SynchroConnect system.

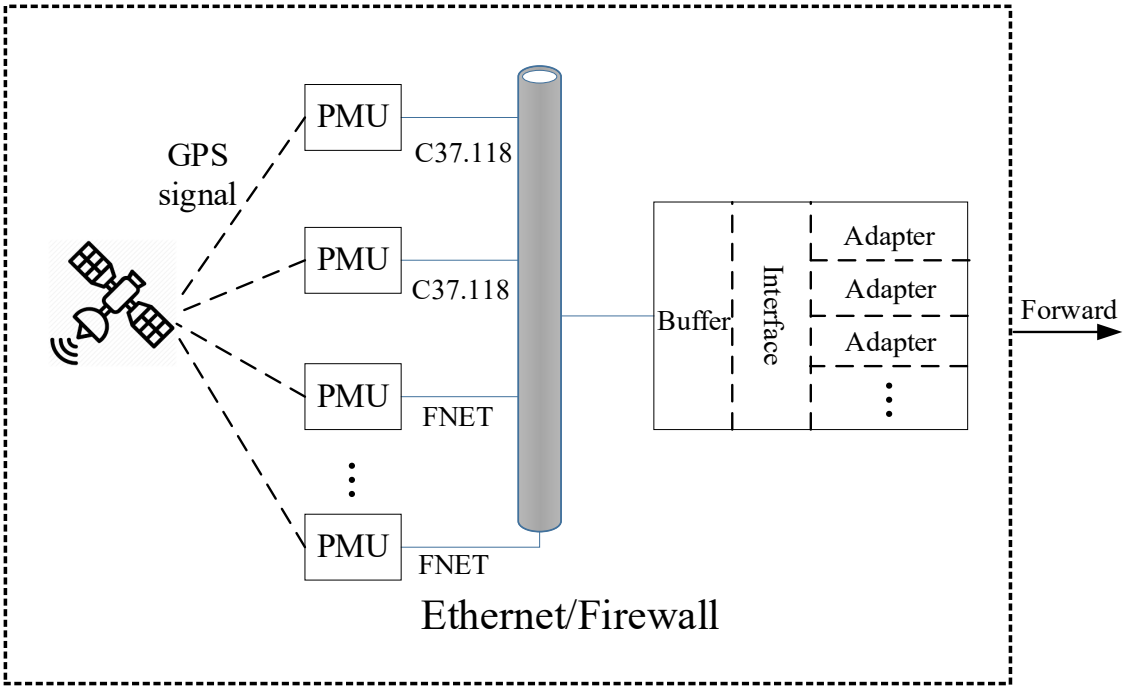


Figure 2-1 The architecture of the data collection module

2.2 Data Forwarding

The SynchroConnect system can also forward collected data to other systems. The data forwarding function is designed to support the real-time synchrophasor applications as well as the data storage in a distributed manner. By forwarding the measurement data to dedicated servers, the collection, analytics, and storage of data are decoupled. This strategy improves the overall reliability of the system so that the failure of a single machine does not affect the others. The data forwarding function is typically implemented by an adapter and the data forwarding module also adopts the TCP client-server model. To forward data, the TCP adapter requests to establish a TCP connection to a remote server as a client, then streams the collected data through the established connection via user-define protocols.

2.3 Data Storage

The SynchroConnect system may store collected data into various storage mediums. Upon receiving the data stream, the SynchroConnect system verifies the validness of the data then begins the storage procedure. The SynchroConnect system may store the data as local formatted files or send it to other time-series databases that are deployed either locally or remotely. In this dissertation, a master-slave model is adopted to store the measurement data in a distributed manner. Specifically, it defines two types of nodes, master node and slave node as shown in Figure 2-2. A master node keeps a list of canonical PMU configurations as a reference but does not store the measurement data. A slave node keeps a local PMU configuration and stores the associated measurement data.

Whenever a change of configuration happens, it will be populated among the slave nodes to ensure the consistency of configurations. The main reason of adopting the master-slave structure is its easiness of data management. By decoupling the configuration and the measurement data, manual changes are only allowed in the master node, which eliminate the potential human errors on updating all nodes. Furthermore, this structure may allow a more efficient redundancy plan, where the redundancy is achieved by assigning only part of the data to a slave node and properly overlapping the slave nodes by redundant data [7]. In this way, the slave nodes can be implemented in a partially redundant mode to save storage space.

Furthermore, the master-slave structure allows heterogenous storage mediums. Due to the upgrade of legacy systems, there can be some old storage mediums that are still be used along with the newer ones. By managing the configurations at the master node, the storage subsystem knows and automatically updates the configurations of the slave nodes. This makes it possible to easily manage the redundancy plan and the configurations for the slave nodes without accessing the configuration of each slave node. Due to its efficiency in data storage and query, this paper uses open source time-series databases to store synchrophasor data on slave node [12], [14]. Table 2-1 shows the time-series databases that are supported by the SynchroConnect system.

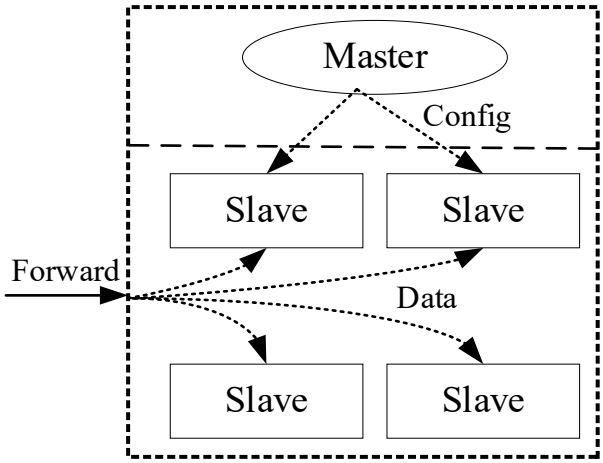


Figure 2-2 The architecture of the data storage module

Table 2-1 Supported time-series databases

Name	Master node	Slave node	
	Configuration	Configuration	Measurements
OSIsoft PI system	Yes	No	Yes
OpenHistorian	Yes	Yes	Yes
InfluxDB	Yes	No	Yes

2.4 Data redundancy

Data redundancy improves the reliability of data storage, offering immunity to occasional machine failures that may lead to data loss. Generally, data can be fully or partially redundant. A fully redundant strategy requires each node host an identical piece of data. The fully redundant strategy offers the best storage reliability. Given the total number of nodes n, each chunk piece of data gets replicated n times, which makes the probability of a data loss extremely low. Nevertheless, this strategy may greatly increase the cost since each disk keeps an identical copy of data, which might be unnecessary and costly in practice. As opposed to it, the partially redundant strategy only requires a piece of data be replicated for x times, where x is much less than n. This strategy can offer similarly low data loss probability, but it is of a much lower cost. The partially redundant strategy is widely adopted in cloud storage technology.

In the SynchroConnect system, both strategies are adopted to satisfy different scenarios. The fully redundant strategy is implemented in the master nodes. This is because the data size of the configuration is small, and it is usually stored in structural query language (SQL) databases, where the fully redundant strategy is easier to implement. On the other hand, the partially redundant strategy is implemented in the slave nodes, where each piece of data is replicated by two slave nodes at a time. This is due to the large size of the synchrophasor data and its nature for easy segmentation. The choice of two replications might seem to be a suboptimal solution to prevent a data loss since three replications can increase the mean time between data loss (MTDL) from 10 to 100 years. However, the main purpose of the data storage system in the SynchroConnect is to host

several years of data. Furthermore, synchrophasor measurements that are older than several years may not be of a great usage unless it contains significant disturbances. Considering these practical reasons, the two-replication strategy is chosen to ensure an acceptable data reliability yet reduces the costs of hardware.

2.5 Node synchronization

Node synchronization is a necessity in a distributed data storage system. Due to its distributed storage structure, the SynchroConnect system also requires a well-designed synchronization system to ensure the consistency of slave nodes. In the SynchroConnect system. the majority of the synchronizations happens in the PMU configuration. The PMU configuration specifies both runtime information such as name, location, reporting rate, types of measurements, TCP connection string, etc. and hardware information including the digital signal processing (DSP) module, internet module, etc. To reduce the network traffic, each slave node maintains and uses a local copy of the configuration. It is worth to note that, configurations between two slave nodes can be different. This is because two slave nodes can be configured to store different part of the data. In the SynchroConnect system, the configuration in the master node is always the latest, while that of a slave node can be nearly latest. A slave node only updates its configuration file by accepting and executing the synchronization command from the master node after a series of updates is posted. The synchronization command can be sent manually or automatically. The manual synchronization command is sent when the PMU administrator finishes updating the configurations and. As opposed to it, the SynchroConnect system also maintains a time interval τ for periodical automatic synchronization. Whenever a τ-length of time elapses, a

synchronization command is sent to the slave nodes to allow them to update their configurations.

To update the slave nodes, the SynchroConnect system first stores the changes, on the master node, that are made by the PMU administrators into an operation cache. When the update on the master node is done, the system analyzes the operation cache and compares the changes with the configurations of the slave nodes, then enqueues necessary updates for all slave nodes. Whenever a manual synchronization command is sent or τ elapses, the master node dequeues the updates and populates them to the slave nodes sequentially. A typical workflow of the synchronization process is shown in Figure 2-3.

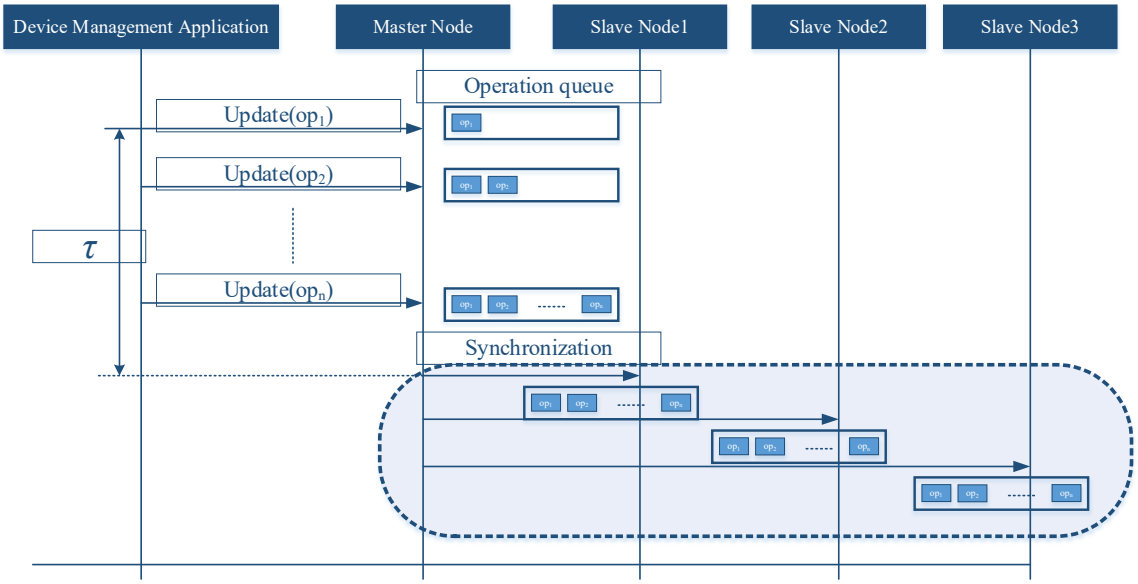


Figure 2-3 Node synchronization workflow

CHAPTER THREE
SYNCHROPHASOR LOSSY COMPRESSION FOR EFFICIENT DATA STORAGE

3.1 Introduction

PMUs have been increasingly deployed in the past decades since their invention, which overperform traditional supervisory control and data acquisition system (SCADA) thanks to their high reporting rates, rich measurement types, and high accuracy. These features enable many advanced applications that help ensure the reliability of power grids **[15]-[18]**. Typically, a PMU collects phasors including voltage magnitude, voltage angle, current magnitude, current angle, frequency, etc. A typical PMU collects GPS-synchronized phasors and streams them to a phasor data concentrator (PDC) at 10-120 Hz reporting rate **[18]**, **[19]**. On the other hand, grid structures nowadays are complex **[20]-[24]**, which require more and more PMUs to cover the transmission system. For example, according to **[25]**, to cover the transmission system in the USA, more than 1100 PMUs are required. Clearly, the high reporting rate and the large number of PMUs will result in a huge amount of data. For example, assuming there are 1100 PMUs reporting data at 30 Hz via the IEEE C37.118 protocol **[26]**, over 700 gigabytes (GB) data will be generated per day. Furthermore, using advanced PMUs, which report at 120 Hz, the total data volume can exceed 2.8 terabytes (TB) per day. Realizing the challenge from the large-volume PMU data, data compression techniques need to be exploited to efficiently compress **[27]** and store the data.

In general, compression techniques can be categorized into lossy and lossless approaches **[28]**. Lossless compression focuses on exploring the statistics of the data and using efficient bit-wise encoding techniques to compress it. Lossless compression allows compressed data to be compressed with no information loss. Comprehensive comparisons are conducted among well-known lossless compression models including Deflate, Bzip2, Lempel-Ziv 77 (LZ77), Lempel-Ziv-Markov-algorithm (LZMA), and the Szip **[29]**, **[30]**. These works imply using the Szip model may achieve the best compression performance for synchrophasors. However, lossless compression methods can hardly reach a high compression ratio (CR) since the dimension of the synchrophasor data is ignored.

On the contrary, lossy compression emphasizes trading controllable errors for a better CR. For synchrophasor data compression, the lossy compression models mainly rely on two philosophies. The first and most straightforward way is to compress the data by analyzing each PMU independently. Models such as discrete wavelet transformation (DWT) **[31]**, **[32]**, improved DWT **[33]-[35]**, exception compression (EC)-swing door trending (SDT) **[36]**, etc. are proposed. Among these models, the SDT method can achieve good CR with small normalized mean square error (NMSE). Another way to compress the data exploits the linearity in the synchrophasor data. Within an interconnected power grid, synchrophasors may contain high linearity. For example, a principal component analysis (PCA) based model is proposed to use dimensionality reduction to perform early event detection **[37]**. This work lays the basis of using dimensionality reduction approaches to analyze PMUs' data in the modern smart grid and it implies its potential to compress the PMU data as well. Towards this end, an SCD-PCA-DWT/DCT model is proposed to

compress the synchrophasors **[38]**. In this work, the PCA compression achieves good performance on various measurements. Similarly, another PCA-based algorithm is proposed to compress the distribution-level synchrophasors **[39]**. In this work, the data can be compressed at good CRs with controllable errors. Recently, a multiscale PCA model is proposed to decide different parameters for the PCA model. It first performs a spatial-wise cluster analysis, then uses different PCA models to compress corresponding clusters. This work can achieve good CR under ambient conditions and acceptable CR under generator trip conditions. However, this model was not extensively tested with disturbance events such as oscillations, and inter-area oscillations, where the clusters can have a similar density.

3.2 Issues of lossy compression methods

Though it seems promising, there are several issues in using PCA to compress the synchrophasor data. The first one is the choice of the principal components or the compression space R. Some have proposed to separate the data into ambient condition and fault conditions and decide the compression space R respectively. However, the choice of principal components is still not well defined. In **[38]**, 80% static normalized cumulative variance (NCV) is used under ambient conditions, while 95% NCV is used under fault conditions. A similar score selection philosophy is implemented in another work **[39]**. This rather static rule has a chance of losing important information during data compression, even if the RMSE seems acceptable.

As shown in Figure 3-1, when the data is compressed and reconstructed under 99.0% score, there is still a significant information loss. For the reconstructed data, the

maximum frequency is reduced from above 60.25 Hz to 60.08 Hz, while the minimum frequency is elevated from 59.80 Hz to 59.88 Hz. This error can result in inaccurate frequency response assessments, which are required in standard BAL-003 **[40]** by North American Electric Reliability Corporation (NERC). Another issue is the effects of disturbances. A related work proposes to use statistical change detection (SCD), where the deviations from the measuring values to the nominal values are quantified given a time window **[38]**. In this work, the "nominal values" are calculated by averaging the measured values of the current device in the time window. However, from the wide-area standpoint, using the average measurements of a single device may lead to several drawbacks. First, under islanding conditions, the measured quantities of some locations can go way off from other locations due to desynchronization. Using the SCD method, the islanded devices may still report itself as running under nominal conditions if their data does not contain large excursions. This will affect the calculation of the SCD because if all devices report themselves as running under nominal conditions, no statistical change will be detected. Second, using the average value of the past several seconds may be insufficient to measure the chaos in the data since it can be sensitive to normal frequency changes under low load conditions. In such a scenario, the synchrophasor measurements can contain high linearity even if the SCD algorithm reports a statistical change.

Figure 3-2 shows the frequency, the SCD, and the wide-area deviation (WAD) of a typical frequency ramping. Here, the WAD is represented by the difference between the

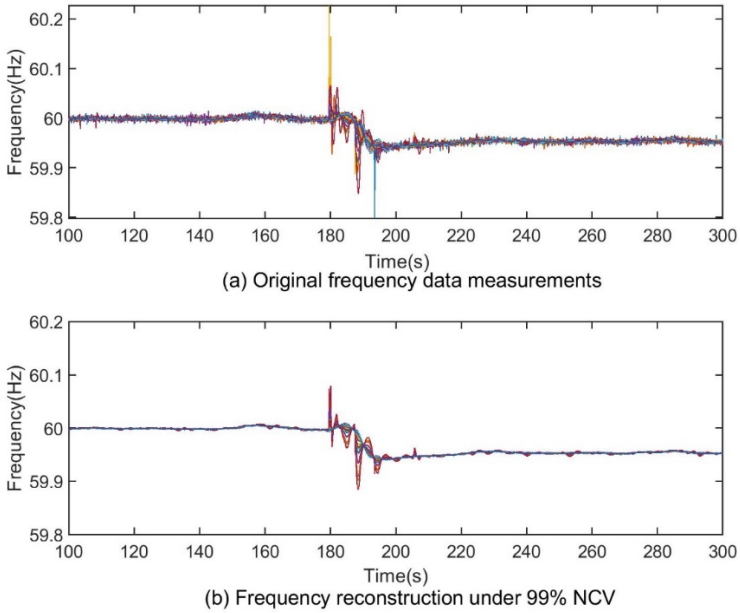


Figure 3-1 Frequency data comparison (110 units)

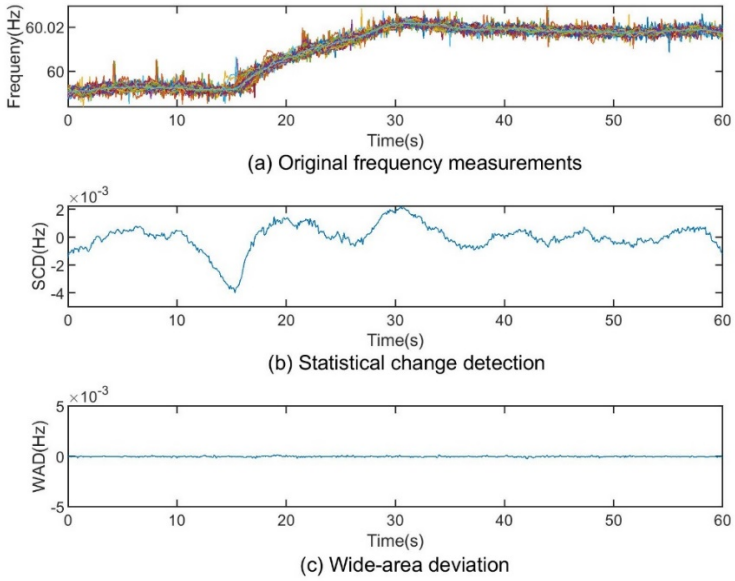


Figure 3-2 Frequency ramping under low load condition

measurements and the system medians. As seen in Figure 3-2, a scheduled load change causes the frequency to ramp up to 60.024 Hz, while the frequencies are still synchronized across the grid. High linearity is observed in the data as the WAD is very smooth. However, the SCD reports a statistical change, which cannot accurately reflect the real system dynamics.

This paper tries to address the abovementioned issues by evaluating and compressing the synchrophasor data via cross-entropy and the state-of-the-art PCA variant, singular value decomposition (SVD) **[39]**. First, this paper exploits a machine learning concept, cross-entropy, to evaluate the patterns within the synchrophasor data. Then, it generates compression periods according to the evaluation result. Finally, the proposed model compresses the synchrophasor data using the SVD algorithm, under relative error thresholds.

3.3 Proposed lossy compression method

Cross-Entropy for Synchrophasor Data

Cross-entropy **[41]** is a widely used concept in machine learning. It is commonly used in machine learning loss functions to measure the difference between the model outputs and the ground truth. For synchrophasors, the cross-entropy can be written as

$$H\left(M^t, \overline{M^t}\right) = -\sum_i P\left(\overline{M^t}\right) log\, P\left(M_i^t\right) \tag{1}$$

where i and t represent the device ID and the time index, respectively. M is the distribution of the measurements, $\overline{M}$ is the distribution of the nominal value, $H\left(M^t, \overline{M^t}\right)$ represents the cross-entropy of M^t with respect to $\overline{M^t}$, $P(x)$ is the probability of sample x.

Cross-entropy can be used as an ancillary criterion for data compression due to its indication of off-nominal patterns. Under disturbance scenarios, the synchrophasor signals can have obvious off-nominal patterns. The off-nominal patterns represent the extent to which the system runs "chaotically". The introduction of cross-entropy helps describe how different the synchrophasor data is from the nominal patterns. Since the off-nominal patterns are observed during fault conditions and the nominal patterns are observed during ambient conditions, the evaluation of the synchrophasor data can be generalized as a bipartite classification problem. A simplified cross-entropy function for a bipartite classification can be written asw

$$H_B(M_i^t) = -\left\{\overline{M_i^t} log[P(M_i^t)] + \left(1 - \overline{M_i^t}\right) log\left(1 - P(M_i^t)\right)\right\} \tag{2}$$

where, $H_B(M_i^t)$ represents the bipartite cross-entropy of M_i^t.

Now, the target is to identify the chunk of data that has off-nominal patterns, evaluate its cross-entropy, and separate it from others that have nominal patterns. Therefore, (2) can be further simplified as (3), since presumably only the logarithmic distance between the distribution M_i^t and the target 0 is concerned.

$$H_B(M_i^t) = -log[P(M_i^t)] \tag{3}$$

In the compression algorithm, the nominal value of the frequency data is defined as the median frequency of an interconnected grid and the distribution M is calculated by subtracting the system median frequency value using PMUs' reported actual frequencies. The nominal value of the voltage magnitude is the normal voltage magnitude per unit (pu), and its distribution M is calculated by subtracting each PMU's normal voltage magnitude (pu) using this PMU's actual voltage magnitude. Finally, the nominal value of the phase

angle is the median of unwrapped angles. It is worth noting that phase angles may vary greatly compared to the frequency and voltage magnitudes. Therefore, for a certain PMU, it is required to subtract its phase angle value at the first timestamp from all rest phase angle values **[42]**, then normalize it through a [0,1] range.

Algorithm 1: Calculate the cross-entropy of the synchrophasor data, and generate partitions according to the cross-entropy levels

Input: $Sdata^l$: the synchrophasor data, where l is the length of the data; $Ndata^l$: the nominal-value data; *Threshold_entropy*: the entropy threshold separating the ambient and the disturbance conditions; *ew_size*: the window size to pre-partition the data; *mw_size*: the window size to merge the pre-partition results.

Output: *Partitions*: the partitions generated by the algorithm

Initialization: $c \leftarrow 0$, $j \leftarrow 0$, *Partitions* ← [], $k \leftarrow 1$, *Merged_patitions* ← [].

```
while j < l do
   entropy ← Sdata_j - Ndata_j
   if entropy > Threshold_entropy then
      if c = 0 then
         s ← j
      end if
    c ← ew_size
   else
       if c > 0 then
          c ← c -1
          if c = 0 then
             e ← j
             Append [s, e] to Partitions
          end if
      end if
while k < l do
   if Partitions[k][0]-Patitions[k-1][1]<mw_size then
      Append [Partitions[k-1][0], Partitions[k][1]] to
Merged_partitions
   end if
k ← k - 1
end while
```

Cross-Entropy-Based Evaluation

The purpose of the cross-entropy analysis of the synchrophasor data is to identify the periods that are chaotic, i.e. they contain low linearity. Identifying these periods are crucial to the data compression because the dimensionality reduction-based compression models exploit the high linearity of the synchrophasor data to achieve optimal compression performance. Therefore, if a chunk of data is of high cross-entropy, a lower CR is required to maintain the information in the data, otherwise, a higher CR may be used to achieve a superior compression ratio without losing too much information.

This paper proposes a cross-entropy based synchrophasor measurement evaluation approach combining the information from a wide area. Algorithm 1 shows a general partitioning method that calculates the cross-entropy and generates the partitions for a chunk of synchrophasor data. However, on some occasions, it will generate partitions that are temporally close to each other. This is because high-linearity and low-linearity periods are interweaved under fault conditions. Although the number of partitions may not affect the compression performance directly, more partitions can result in excessive overheads that may take considerably large space when the data chunk is small. To avoid excessive overheads, the temporally close partitions are merged to reduce the number of partitions.

Error! Reference source not found. shows the entropy distribution of a chunk of frequency measurements. As is seen from the figure, the entropy of the ambient periods is around 10^{-7}, which is relatively trivial compared to that of a fault period. Meanwhile, the entropy of the fault period goes up to over 10^{-3}, which is 104 times larger than that of an ambient period. With entropy being calculated, generating partitions becomes a rather

straightforward task. Figure 3-3 also shows the merged result from step 2. For the frequency data, the nominal value at timestamp always equals to the median frequency of the grid. Then this paper uses the proposed algorithms to partition the frequency data into chunks. As is seen from the figure, the frequency data is partitioned into 3 chunks. The first and the last partitions reflect the system-wide frequency distribution under ambient conditions, while the second partition reflects that under a generator trip condition. The first and the last partitions also imply high linearity, as their data are more "concentrated". In the meantime, the second partition shows low linearity, as its data contains more excursions.

Synchrophasor Data Compression via Singular Value Decomposition Considering Disturbances

Using dimensionality reduction models to compress the synchrophasor data is not a new area of research. However, as aforementioned, the optimal decision of the compression space R is more of a human experience-based trade-off between compression performance and accuracy.

A static choice of R is relatively biased in terms of the type, the volume, the resolution, and the entropy of the synchrophasor data. The choice of R for a small synchrophasor network may not work for a large synchrophasor network assuming higher linearity exists when the number of devices is larger. On the other hand, under disturbance-involved power system dynamics, the linearity of the synchrophasor data can change drastically **[42]**, which makes the choice of R rather difficult. Information vanishing is likely to happen if improper R is chosen.

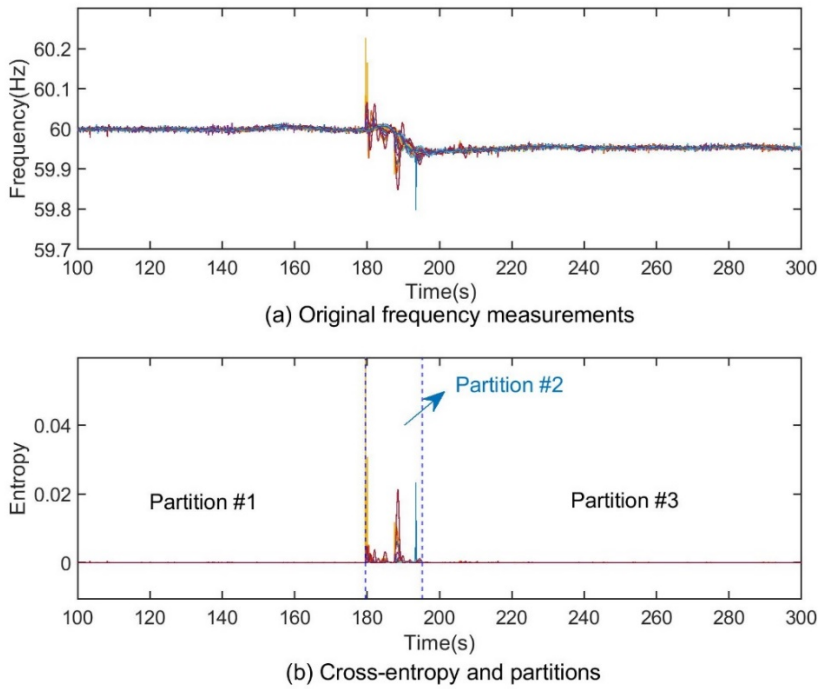


Figure 3-3 Entropy distribution of frequency measurement during a generator trip

To address this issue, this paper exploits the local characteristics of the synchrophasor data, proposing a dynamic singular value decomposition model to decide the best R for each data chunk. On the other hand, the proposed model also uses a relative evaluation methodology, which is capable of tracing very small fluctuations in the data.

Singular Value Decomposition

SVD is a widely accepted dimensionality reduction algorithm, which decomposes a large matrix $\boldsymbol{M_{m\times n}}$ into three smaller matrix $\boldsymbol{U}_{m\times n}$, $\boldsymbol{\Sigma}_{n\times n}$, and $\boldsymbol{V}_{n\times n}$. The SVD algorithm can be represented as

$$\boldsymbol{M}_{m\times n} = \boldsymbol{U}_{m\times n}\boldsymbol{\Sigma}_{n\times n}\boldsymbol{V}^T_{n\times n} \tag{4}$$

where m is the number of samples, n is the number of PMUs, $\boldsymbol{U_{m\times n}}$ is the left singular vectors, $\boldsymbol{\Sigma_{n\times n}}$ is the diagonal matrix that represents the singular values, and $\boldsymbol{V_{n\times n}}$ is the right singular vectors.

The compression algorithm takes the top K singular vectors out of the N singular vectors. Therefore, the SVD reduces the problem to

$$\boldsymbol{M}'_{m\times n} = \boldsymbol{U}_{m\times k}\boldsymbol{\Sigma}_{k\times k}\boldsymbol{V}^T_{k\times n} \tag{5}$$

The compression ratio is calculated by measuring the total number of values of the original matrix and the reduced matrix. Therefore, the compression ratio (CR) is calculated by

$$\boldsymbol{CR} = \frac{m\times n}{m\times k+k\times k+n\times k} = \frac{m\times n}{k(m+k+n)} \tag{6}$$

Model Tuning via Local Characteristics

In this paper, a local characteristic evaluation methodology is proposed to address the vanishing of information caused by using a static NCV score as the threshold. The local characteristic LC is represented by

$$LC^{s,e} = \max_{i=1,\ldots,n} \left| M_i^{s,e} - median(\boldsymbol{M}^{s,e}) \right| \tag{7}$$

where $LC^{s,e}$ is the local characteristic of the ith measurement within a period, s is the start time, e is the end time, $M_i^{s,e}$ is the ith measurement data within the [s,e] time period.

The idea of introducing the system median is it represents the most common distribution of the data. By calculating the maximum absolute deviation between the measurement and the system median, the proposed method recognizes the largest excursions that are caused by the disturbances.

This paper calculates a proportion of the LC as the criteria to decide the tolerated reconstruction error threshold (TRET) by

$$TRET^{s,e} = \lambda \cdot LC^{s,e} \tag{8}$$

where λ is a static coefficient that represents the TRET in percentage. This paper uses 0.05 as the value of λ throughout the performance evaluation. It is noted that the choice of λ is subject to the requirements of users. Users may choose a smaller λ to preserve more information or a larger λ to get bigger CR per the requirements of applications.

3.4 Performance evaluation

Simulated Data

This paper uses the "savnw" 23-bus system provided by PSS®E 33 **[43]**, assuming each bus is equipped with a PMU, which measures the bus voltage (VM), voltage phase angle (VH), and frequency (F) at a reporting rate of 120 Hz. In this paper, all simulations last 60 seconds. Since synchrophasor data may subject to local distribution or transmission characteristics, it is common to observe noises in such data **[44]**. To make the simulated data more authentic, white Gaussian noises of 75dB, which equals to the observed average noise level in the field-collected data **[45]** as well as random phase and frequency jumps **[46]** are added to the simulated synchrophasor dataset.

In the simulation, this paper considers disturbances including bus fault (BF), line fault (LF), transformer switch off (TF), and line trip (LT).

In this paper, 2 criteria are considered to evaluate the reconstruction performance, which are maximum absolute error (MAE) and average root mean square error (ARMSE). The MAE is calculated by

$$MAE = \max_{m,n} |\boldsymbol{M}_{m\times n} - \boldsymbol{M}'_{m\times n}|, \tag{8}$$

while the ARMSE is calculated by

$$ARMSE = \frac{\|\boldsymbol{M}_{m\times n} - \boldsymbol{M}'_{m\times n}\|}{\sqrt{mn}} \tag{9}$$

Table 3-1 shows the comparison of the proposed cross-entropy-based SVD (SVD-CE) approach and the state-of-the-art statistical change detection-based SVD (SVD-SCD) algorithm. As is seen from the table, the proposed SVD-CE generally outperforms the

SVD-SCD algorithm. For the comparison of the voltage magnitude data under BF, the performance of the SVD-CE algorithm has better CR while keeping lower recovery errors. This is because, under BF, steps changes happen after the short-circuits on the buses, which cannot be easily detected by the SVD-SCD algorithm. It is also seen from Table 3-1 that the SVD-SCD can achieve better CRs on some occasions, but they usually imply unaffordable information loss. This is because the SVD-SCD algorithm may generate a single data chunk, where the high-linearity and low-linearity periods are interweaved. During events where high-linearity periods outnumbers low-linearity ones, the overall linearity of the data chunk may rise. Under these scenarios, the SVD-SCD algorithm can compress the data more aggressively. Moreover, this feature can also result in lower ARMSEs among the high-linearity periods since the algorithm tends to fit the high-linearity periods but undermine the low-linearity periods. These observations mostly happen under the LF and the LT conditions because under both conditions the high-linearity periods well outnumber the low-linearity periods. As opposed to it, the proposed SVD-CE algorithm differentiates the high-linearity periods and the low-linearity periods using their cross entropies. It assigns the best CR to each period, while keeps superior reconstruction accuracy. As is shown in Table I, the proposed algorithm can restrain the MAEs within lower ranges, while maintaining comparable CRs.

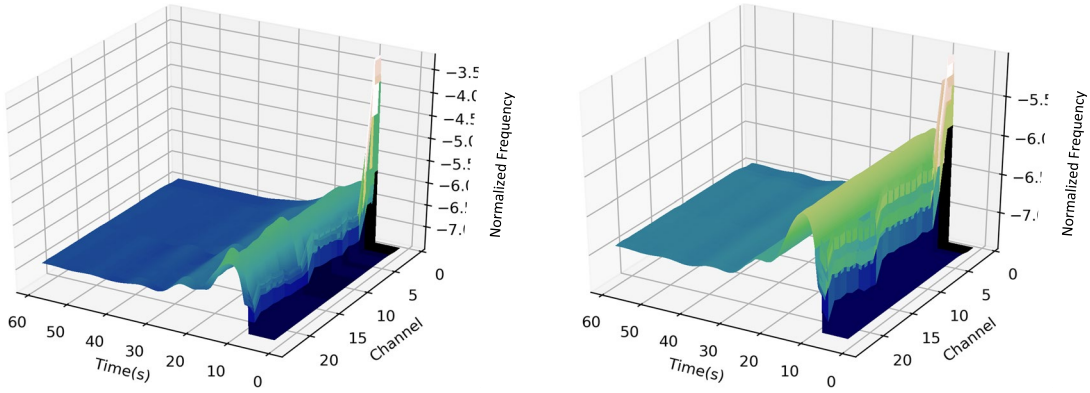
As a case study, Figure 3-5 shows the recovered angle data under a BF. The proposed SVD-CE algorithm reconstructs the data more accurately, while the SVD-SCD algorithm has human-eye perceivable errors at many time instants. This is because the SVD-SCD algorithm tends to find fewer principal components and causes the loss of

critical information. In this case, the average CR of the SVD-CE algorithm is 2.7, while that of the SVD-SCD algorithm is 3.8. Therefore, the SVD-CE algorithm reaches a comparable CR rate while keeps relatively lower errors. Figure 3-4 shows the reconstruction of frequency data under a complex disturbance, where an LT follows an LF. As seen in Figure 3-4 (a), there are a few frequency spikes in the original data around the fault location. After the disturbances, the frequency first rises, then drops to a steady level as frequency responses take place. As seen in Figure 3-4 (b), the proposed SVD-CE approach can reconstruct the data to reflect the dynamics under the disturbance. However, as seen in Figure 3-4 (c), the SVD_SCD algorithm over-generalizes the data. It only includes the trend of the frequency while loses critical information around the fault location.

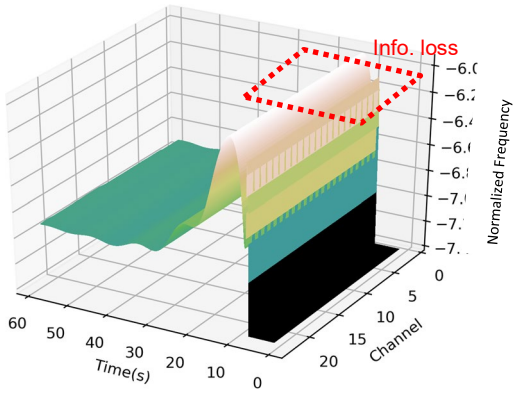
In conclusion, for the simulated data, the proposed model can maintain critical disturbances information while achieving a comparable CR rate. Nonetheless, as seen from Table 3-1, the CR improvement by the proposed SVD-CE algorithm is usually not obvious. This is because the simulated data contains less noise. The cleaner simulated data is mainly caused by the simplicity of the simulation system. In this simulation system, there are few renewable sources or power electronic interfaces, making the noises caused by harmonic pollution, etc. less obvious than those in a real, complex system. Therefore, there has not seen a significant improvement in the CRs on all signals.

Table 3-1 Performance Comparison of Simulated Data

	Sig.	BF			LF			TF			LT		
		CR	MAE	ARMSE	CR	MAE	ARMSE	CR	MAE	ARMSE	CR	MAE	ARMSE
SVD-SCD	VM	5.8	1.2×10^{-16}	7.4×10^{-20}	5.7	3.2×10^{-3}	1.7×10^{-5}	4.6	2.2×10^{-4}	3.5×10^{-6}	5.7	7.4×10^{-4}	1.4×10^{-5}
	VH	3.8	4.6×10^{-5}	5.4×10^{-6}	11.5	5.1×10^{-5}	3.6×10^{-7}	11.5	1.6×10^{-5}	1.1×10^{-6}	11.5	5.7×10^{-6}	5.5×10^{-7}
	F	1.05	1.7×10^{-4}	2.5×10^{-5}	7.6	2.0×10^{-4}	3.4×10^{-6}	11.5	9.6×10^{-5}	3.7×10^{-7}	11.5	1.5×10^{-4}	3.0×10^{-6}
SVD-CE	VM	**9.5**	$\mathbf{3.5\times10^{-17}}$	$\mathbf{2.5\times10^{-20}}$	4.7	$\mathbf{7.7\times10^{-4}}$	4.8×10^{-5}	4.7	$\mathbf{1.2\times10^{-4}}$	2.9×10^{-6}	4.7	$\mathbf{2.0\times10^{-4}}$	3.4×10^{-6}
	VH	2.7	4.3×10^{-5}	1.37×10^{-7}	11.1	$\mathbf{5.8\times10^{-6}}$	2.0×10^{-7}	9.7	$\mathbf{8.3\times10^{-6}}$	2.2×10^{-7}	11.2	5.2×10^{-6}	3.1×10^{-7}
	F	**1.0**	$\mathbf{2.7\times10^{-18}}$	$\mathbf{4.0\times10^{-18}}$	7.7	$\mathbf{1.0\times10^{-4}}$	3.3×10^{-6}	11.4	$\mathbf{2.3\times10^{-5}}$	4.9×10^{-7}	7.8	$\mathbf{4.1\times10^{-5}}$	2.4×10^{-6}



(a) Original frequency measurements (b) Reconstruction via SVD-CE



(c) Reconstruction via SVD-SCD

Figure 3-4 Frequency reconstruction performance LF & LT (23 units)

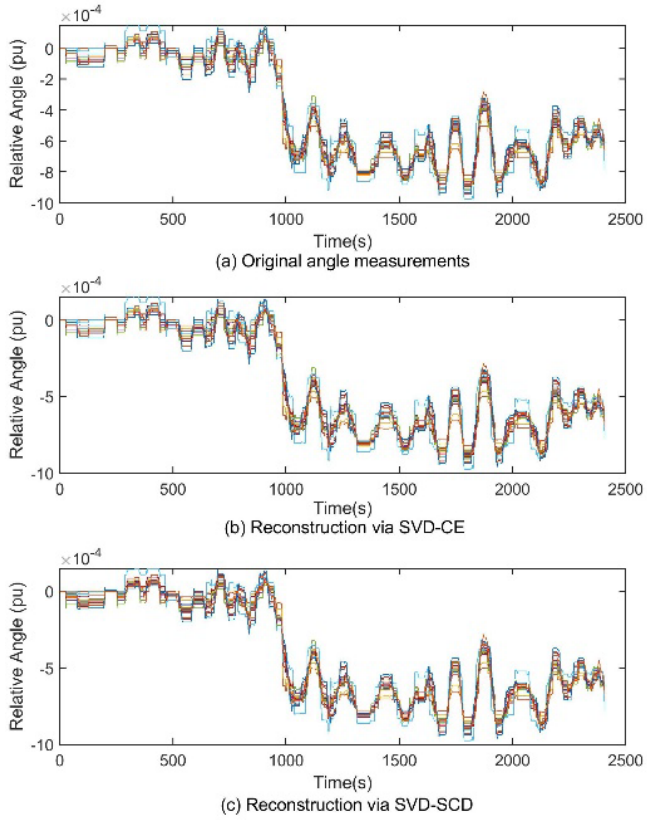


Figure 3-5 Reconstruction performance angle BF (23 units)

Field Data

In this subsection, the field-collected frequency and voltage phase angle from the U.S. eastern interconnection provided by the distribution-level wide-area monitoring system (WAMS) FNET/GridEye are used for performance evaluation **[47]**. For the field data, this paper evaluates the dynamic conditions including generator trip (GT), frequency ramping (FR), oscillation (OC), and forced oscillation (FO).

Table 3-2 shows the performance comparison of the filed collected data. As is shown from table, under real-world scenarios, the proposed SVD-CE algorithm generally outperforms the SVD-SCD algorithm. Under GT and FO scenarios, although the CRs of the proposed algorithm is similar to the SVD-SCD algorithm, their errors are much less than the SVD-SCD algorithm because the SVD-CE algorithm has a stronger ability to pinpoint the high-entropy periods in the real-world scenarios, which have lower linearity for synchrophasor data in different locations. By specifying these periods, the algorithm can fit these periods with a tailor-made error threshold instead of an NCV.

As is seen from Table 3-2, under LT scenarios, the proposed algorithm can achieve a much better result than the SVD-SCD algorithm. The reason is that static NCV does not work well when LT events present. The NCV tends to force the algorithm to find a much higher compression dimension to meet the NCV. However, the proposed algorithm finds the lowest possible compression dimension while keeping relatively low reconstruction errors. Moreover, under LT scenarios, there are tiny phase steps in the point of wave (POW) measurements that cause frequency deviations [48].

Table 3-2 Performance Comparison of Field Data

	Sig.	GT			LT			FR			FO		
		CR	MAE	ARMSE	CR	MAE	ARMSE	CR	MAE	ARMSE	CR	MAE	ARMSE
SVD-SCD	VH	49.6	1.2×10^{-4}	1.7×10^{-6}	17.1	3.7×10^{-5}	1.8×10^{-6}	**38.8**	**1.7×10^{-4}**	**3.4×10^{-6}**	40.2	6.8×10^{-6}	2.9×10^{-6}
	F	9.3	1.3×10^{-3}	3.6×10^{-5}	11.9	2.2×10^{-4}	2.1×10^{-5}	92.0	8.3×10^{-5}	2.4×10^{-8}	1.7	2.9×10^{-4}	1.8×10^{-5}
SVD-CE	VH	49.4	3.2×10^{-5}	1.5×10^{-6}	19.0	3.6×10^{-5}	1.7×10^{-6}	**41.9**	**1.7×10^{-5}**	**3.7×10^{-6}**	52.1	6.8×10^{-4}	4.4×10^{-6}
	F	9.5	4.1×10^{-4}	2.6×10^{-5}	15.5	2.2×10^{-4}	1.7×10^{-5}	92.0	8.3×10^{-5}	2.4×10^{-8}	1.6	2.1×10^{-4}	8.0×10^{-6}

Under LT scenarios, the tripping of the line usually causes tiny phase steps in the POW, which in result causes tiny local angle variations. As is shown in Figure 3-6, the LT event causes angle variations at around 14.8s. The magnitudes of these variations are very small, making them hard to recognize via human eyes. However, these tiny variations represent the intrinsic characteristics of the POW data, which is of great significance in terms of deciding the credibility of the LT event. As is seen, the SVD-CE algorithm successfully preserves these tiny variations that the SVD-SCD algorithm does not reflect. On the other hand, it also suggests, under LT scenarios, the SVD-CE algorithm achieves better CR while keeping good reconstruction accuracy.

Moreover, as Figure 3-7 shows, under FO scenarios, the proposed SVD-CE algorithm successfully retains the oscillation information, while the SVD-SCD algorithm only retrains partial oscillation information. Moreover, under FO scenarios, the information loss of the SVD-SCD algorithm can greatly affect the event analysis, since it loses critical sinusoidal signals at multiple points. As a result, the reconstructed data of the SVD-SCD algorithm would fail the classic forced oscillation analysis. Therefore, even though the SVD-SCD algorithm achieves better CR, it fails to retain critical information, which is crucial to event analysis.

Data compression under complex disturbance conditions

In field operations, the characteristics of disturbances are complex. Complex disturbances can greatly affect the dynamics of an interconnected power grid; thus, they can seriously deteriorate the performance of the compression algorithms.

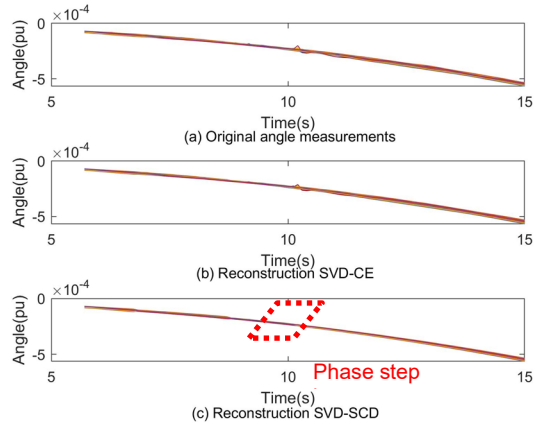


Figure 3-6 Reconstruction performance angle LT (110 units)

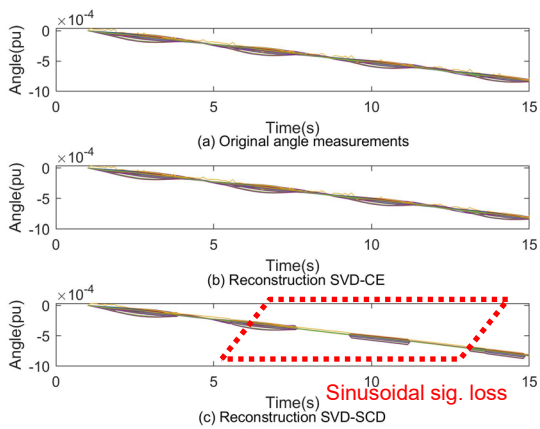


Figure 3-7 Reconstruction performance angle FO (110 units)

A notable feature of the proposed method is its superior information retaining ability under complex disturbances. Figure 3-8 shows the performance comparison during an GT event. In Figure 3-8 (a)-(b), the x-axis represents the PMU channels, while the y-axis represents the time index. A GT disturbance is observed at time index 192 and it causes system-wide frequency drops in all channels. As seen, the proposed SVD-CE method achieves good reconstruction performance. This is because the it can successfully identify the disturbances period and perform efficient compression strategies on disturbance period and ambient periods respectively.

For the SVD-SCD method, it loses critical information during the disturbance and post-disturbance periods. The high reconstruction error during the disturbance period is caused by its inability to bound the TRET. While the high reconstruction error during post-disturbance period is since the SVD-SCD approach treats the post-disturbance period as ambient periods. Therefore, it tends to over-simplify the post-disturbance characteristics by finding a much smaller compression space regardless of the important local characteristics. As opposed to it, the proposed SVD-CE algorithm still retains more information. A similar example is shown in Figure 3-9, where an LD disturbance is involved. As seen from Figure 3-9 (c), the SVD-SCD approach finds less accurate representations of the data. It still over-simplifies the data and causes inaccuracies in the post-disturbance period. On the other hand, on some channels, this approach results in erroneous reconstructions including spikes that are not presented in the original data.

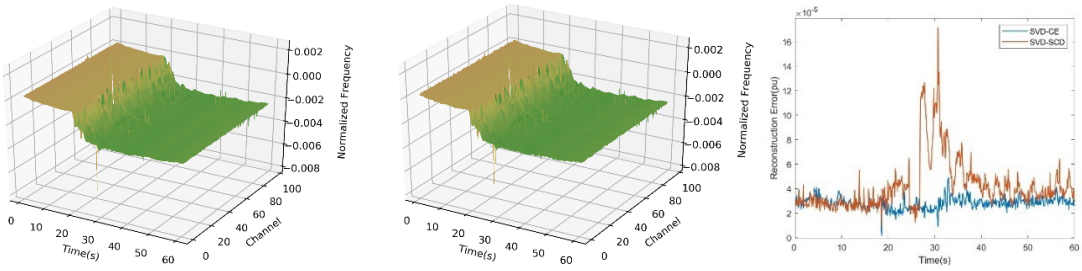
A more complex disturbance scenario is shown in Figure 3-10, where an LT happens at 4.0s and a GT happens at 16.0s. As seen from Figure 3-11 (c), the SVD-SCD

algorithm is able to accurately preserve the LT information at 4.0s. However, it fails to generalize the complex disturbance due to the high reconstruction errors during the GT and post-GT periods. However, the proposed SVD-CE algorithm succeeds in identify the LT and the GT and compresses the PMU data in a more accurate manner. As seen, the reconstructed data by the proposed approach is almost identical to the original data. Figure 3-11 shows the reconstruction results under a continued FO. As seen, the reconstruction performance of the proposed model is superior. The reason is that during the FO event, the off-nominal characteristics from the wide-area are obvious. Therefore, the SVD-CE based method successfully recognizes the disturbance periods and selects proper CR to compress the data. For the SVD-SCD approach, the statistics of the synchrophasor data keep changing, thus the SCD algorithm is constantly triggered and the whole period is recognized as a disturbance period. However, like Figure 3-8 (c) without bounding the TRET, the reconstruction error of the SVD-SCE approach is still much higher than that of the proposed SVD-CE method.

3.5 Discussion

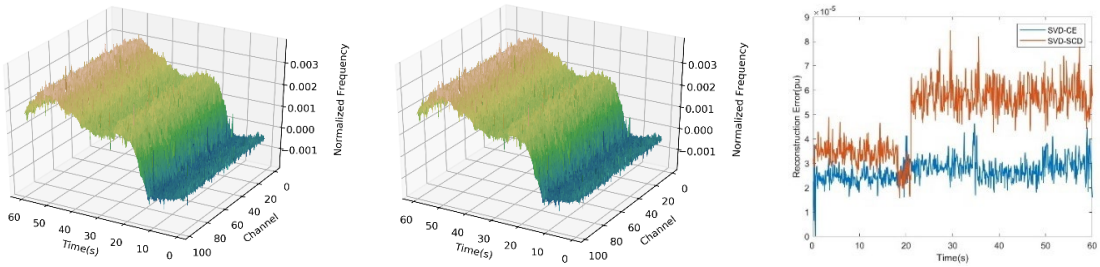
Online Implementation and Compression Time

Since the matrix factorization algorithms like SVD are usually costly in time **[49]**, it is necessary to evaluate its execution time for online adoption. Due to the high reporting rate of PMU, immediately processing the data once it arrives at the PDC may introduce unaffordable time overhead. Therefore, to overcome this issue, this paper follows a widely adopted batch-processing strategy **[50]** to perform an online data compression.



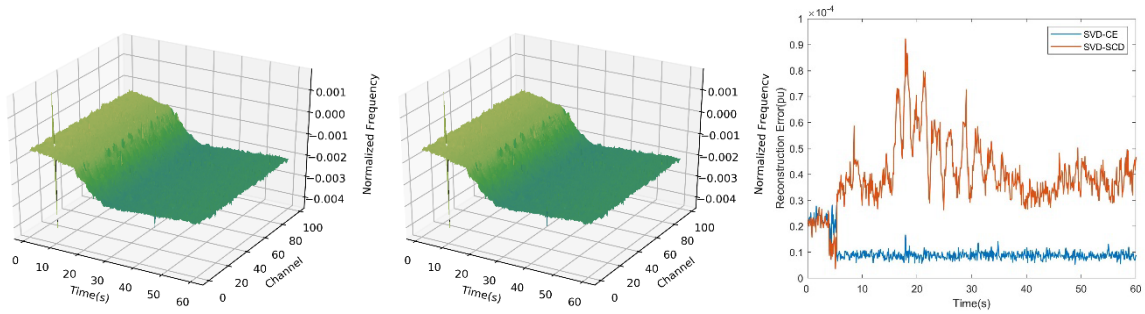
(a) Original (b) SVD-CE (c) Reconstruction Errors

Figure 3-8 Reconstruction performance on frequency under simple GT (110 units)



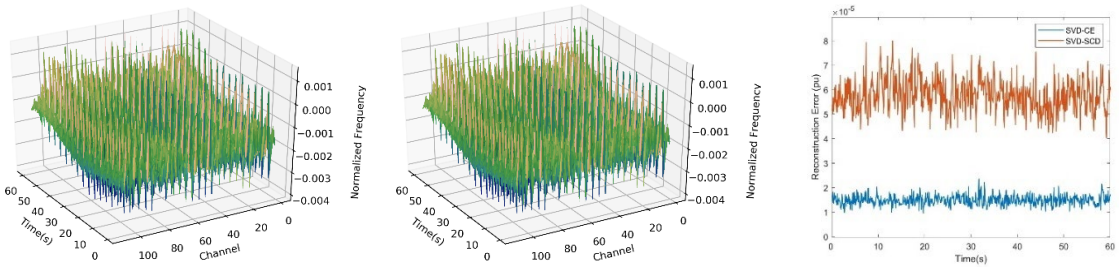
(a) Original (b) SVD-CE (c) Reconstruction Errors

Figure 3-9 Reconstruction performance on frequency under simple LD (110 units)



(a) Original (b) SVD-CE (c) Reconstruction Errors

Figure 3-10 Reconstruction performance on frequency under LT & GT (110 units)



(a) Original (b) SVD-CE (c) Reconstruction Errors

Figure 3-11 Reconstruction performance on frequency under simple FO (110 units)

The experimental computer is equipped with an Intel Core i7-8700 3.20GHz CPU, 16.0GB memory, and Python 3.7. In this paper, when the PDC receives the measurements, it caches the data, where the cache has a capacity. Then, once the capacity is exceeded, the PDC invokes the compression algorithm to compression the cached data chunk. Here, the capacity of the data chunk is defined as 600, which equals to 60-second data at 10Hz reporting rate. The choice of the 60-second windows is primarily due to the requirement of disturbance analytical applications **[51]**, **[52]** and it equals to around 3 megabytes (MB) data in the memory. Table 3-3 shows the average time consumption of the proposed algorithm. Utilizing the batch compression strategy, both 10Hz and 120Hz data can be compressed in a short time. It is worth to note the compression time of the field-collected 10Hz data is greater than that of the simulated 120Hz data. This is because the field-collected has greater variations, introducing greater entropy. Given this fact, the SVD-CE algorithm tends to search exhaustively for the best CR, so it takes a relatively long time to execute. However, under both scenarios, the compression procedure catches up well with the data collection procedure.

Choice of coefficient λ

In this paper, the λ is set to 0.05, tolerating a maximum of 5% reconstruction error. Setting λ to 0.05 is mainly due to the requirement from compliance investigation. However, the determination of the λ depends on how the reconstructed will be used. For example, if there is no concern about the accuracy of each PMU's data, the λ may be set to a much large value, e.g. 0.2.

Table 3-3 Average Time Consumption

	Field Data (10Hz)		Simulated Data (120Hz)	
	Data Length	Compression Time	Data Length	Compression Time
Ambient	18.3s	46.0ms	1.6s	7.2ms
	28.0s	68.3ms	3.4s	18.8ms
	46.3s	114.3ms	4.8s	25.6ms
Event	2.6s	49ms	0.3s	2.1ms
	7.6s	98.9ms	0.7s	5.7ms
	12.2s	150.1ms	1.1s	7.6ms

The advantages of setting the λ to a larger value are it speeds up the execution of the compression procedure, and it usually acquires a high CR. However, under this condition, some important information will not be retained during disturbance periods. On the other hand, if there is a stringent requirement of data accuracy, e.g. compliance purposes, the λ shall be set to a lower value. Otherwise, the inaccurate reconstructed data could eventually result in a financial loss of power companies because it brings inaccuracy to the compliance investigation. Moreover, in this paper, the performance evaluation indicates that the 0.05 value of the λ is independent of data types, data volumes, disturbance types, etc. and using this value meets the requirement of compliance standards. However, it is urged to keep the λ within 0.1 (10%), where a series of data loss is observed under data reconstruction.

CHAPTER FOUR
DEVELOPMENT OF THE SYNCHROSERVICE SYSTEM

4.1 Introduction

A phasor measurement unit (PMU) is one of the core components in a synchrophasor network. It provides high-resolution, rich-type, and GPS-synchronized measurements to help understand the real power system dynamics. Many synchrophasor applications including frequency monitoring [91], disturbance detection [92], frequency response analysis [93], visualization [94], cyber security [95], etc. are developed to assist the day-to-day operation of power grids. The synchrophasor measurements from PMUs are collected by phasor data concentrators (PDC). In recent years, with the expansion of the PMU network and the increasing of the PMU reporting rate, the data volume is growing rapidly. Take the example of frequency monitoring network (FNET/GridEye) [91], a distributed level wide area monitoring system (WAMS). The number of the deployed synchrophasor measuring devices (SMD) has grown to 610 as of Jan. 2019. If each SMD sends its data at 10Hz rate, the server will receive over 8 gigabytes of data in total per day, not to mention the 120Hz mobile device PMU (MDPMU) that has 120Hz reporting rate [96]. Moreover, according to a report published by NASPI, over 1700 production level PMUs are deployed in North America [97]. Given the typical 30 Hz reporting rate, these units are creating over 80 gigabytes data per day. Obviously, the growing number of data is challenging the data collection, storage, and query of the nowadays synchrophasor network.

For the data collection side, some efforts have been made on developing low-cost, low-

latency, high-throughput, highly reliable PDCs [98]-[101]. For the data storage, MySQL [99], BTrDB [102], Hadoop [103], etc. have been exploited to store the synchrophasor data. However, few has studied the data retrieval issue for synchrophasor systems. In fact, a data retrieval system is an important component that supports critical functionalities including situation awareness, operation, compliance, etc. On the other hand, with the increasing penetration of distributed energy resources (DER) [104], the operational scheme [105] and the power system dynamics during disturbances with high DER penetration are different to conventional ones [88]. Therefore, existing applications are constantly improved to adopt to large-scale DER penetration [107], which further challenges the existing information system architecture. Moreover, although many advanced synchrophasor applications are developed to assist the monitoring, operation, control, etc., it is still hard to implement them due to the differences between programming languages. Therefore, it is urgent to develop an efficient software infrastructure to 1) facilitate the information exchange among regional coordinators, balancing authorities, and electric reliability organizations; and 2) connect the advanced synchrophasor applications to support the control room functionalities.

Towards this end, this paper develops a novel software as a service (SaaS) infrastructure, SynchroService, to increase the availability of synchrophasor systems. The proposed system exploits the latest advances in the micro-service technology, providing an efficient data collection, storage and distribution functionalities to facilitate the enterprise-level data exchange and to expose advanced synchrophasor applications as accessible services. In this paper, the developed SynchroService is deployed in the FNET/GridEye system, and its

performance is evaluated by some synchrophasor applications.

4.2 SynchroService Structure

The proposed SynchroService infrastructure consists of four layers, which are collection layer, analysis layer, storage layer, and service layer. The structure of the SynchroService infrastructure is shown in Figure 4-1.

Collection layer

As seen in Figure 4-1, in the data collection layer, SMDs connect to two PDCs through Ethernet via TCP/IP protocol. Once connected, SMDs send data frames that contain high-resolution phasor measurements to PDCs via standard PMU communication protocols including IEEE C37.118-2011 [26]. Firewalls are configured on the servers where the PDCs reside to only accept interesting traffic from authorized IPs. Once the data frames are received, the PDCs forward them to the data storage and data analytical layers for future processing.

It is worth to note that due to difference in the actual dynamics of PMUs, practical issues including filter-related timestamp shift, GPS-induced time inconsistency are common in commercial PMUs. To address these issues, the data collection layer performs an angle-frequency-based time shift correction when the data is received [90].

Analysis layer

In the analytical layer, two PDCs host a same set of applications simultaneously to provide critical functionalities including event detection, location, statistical analysis, etc.

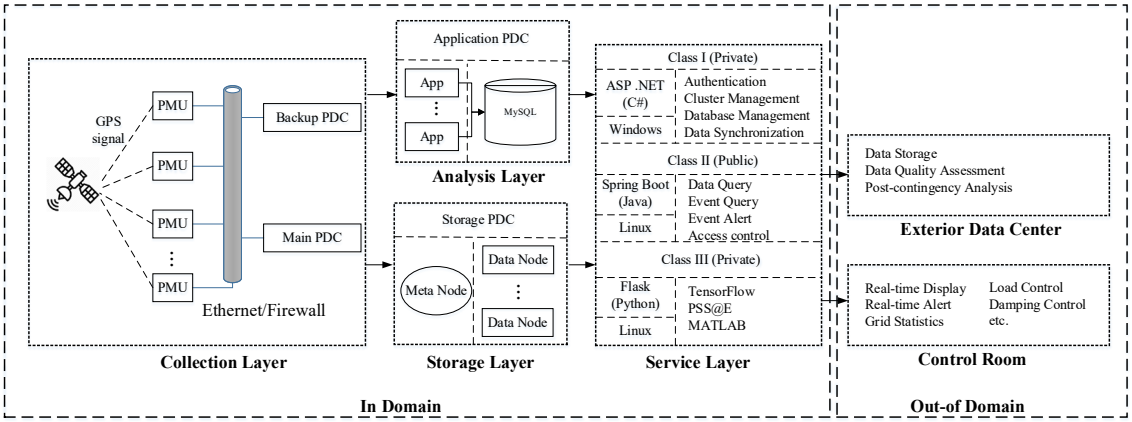


Figure 4-1 Architecture of the SynchroService system

Each application accepts the real-time synchrophasor data and uses dedicated algorithm to analyze it. If the analytical algorithm has an output, it will be stored in a relational database, such as MySQL [111] and an alarm request will be forward to the service layer.

Storage layer

In the storage layer, two types of nodes, the meta node and the data node are defined. A meta node keeps a list of canonical SMD configurations as a reference but does not store the synchrophasor data. A data node keeps a local copy of the canonical SMD configurations and stores the associated synchrophasor data. Changes in the SMD configurations will be populated among the data nodes to ensure data consistency. To improve the data availability, the meta nodes and the data nodes are implemented in a fully redundant mode. Due to its efficiency in data storage and query, this paper uses an open source time-series database to store synchrophasor data on data nodes [112].

Service layer

The service layer connects synchrophasor applications to the analysis layer and storage layers, providing application programming interfaces (API) for synchrophasor application to 1) query the synchrophasor data, 2) retrieve the analytical results, 3) perform remote procedure call (RPC). As Figure 4-1 shows, the service layer is implemented via a micro-service technology in different programming languages. The main reason is to reduce the number of cross-language RPCs, which add time delays to the proposed system.

The service layer includes 3 service class. The class I services implement fundamental functionalities that support the SynchroService infrastructure. Such functionalities include

active domain (AD) authentication, data node management, database management, configuration synchronization, internal logging. For security consideration, the first service class is implemented via ASP .NET as a private component, which is only visible to in-domain users. The class II services provide functionalities that support synchrophasor applications. Such functionalities include synchrophasor data query, disturbance events query, real-time alarms, access control, etc. The class II services are implemented in Spring Boot as representation style transfer (REST) APIs via JavaScript Object Notation (JSON) format. The class III services are implemented in Flask as a public component which is visible to out-of-domain users. The class III services provide RPC interfaces for users to invoke advanced analytical programs including TensorFlow, PSS@E, MATLAB, etc.

4.3 Result Formats

The result format may vary based on the functionality. As seen from Figure 4-2 (a), for synchrophasor data query, the protocol includes

- Status_word (**NR**): The status of the query result.
- Grid_ID (**NR**): The ID of an AC-connected power grid.
- Signal_ID (**NR**): The ID of the phasor signal.
- Timestamp (**R**): The initial time of the event.
- Value (**R**): The measurement value.
- Data_quality (**R**): The quality of Value.

Here, **NR** indicates that the field is non-repeatable, while **R** indicates the field is repeatable. Therefore, the response to each synchrophasor data query request may contain multiple records, which allows the flexibility to support various applications.

As seen from Figure 4-2 (b), for event analytical result query, the protocol includes

- Status_word (**NR**): The status of the query result.
- Event_ID (**NR**): The ID of the event.
- Grid_ID (**NR**): The ID of an AC-connected power grid.
- Timestamp (**NR**): The initial time of the event.
- Field (**R**): A feature value of the event.

Here, the Field is a repeatable key-value pair. A key indicates the name of an analytical result (e.g. event magnitude, event coordinate, etc.). The number of Fields depends on the type of the event.

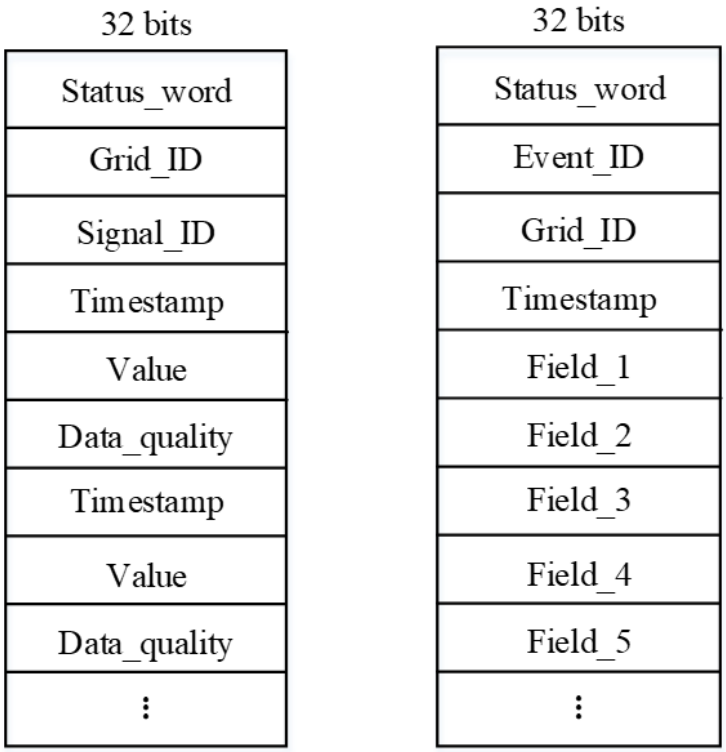
4.4 Performance Analysis

This paper uses query speed and throughput to measure the performance of the developed infrastructure against the conventional database [99].

The query strategy utilizes the computational capacities of the data nodes. Assuming the computational capacities are the identical for each data node, the total approximated query time is

$$T(n) = t(n/k) + k(t_p) + max(t_r^i) + t_s(k) \quad (1)$$

where $T(n)$ is the total query time for n data points, k is the number of the data nodes, $t(n/k)$ is the time to query n/k data points in a data node, t_p is the time overhead to transmit a query request to a data node, t_r^i is the time to return the i_{th} result, and $t_s(k)$ is the time to reduce k results. In practice, t_p is negligible.



(a) Synchrophasor data (b) Event analytical result

Figure 4-2 Data format

Furthermore, if assuming a high-speed internal network, the t_r^i is also negligible. Therefore, the query time can be written as

$$T(n) = t(n/k) + t_s(k) + a + b \quad (2)$$

where a represents the latency of the partition procedure, and b represents the of returning partial results.

Note we assume the computational abilities of the slave nodes are identical. Hence, if the sub-requests are sent sequentially, the partial results will be returned in partially sorted order. Under this condition, a linear time sorting algorithm can be used to sort the partial results [113], such that

$$t_s(k) = O(k) \quad (3)$$

In conclusion, theoretically, given a fixed number of data points, the query time will decrease while the number of the partitions increases. Table 4-1 shows the total query time comparison between the SynchroService and the MySQL database at various down sample rates (DSR) and data points level. In Table I, the total query time includes the time to transmit the request, to query the data, and to transfer the data. As seen from the figure, the query time of the MySQL database grows linearly as the requested data volume increases. As opposed to it, the developed SynchroService has good data query speed. Under a low DSR and small data volume, the developed system may support synchrophasor-based applications including monitoring and control [114].

Furthermore, it is also seen that the throughput of the system increases with the requested data volume. This is due to the parallel query mechanism. For large-volume data request, the data query request is broken into k sub-requests. Each sub-request is sent to a data node, thus the amount of data requested for a single data node is reduced.

Table 4-1 Data Query Performance Comparison

Data Points	DSR	Query Time		Throughput	
		Synchro-Service	SQL Method	Synchro-Service	SQL Method
2835	4s	16.5 ms	67 ms	5.50 MB/s	1.23MB/s
11340	1s	206 ms	268 ms	1.81 MB/s	1.39MB/s
113400	0.1s	504 ms	2.68 s	7.26 MB/s	1.35MB/s
226800	0.1s	880 ms	5.36 s	8.25 MB/s	1.35MB/s

As a result, to query same amount of data, the SynchroService generally consumes less time, making its throughput higher.

4.5 System Security

System security defines which part of the system is accessible by a specific user. In the SynchroService system, users are categorized into several types such are developer, administrator, domain user, and outside user.

Developers can access all parts of the system including the file system (FS). Developers can view the logs of the system, which are written into the FS, for debugging. Developers can also write logs into the FS to do troubleshooting in real-time.

Administrators can access most parts of the system including a read-only privilege to the FS. Administrators can view the logs of the system, restart the system, invoke application programming interfaces (APIs). and manage the members of the system.

Domain users have access to Class II and III APIs. For the domain users, no further restriction is placed when invoking the APIs. Outside users only have access to Class II APIs. Outside users are further categorized into several trusting levels. Different restricts are put onto these trusting levels. Generally, the SynchroService system puts stricter restricts as the trusting level goes lower. Table 4-2 shows the relationship of the four user types, where R, W, and E represent read, write, and execute respectively.

4.6 Communication Protocols

The SynchroService system provides two ways for the users to invoke.

Table 4-2 Security system design

	FS access	Service Management	Private APIs (Class II&III)	Public APIs (Class III)	Trusting level-based restrictions
Developer	R/W/E	E	E	E	R/W
Administrator	R	E	E	E	R/W
Domain users			E	E	
Outside users				E	

First, simple object access protocol (SOAP) is a permitted protocol. SOAP is an extensible and independent communication protocol that is especially suitable for web services. SOAP uses extensible markup language (XML) to wrap the request as a simple object for the SynchroService to parse. The main advantage of using SOAP as the protocol is its independence. SOAP allows users to invoke web services and receive responses independent of programming language and operating system. Figure 4-3 demonstrates the template of a SOAP request and a SOAP response the web function "GetMultipleEvents". In this figure, part (a) is the SOAP request, while part (b) is the SOAP response. In the both parts, an HTTP header covers information from the start to "Content-Length: length", while the HTTP content convers information sfrom "?<xml …" to "</sopa12:Evelope>". For the request, there are five parameter wrappers (eventType, username, userPwd, startTime, endTime), which are further wrapped by a "GetMultipleEvents" markups. These parameters must be given actual "string" values when the user tries to invoke the web function. On the other hand, for the response, there is a return value, which is wrapped by the "GetMultipleEventsResult" markup. SOAP is by default using HTTP POST. Alternatively, the user can also use pure HTTP GET/POST method to invoke the web function. For the HTTP GET method, the user can specify the values of parameters in the uniform resource locator (URL) directly, which is included in the HTTP header. Then, the response will be returned as an XML file. For the HTTP POST method, the user should specify the values of the parameters in the HTTP content instead of the HTTP header.

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FNETServices/FNETServices.asmx HTTP/1.1
Host: powerit5.eecs.utk.edu
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetMultipleEvents xmlns="http://tempuri.org/">
      <eventType>string</eventType>
      <userName>string</userName>
      <userPwd>string</userPwd>
      <startTime>string</startTime>
      <endTime>string</endTime>
    </GetMultipleEvents>
  </soap12:Body>
</soap12:Envelope>
```

(a) SOAP Request

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetMultipleEventsResponse xmlns="http://tempuri.org/">
      <GetMultipleEventsResult>string</GetMultipleEventsResult>
    </GetMultipleEventsResponse>
  </soap12:Body>
</soap12:Envelope>
```

(b) SOAP Response

Figure 4-3 SOAP Communication format

HTTP GET

The following is a sample HTTP GET request and response. The placeholders shown need to be replaced with actual values.

```
GET /FNETServices/FNETServices.asmx/GetMultipleEvents?eventType=string&userName=string&userPwd=string&startTime=string&endTime=string HTTP/1.1
Host: powerit5.eecs.utk.edu
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

HTTP POST

The following is a sample HTTP POST request and response. The placeholders shown need to be replaced with actual values.

```
POST /FNETServices/FNETServices.asmx/GetMultipleEvents HTTP/1.1
Host: powerit5.eecs.utk.edu
Content-Type: application/x-www-form-urlencoded
Content-Length: length

eventType=string&userName=string&userPwd=string&startTime=string&endTime=string
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

Figure 4-4 HTTP Communication format

CHAPTER FIVE
ADVANCED GRID VISUALIZATION APPLICATIONS

5.1 Introduction

Nowadays, with the advances of synchrophasor technology, more accurate, small-granularity data becomes available to the operators. However, in the modern control center, the synchrophasor data is usually displayed as plots and is not fully exploited to help the day-to-day applications. First, synchrophasor data has not been utilized in the real-time reliability assessment in a visual manner. Moreover, synchrophasor data has not been utilized with other data such as critical infrastructures to provide operators a comprehensive situation awareness tool to pinpoint the fault.

Therefore, in this chapter, some advanced grid visualization applications are developed to assists operators on various perspectives including resource monitoring, reliability assessment, fault diagnostics, etc.

5.2 Resource Adequacy Visualization Tool

Introduction

As an indicator for bulk power system reliability, resource adequacy represents the ability of the electric system to supply the aggregate electric power and energy requirements of electricity consumers at all times, considering scheduled and expected unscheduled outages of system components. Furthermore, resource adequacy is usually measured within a balancing area. Due to the importance of the resource adequacy, it is

urgent to develop a visualization software that dynamically displays the resource adequacy in the balancing areas.

System structure

The resource adequacy visualization tool is built on an OSIsoft PI historian, which receives the real-time measurements via the SynchroService system. The visualization tool then retrieves and displays the real-time area control error (ACE) data in a dynamic mode so that operators may monitor the demand and supply relationship of each reliability coordinator (RC) and balancing authorities (BA).

Example

Figure 5-1 shows an example of the developed system. In this figure, the visualization at the center is displaying the ACE data of three North American interconnections (Eastern, ERCOT, and Quebec) at the RC level. In the figure, a red or orange colored area represents that this RC is supplying more power than its demand, while a blue or green colored area represents that this RC is supplying less power than its demand. The cells at the bottom are an information board showing current entity's ACE value and description, a list of ACE values for all RCs, a list of ACE values for BAs, and a system info console. By default, the list of BA ACE values is empty. It will only be shown when the operator clicks on a row in the list of RC ACE values. The tree view on the right is showing the RCs and the BAs within their geographical footprints. Clicking and "+" button of an RC will expand the view to show BAs. Clicking any item in the tree view will update the information of the information board (left most cell at the bottom).

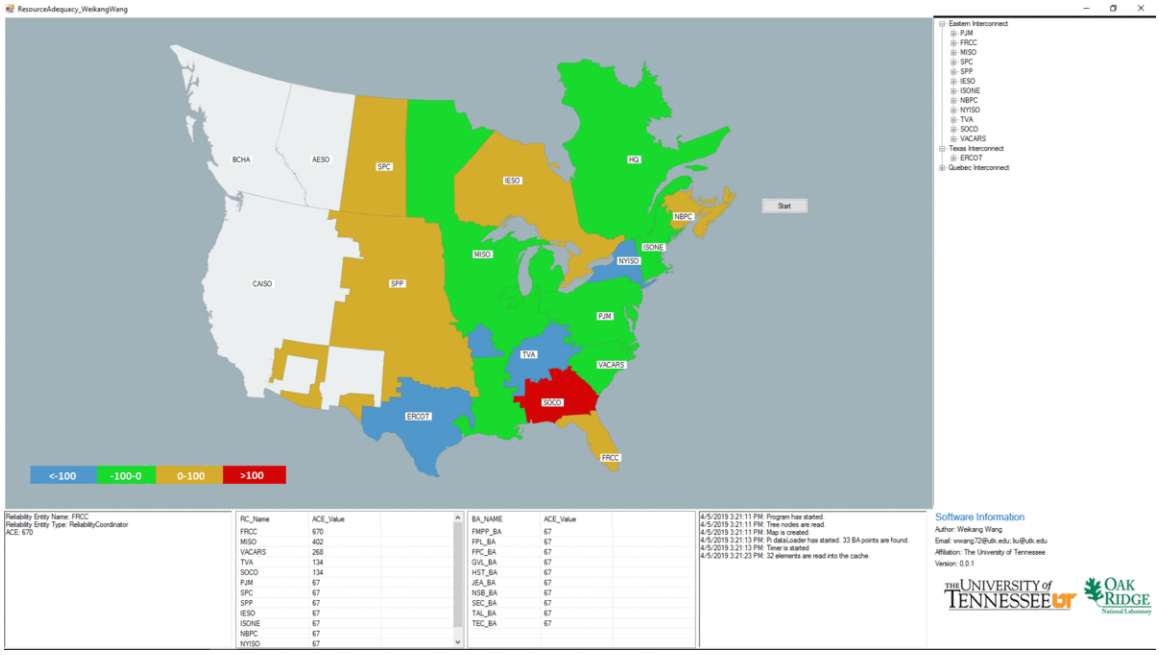


Figure 5-1 HTTP Communication format

5.3 Balancing Authority ACE Limit Visualization Platform

Introduction

Balancing authority ACE limit (BAAL) is defined in the BAL-001-2 standard. BAAL measures the operating performance of balancing authorities (BAs) across the North American. In the standard, it is required that the clock-minute average of Area Control Error (ACE) does not exceed its BAAL for more than 30 consecutive clock-minutes, it is an urgent yet practical need to develop a real-time visualization platform to help operators assess the operating performance from the BA level.

System structure

The resource adequacy visualization tool is also built on an OSIsoft PI historian, which receives the real-time measurements via the SynchroService system. The visualization tool then retrieves and displays the real-time area control error (ACE) and system median frequency data in a dynamic mode so that operators and auditors may view the performance of each RC against the NERC's reliability standard. Last but not the least, this system connects to the RCIS message system via Microsoft outlook, so that it can retrieve the time error correction (TEC) schedules since it will change the scheduled frequency.

Example

Figure 5-2 shows an example of the BAAL visualization platform. The platform mainly includes two modules.

The first part is the visualization module. The center cell is a dynamic visualization of the RCs' performance against the standard BAL-003 in the U.S. Eastern Interconnection.

The horizontal axis represents the frequency while the vertical axis represents the ACE. Two curves are drawn on the visualization to display the exceedance areas. Each solid circle represents the current frequency-ACE standing of an RC. When they are not in the exceedance areas, they are colored in green, while when they are in either exceedance area, their colors will change to the exceedance's boarder's color. The top right cell shows a list of all BAs and their current ACE data.

The second part is the TEC schedule module, which is in the bottom left corner. In this module, the engineers can set up TEC schedules to update the visualization. In this module, operators can create, edit, and synchronize TEC schedules. As Figure 5-3 shows, clicking "New" or "Edit" button pops up a dialog that allows the engineer to type in the information of the TEC schedule. After clicking confirmed, the TEC schedule will show up in the TEC schedule module's list display. Besides, the engineer can click "Pull TEC" button to pop up another dialog, as shown in Figure 5-4. Here the engineer can click "Sync RCIS emails" button to allow the system to retrieve the RCIS emails and parse them into TEC schedules. Then, the engineer may click on a TEC schedule and click "confirm", to add it into the TEC schedule module's display.

5.4 Situation Awareness for NERC, FERC and Regions (SAFNR)

Introduction

Situation awareness in crucial to the operation of an electric power system since operators may assess the current situation and take actions to mitigate the impacts caused by faults, disturbances, etc.

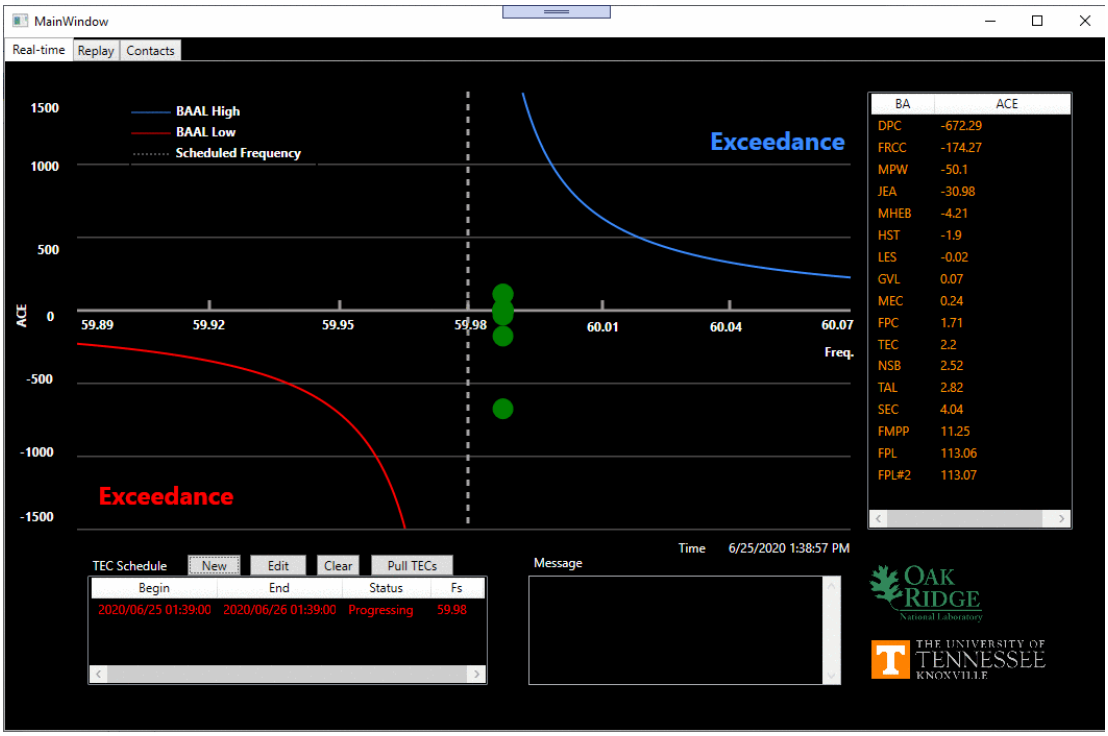


Figure 5-2 BAAL Visualization Platform - Main Display

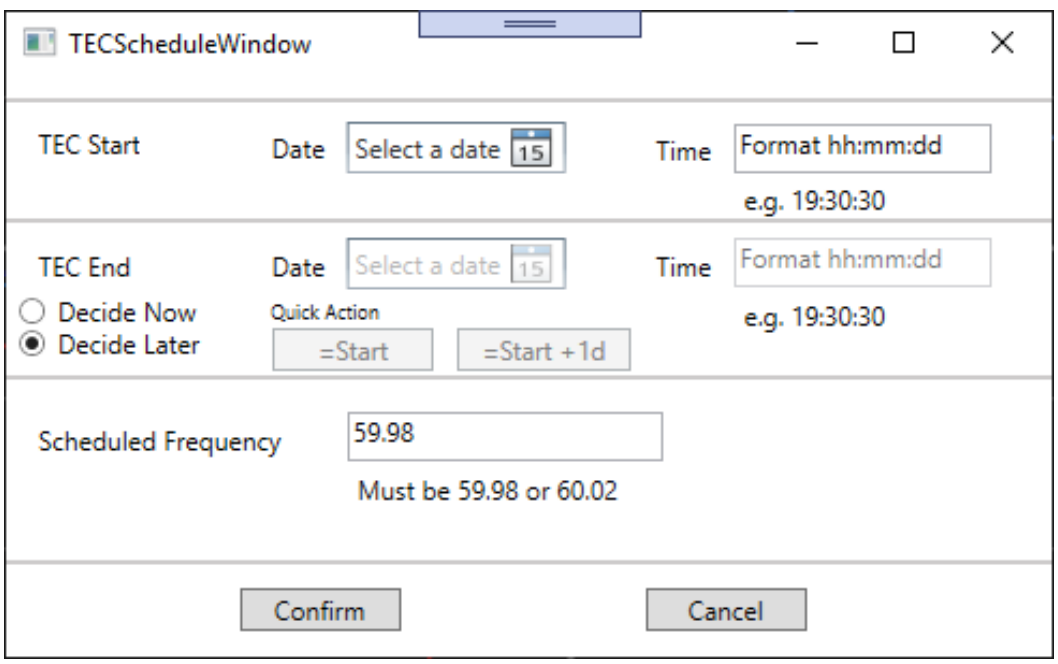
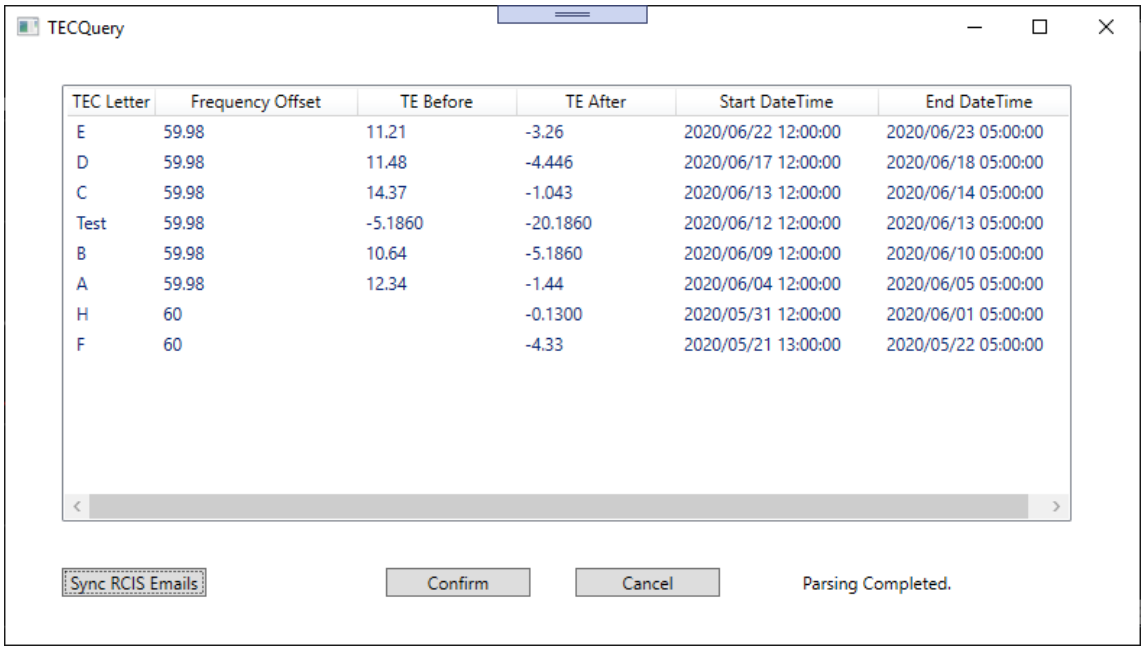


Figure 5-3 BAAL Visualization Platform – TEC Schedule Subsystem



TEC Letter	Frequency Offset	TE Before	TE After	Start DateTime	End DateTime
E	59.98	11.21	-3.26	2020/06/22 12:00:00	2020/06/23 05:00:00
D	59.98	11.48	-4.446	2020/06/17 12:00:00	2020/06/18 05:00:00
C	59.98	14.37	-1.043	2020/06/13 12:00:00	2020/06/14 05:00:00
Test	59.98	-5.1860	-20.1860	2020/06/12 12:00:00	2020/06/13 05:00:00
B	59.98	10.64	-5.1860	2020/06/09 12:00:00	2020/06/10 05:00:00
A	59.98	12.34	-1.44	2020/06/04 12:00:00	2020/06/05 05:00:00
H	60		-0.1300	2020/05/31 12:00:00	2020/06/01 05:00:00
F	60		-4.33	2020/05/21 13:00:00	2020/05/22 05:00:00

Figure 5-4 BAAL Visualization Platform – TEC Query System

Situation Awareness for NERC, FERC, and Regions (SAFNR) is a software platform that visualizes the transmission lines in USA. To enhance the situation awareness capability of the SAFNR system, it is of a great significance to include the critical disturbance detection functionalities in such a system. With the development of the SynchroService, disturbance event information can be easily share with the SAFNR platform to provide an instant notification to operators and immediately raise their attentions. The SAFNR desktop application directly connects to the SynchroService and periodically queries the most recent forced oscillation disturbance event. Whenever a new event is detected, the application updates its interface to show the event.

Example

Figure 5-5 demonstrate a snapshot of the SAFNR application, which incorporates the forced oscillation information from the FNET/GridEye. As seen, the measuring units that first detect the forced oscillation as marked as red "FO" squares. Then, an elliptical is drawn to indicate the area, where the forced oscillation is possible to evolve. At the same time, the event record is inserted into a dashboard on the left. Operators may click on the record to update the UI for any specific event.

5.5 Reliability Monitoring for Advanced Grid Visualization Application

Introduction

Grid visualization applications are required to have high availability. The down time of such applications should be controlled as minimum in real-time operation.

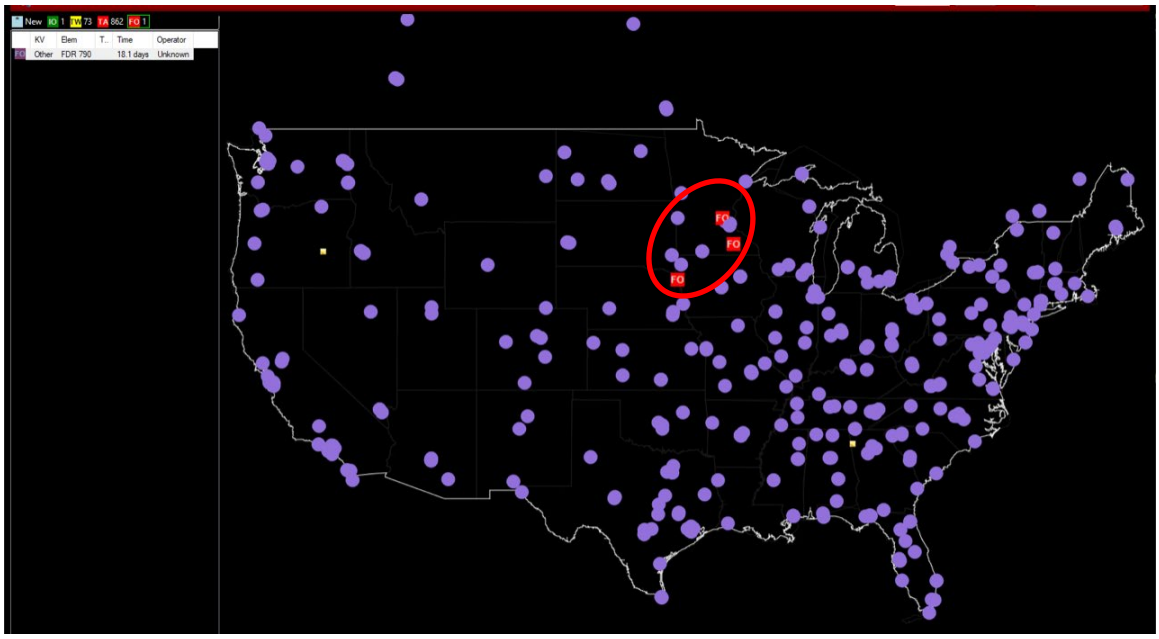


Figure 5-5 SAFNR_V3 Forced Oscillation Display

It is crucial to monitor the applications and ensure they are running in good status. The traditional approach to monitor a self-developed application are mainly using text logs. The advantage of this approach is its superior customizable ability. The software developers may design a text format, in which the logs are written, to provide a structured view of the system's behavior. Meanwhile, the disadvantages of this approach are obvious. First, it lacks a retention policy that periodically clears the historical logs that fall out of interest. This can be unaffordable at a long run since the log may take up much storage space. Second, it is hard to quantitatively visualize the system's behavior due to the lack of a visualization module. Third, the developer must develop a series of monitoring programs to translate the logs to meaningful information to alert users for system down times.

In this section, a time-series database-based approach is proposed to monitor the performance of the system. In the proposed approach, Applications send statistical information to the time-series database periodically. Then the time-series databases handles the data retention, visualization, and alerts.

System Structure

The time-series database provides an application programing interface (API) for the developer to insert data. At the same time, it stores the statistics of each application in a specific bucket. Each application uses this API to send formatted stats to the time-series database. The time-series database also includes some preconfigured alerting rules, against which the newly received stats are evaluated. If the alerting rule passes, the system will alert the users.

Example

Figure 5-6 shows a snapshot of the Product Stats dashboard. It displays the statistics of all products provided by the GridEye system and provides a core reference when abnormalities happen. In this dashboard, administrator can specify a timespan, within which the data is retrieved to check the health of the GridEye system as a whole for the past several hours, and days. Each cell in the dashboard is configured with a corresponding CA rule, which ensures the product is running in good health. Once a CA rule is violated, the monitoring system will send an alert to the information hub, based on Slack. As Figure 5-7 shows, a formatted message will present the product that violates the CA, and the violation's datetime and severity.

5.6 Resource Mix Change Observation using Inertia and Load Data

Introduction

Nowadays, power grids, like ERCOT, have seen a steady increase on load due to the growing population. To satisfy the increasing load, more and more power resources are brought into the power systems. However, the increasing introduction of inverter-based resources leads to a non-linear change of the inertia-to-load ratio. This means although the overall system inertia may increase, the integration of more inverter-based resources can make the increase insufficient with respect to the increase of load. This impedes the reliability of the grid since the inverter-based sources usually contributes very low inertia in frequency response. Therefore, it is important to track and understand the inertia-to-load ratio to understand how resource mix affects the inertia of the grid.

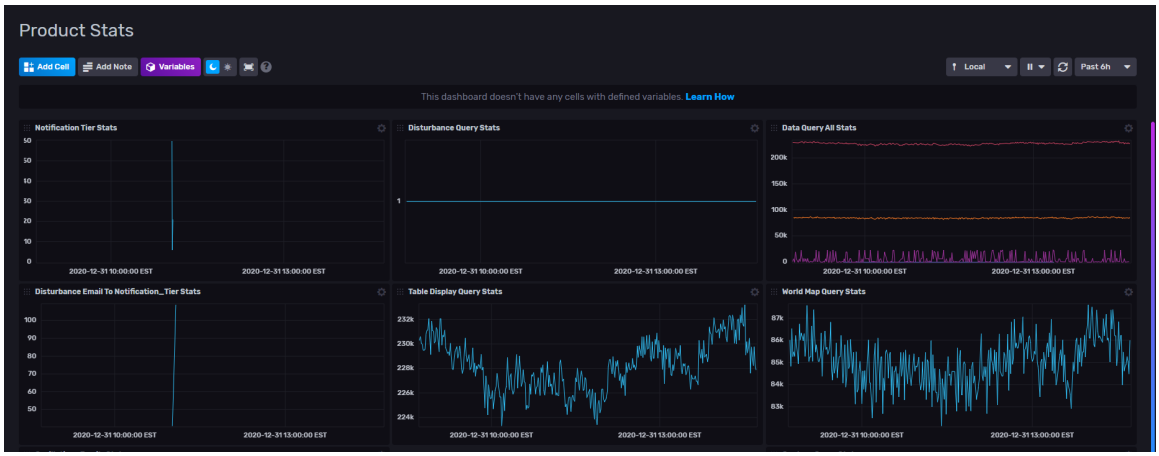


Figure 5-6 SynchroService Internal Monitoring

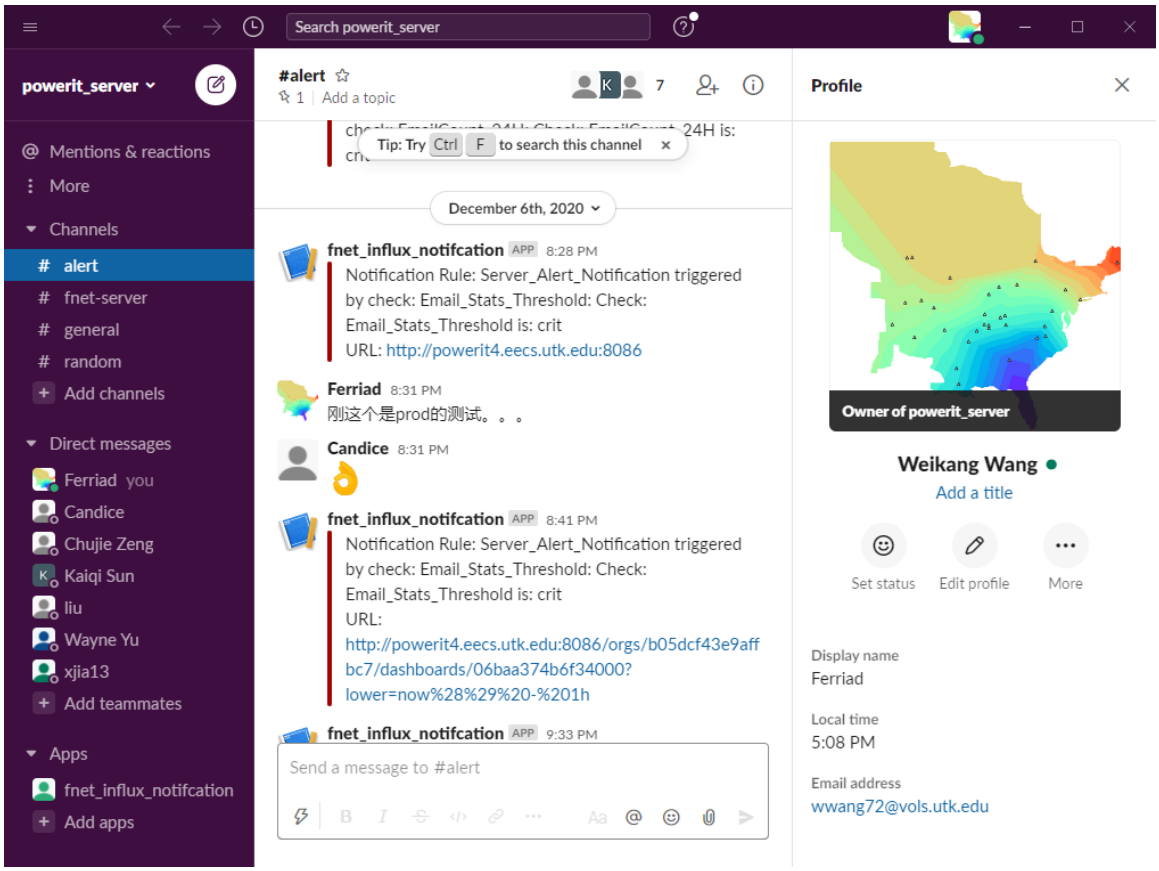


Figure 5-7 SynchroService Internal Alerting

To track the change of the inertia-to-load ratio, this section explores the seasonal load and inertia data of ERCOT to depict the change of resource mix from 2016 to 2020. The result shows that the inertia-to-load ratio keeps decreasing through the years, which may indicate a decrease of the system's reliability.

Standalone Analysis of Inertia and Load

Power system inertia mainly comes from traditional energy sources including coal and/or combined cycle gas generation units. As seen from Figure 5-8 and Figure 5-9, in ERCOT, the inverter-based power generation has risen from around 15% to 25%, while the gas/coal-based power generation has seen a drop from 76.4% to 64% of the total generation [53]. Although the generation percentage drops, the gas/coal-based power generation has not seen an obvious drop, nor a rise, in its absolute value. This has led to a steady curve of system inertia. As seen from Figure 5-10 and Figure 5-11, the monthly minimum and maximum inertia does not see drastic changes in each season from 2016 to 2020. Nevertheless, the minimum monthly inertia does see some rises in the summers from 2016 to 2019, due to the intermittent nature of inverter-based resources and the needs to meet the load requirement during the day [53].

For load, with the rapid growth of move-in population, ERCOT has seen a steady increase in load in the past several years. As seen from Figure 5-12 and Figure 5-13, the load has been steadily increasing in most of the seasons from 2016 to 2019 despite a drop in summer 2020 due to the COVID-19 pandemic.

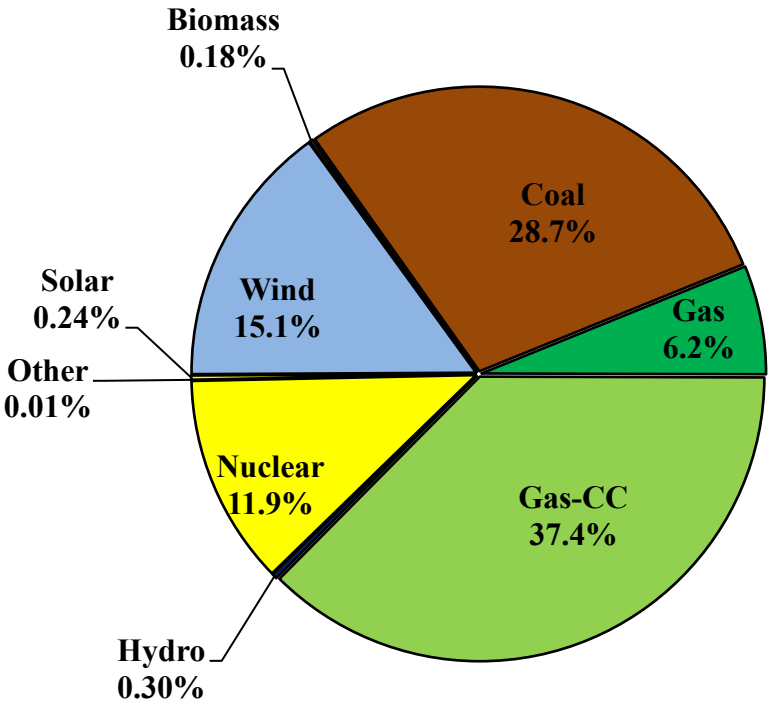


Figure 5-8 Generation Source Mix 2016

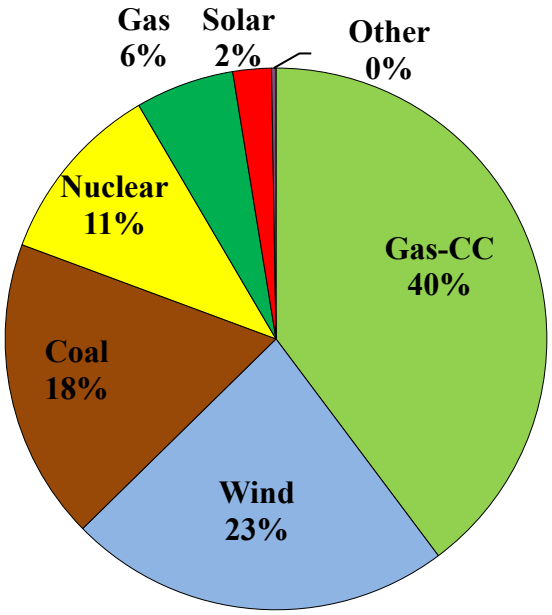


Figure 5-9 Generation Source Mix 2020

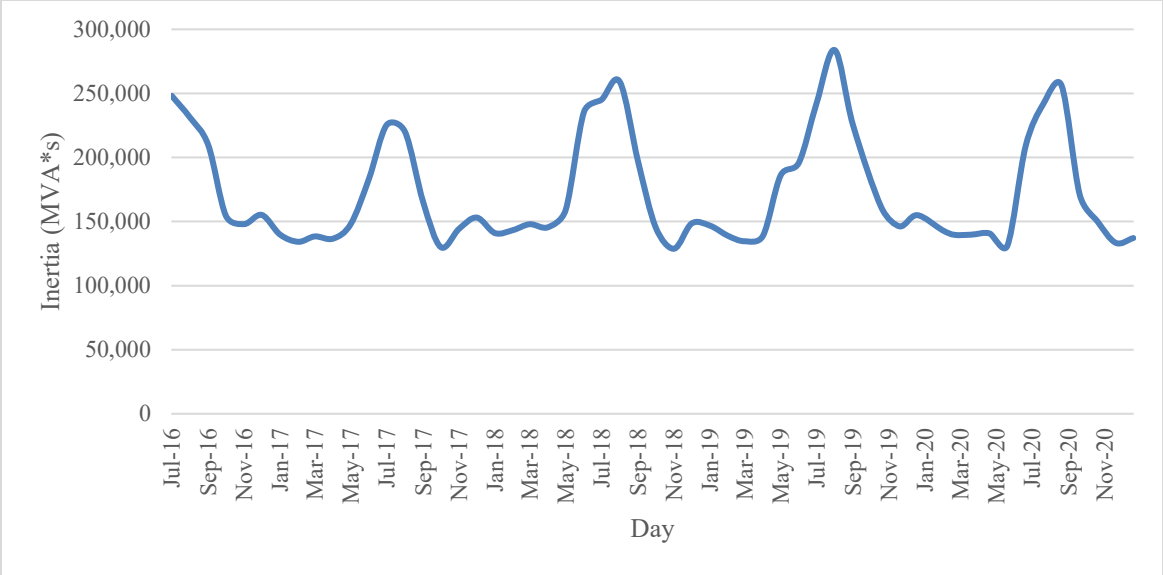


Figure 5-10 Monthly minimum inertia 2016-2020

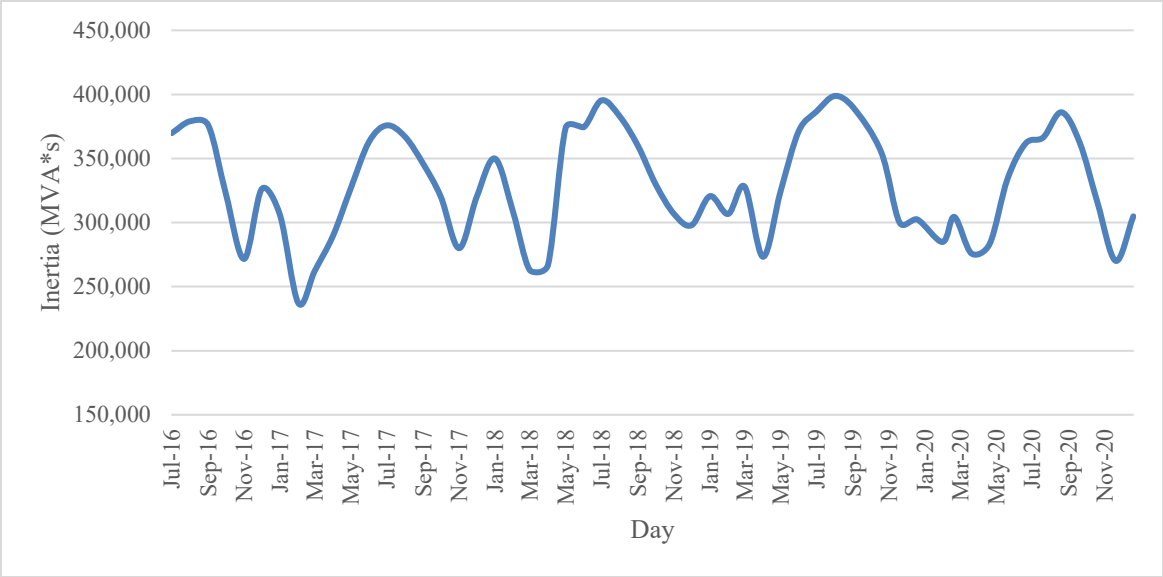


Figure 5-11 Monthly maximum inertia 2016-2020

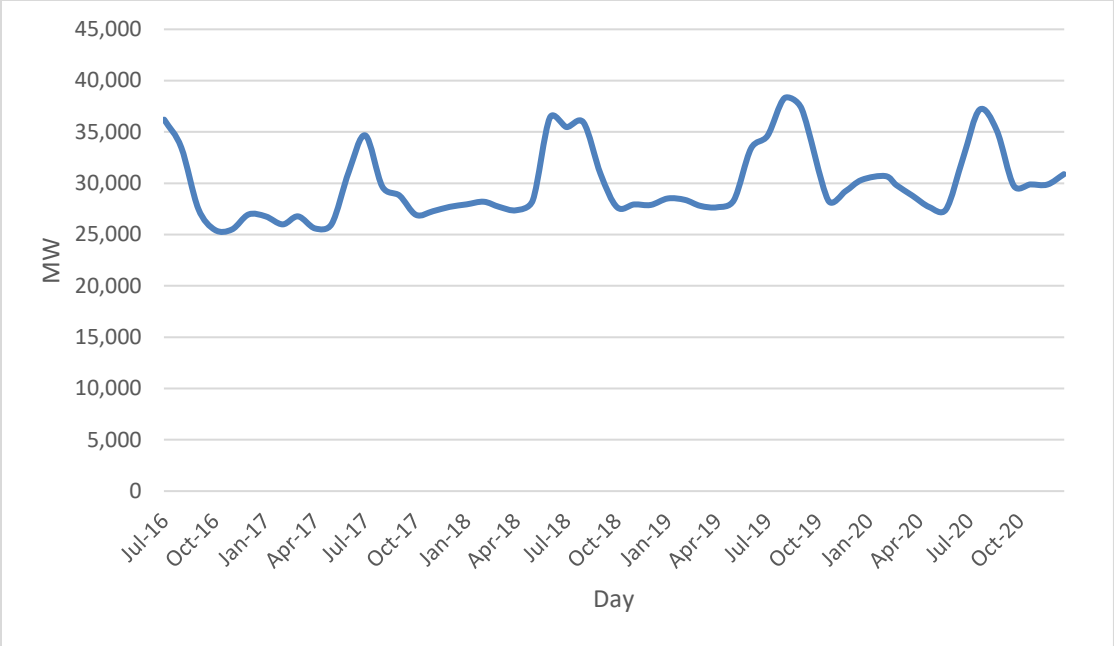


Figure 5-12 Monthly minimum load 2016-2020

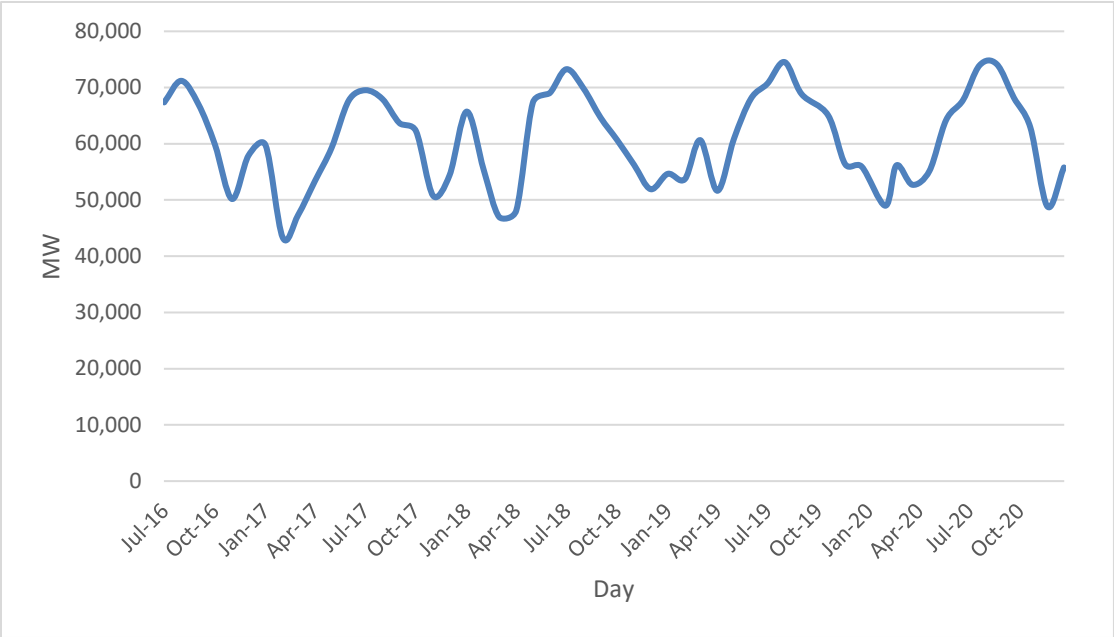


Figure 5-13 Monthly maximum load 2016-2020

Analysis of Inertia-load ratio

Figure 5-14 and Figure 5-15 show the minimum (MIN) and maximum (MAX) inertia-load ratio of ERCOT from 2016 to 2020, respectively. As seen, the MIN inertia-load ratio decreases with years. The clear decreasing trend is caused by several reasons. First, ERCOT has drastically increased the wind generation as a response to its growing electricity demand. This has led to more low-inertia generation units, while the load keeps rising. Second, the typical generation profile of ERCOT shows wind power does not align well with the market demand, because most of the wind generation happens at nights (20:30 – 4:30), when the load is low. Therefore, the decreasing trend of ERCOT's inertia-load ratio is obvious. As opposed to it, the MAX Inertia/load ratio does not have a clear trend. This is primarily because the high-load conditions are mostly seen during the day, where traditional generation units are dominant.

For a seasonal analysis, Figure 5-17 shows the generation mix in ERCOT from July 2016 to December 2020. As the figure shows, the inverter-based sources are increased with the years. There are three notable trends. First, wind power has been greatly increased, especially in summers. Second, coal-based generation has been decreased rapidly. Third, the solar-based generation has been growing fast. They all contribute to the decrease of the overall system inertia.

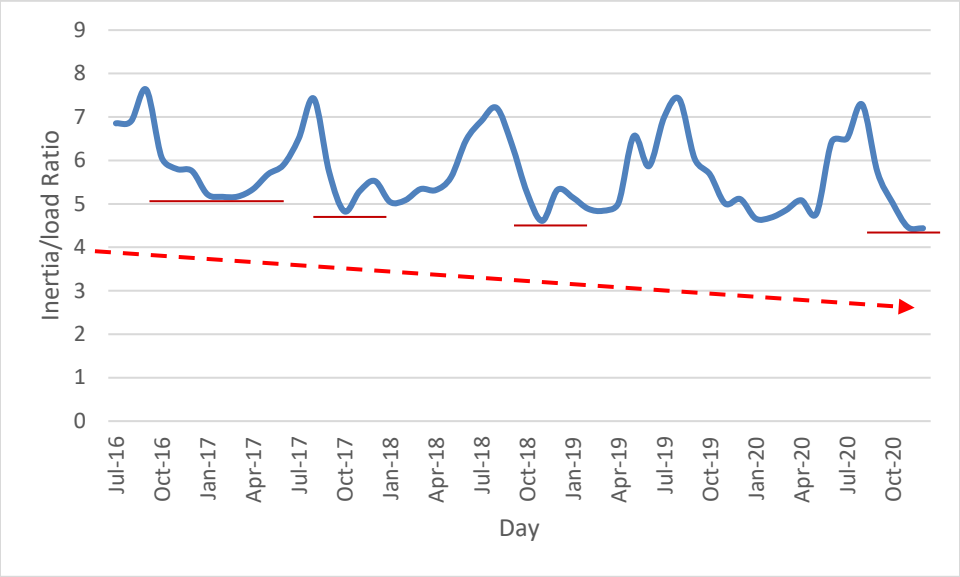


Figure 5-14 Texas Monthly MIN Inertia/Load Ratio 2016-2020

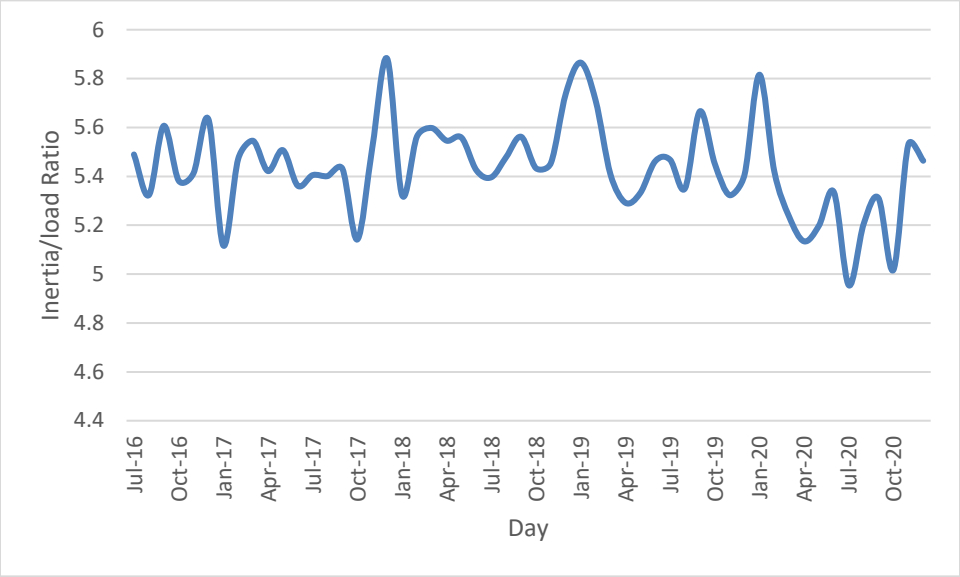


Figure 5-15 Texas Monthly MAX Inertia/Load Ratio 2016-2020

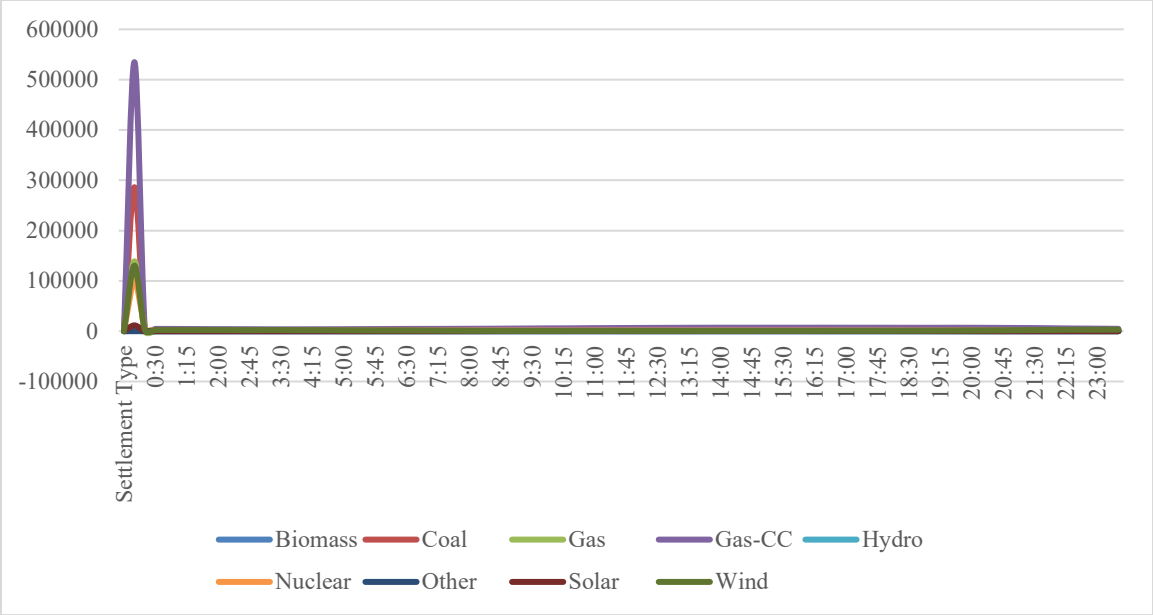


Figure 5-16 Typical Resource Mix under a High-load Season (August, 2020)

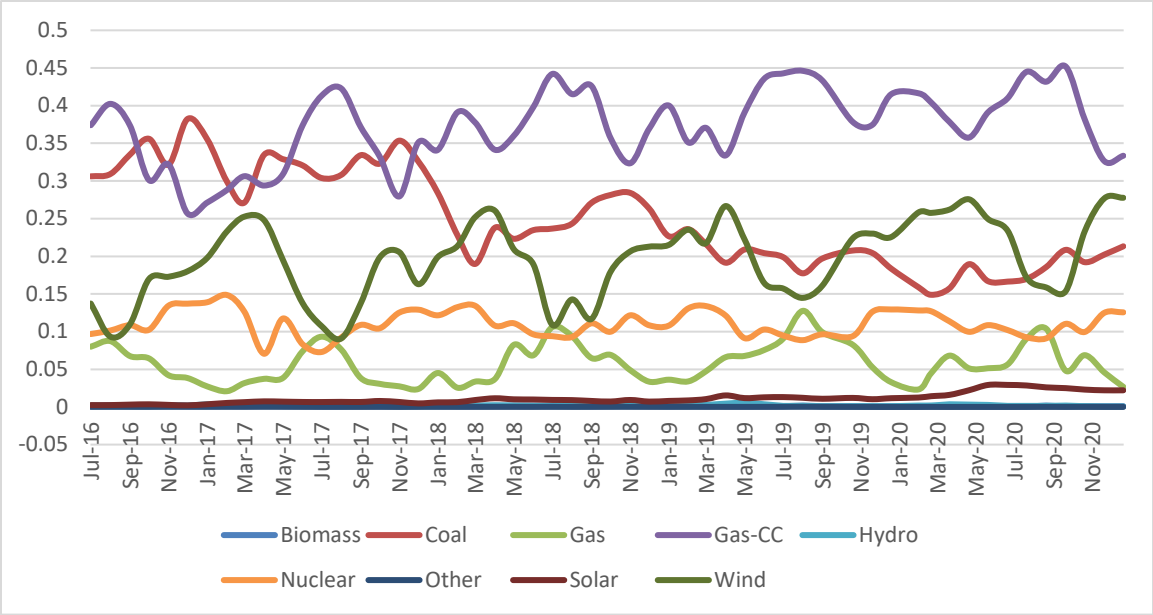


Figure 5-17 Monthly Generation Mix 2016-2020 (Percentage)

CHAPTER SIX DEEP LEARNING BASED EVENT DETECTION SYSTEM

6.1 Introduction

Power system disturbances can be caused from small to large impacts on the operation of an interconnected power grid. While a small disturbance may only cause a negligible frequency variation, a rather big disturbance usually implies more serious power quality issues. These power quality issues can greatly affect the grid and cause severe consequences such as large-scale blackout, which can cost up to $7-$10 billion [54]. The frequency disturbances are commonly caused by generator trips and load disconnections. Therefore, the detections of both events are of great importance in terms of monitoring the resource adequacy, pinpointing the fault location, and ensuring the reliability of the grid.

The detection of the frequency disturbance event has become easy to implement thanks to the invention of phasor measurement units (PMUs). A PMU provides GPS synchronized measurement of electrical quantities (synchrophasors) from across the power system. The increasing deployment of the PMUs gives a rather thorough understanding of the system dynamics, making the event detection and location more accurate. In recent years, PMUbs have been widely used in wide area monitoring [55], load control [56] and disturbance event detection and location [57]. PMU based event detection is mainly model-driven or data-driven. Model-driven approaches rely on a known system topology. Many model-driven approaches are proposed to address the detection of line outages [58] and oscillations [59]-[60]. Data-driven approaches are becoming popular with the advances on the computational power of the modern computers. Some data-driven models have been

proposed to detect simple frequency disturbances [61]-[64], and complex frequency disturbances [65].

With the evolutionary advance in combining the graphical processing unit (GPU) with deep learning, some have adopted deep learning in solving various power system problems including event classification [66], cybersecurity [67], wind forecasting [68], load forecasting [69], security screening [70], and power quality classification [71]. It may sound unusual to take advantage of an efficient, emerging, image recognition tool, convolutional neural network (CNN), to detect frequency disturbance events. However, due to its extraordinary feature generalization ability, it is logical to exploit this ability in detecting frequency disturbance events, which contain complex spatio-temporal characteristics.

Recently, a CNN based model was proposed to recognize the event type [66]. The experimental results show this model can successfully classify frequency events. However, this model, as well as other data-driven models, focus on exploiting the frequency signal alone to detect frequency disturbance events. A potential issue is that the frequency signal alone may not be accurate enough for event detection. Since the motor load provides frequency response to the power grid, under a light load condition, less frequency response can be obtained from the motor. In the meantime, the frequency will keep ramping down since not enough frequency response is provided, even if no event is involved [72]. In this scenario, many of the aforementioned models could have sent out false alarms due to the failure of telling a frequency ramping from a real event.

To address this issue, the relative angle shift (RAS) signal is introduced as another

indicator. An event creates an electromechanical wave that propagates through an interconnected power grid with finite speed [73]. The wave causes the angle shifts among the power grid [61]. Based on this fact, some literatures have proposed to use the phasor angle to locate event source [57], and detect single line outage [58]. On the other hand, as a transformation of the frequency signal, the rate of change of frequency (ROCOF) is used by some to detect frequency events [74]. Both works achieve good results since the ROCOF signal demonstrates more obvious frequency change characteristics under event conditions.

Since the ROCOF signal and the RAS signal are both good indicators for frequency disturbance events, this paper exploits the CNN model using both signals as inputs to build an efficient and accurate event detection model. This paper chooses two important types of events: generation trip (GT) and load disconnection (LD) [64] to compare the performance of the proposed model with the conventional event detection model, and frequency only CNN model [66].

The main contribution of the paper is four-fold: 1) it analyzes the angle-wise difference between an event and a frequency ramping; 2) it constructs informative ROCOF and RAS images for the CNN model; 3) it proposes a novel ROCOF_Net-RAS_Net model to detect events; 4) it conducts extensive evaluation of the proposed model using a large number of manually classified disturbance events from the U.S. eastern interconnect (EI).

The rest of this paper is organized as follows: Section 6.2 gives theoretical analysis and comparison of a generator trip and a frequency ramping. Section 6.3 demonstrates the proposed ROCOF_Net-RAS_Net model. Section 6.4 demonstrates the effectiveness and

the accuracy of the proposed model. Section 6.5 discusses requirements, potential limitations, and implementations of the proposed model, Finally, the conclusions and future work are discussed in Section 6.6.

6.2 Frequency Event Modeling

Consider a sinusoidal waveform with signal of frequency $f = 2\pi/\omega$ given by

$$x(t) = X\,cos(\omega t + \varphi) \tag{1}$$

where X is the amplitude, ω is the angular velocity, and φ is the phase angle.

The voltage angle of the generator with respect to a synchronously rotating reference is given by

$$\theta(t) = \omega_{syn} \cdot t + \delta \tag{2}$$

where ω_{syn} is the angular velocity of synchronously rotating reference and δ is the relative angle with respect to the synchronously rotating reference.

The swing equation is shown as follows

$$\omega = \frac{d\theta}{dt} = \omega_{syn} + \frac{d\delta}{dt} \tag{3}$$

(3) can be further written via frequency f, so that

$$f = f_{syn} + \frac{1}{2\pi} \cdot \frac{d\delta}{dt} \tag{4}$$

$$f - f_{syn} = \frac{1}{2\pi} \cdot \frac{d\delta}{dt} \tag{5}$$

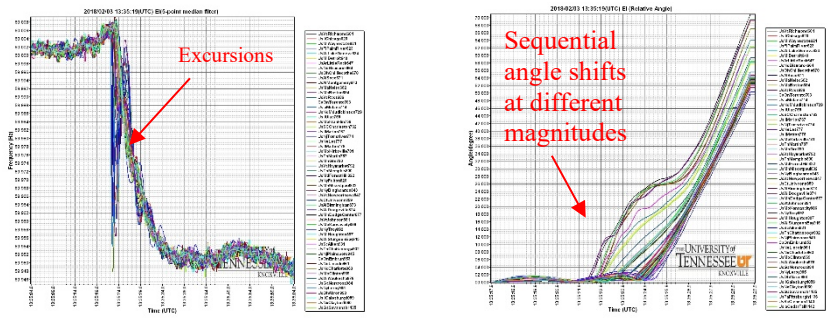
where f_{syn} is the frequency of the synchronous reference.

From (5), theoretically, the change of the magnitude of the angle shift is decided by the frequency difference of the current generator and the synchronously rotating reference. During an event, since the frequency differences vary among the grid, the resulted angle shifts will vary as well. Large angle shifts are usually observed near the event source, since the frequency differences are large. Similarly, smaller angle shifts are observed far from

the event source, since the frequency differences are small.

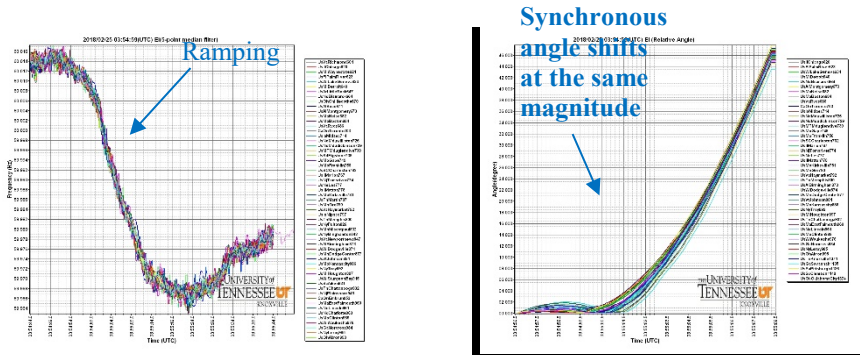
Take a generator trip and a frequency ramping. When a generator trips offline due to a fault, it creates an instant mismatch between the source and the load. To compensate the mismatch, other generators immediately consume their kinetic energy, thus their speeds are slowed down. Because the generators are affected sequentially, the angle shifts appear sequentially as well. However, a frequency ramping is usually caused by scheduled control changes in an interconnected power grid. Since a scheduled change is a planned reduction or increase of the generator output power, no fault is involved, thus the angle shifts are synchronous. A generator event and a frequency ramping captured by the FNET/GridEye are shown in Figure 6-1 and Figure 6-2 respectively. In Figure 6-1 (a), frequency excursions are observed before the system frequency drops, which create large positive ROCOF signals. In Figure 6-1 (b), angle shifts happen sequentially at different magnitudes, creating obvious RAS signals. In Figure 6-2 (a), frequency drops are seen even though no frequency excursions are observed, which creates similar ROCOF signals. However, in Figure 6-2 (b), angle shifts happen simultaneously at similar magnitudes, where no obvious RAS signals are created.

Load disconnection events have similar characteristics. Generally, the disconnection of loads causes similar RAS signals and negative ROCOF signals. This paper summarizes the characteristics of all conditions in Table 6-1.



(a) Frequency (b) Unwrapped relative angle

Figure 6-1 Frequency Disturbance



(a) Frequency (b) Unwrapped relative angle

Figure 6-2 Frequency Ramping

Table 6-1 Frequency Disturbance Characteristics

Type	***ROCOF***	***ROCOF Sign***	***RAS***
Generator Trips	√	+	√
Ramping down	√	+	
Load Disconnection	√	-	√
Ramping up	√	-	

6.3 Proposed Convolutional Neural Network Model

This paper designs two CNN models based on the typical CNN components [75]. In the following sections this paper explains the construction of the input images, the design of the layers, the choices of the parameters, and the logic of decision.

CNN Input Characteristics

Data preparation or feature engineering is the key to most, if not all, of the machine learning models [76]. In order to prepare informative inputs for a CNN, two things need to be considered:

Spatial information: The first and most important thing to consider is the spatial information. The CNN is efficient on image recognition because it can generalize multiple spatial features. To exploit this ability, the input must have enough spatial information.

Channel: Channel is important since the image recognition class usually incorporates color as an important feature. However, multiple channels may not help when spatial information is much more important than colors.

Gray Image

In image recognition, the spatial information can be independent to the color channels. For example, as seen from Figure 6-1, the spatial information of the RAS signal is more obviously reflected by the magnitude of the shifts instead of its color. Therefore, gray-scale images can be yet another representation of the spatial information. This paper uses the following function [77] to generate the gray-scale images:

$$gd_{t(k)} = 0.3 \times r_{t(k)} + 0.59 \times g_{t(k)} + 0.11 \times b_{t(k)} \tag{6}$$

where $gd_{t(k)}$ is the gray degree as of timestamp k.

ROCOF Image Construction

A ROCOF calculation is written as follows

$$ROCOF_{t(k)} = \frac{F_{t(k-\tau+1)} - F_{t(k)}}{t(k-\tau+1) - t(k)} \tag{7}$$

where $ROCOF_{t(k)}$ is the ROCOF value at timestamp k, $F_{t(k)}$ is the frequency at timestamp k, and τ is the time interval.

The ROCOF is time-series data, with temporal characteristics enclosed. However, as CNN features spatial characteristics extraction, the temporal characteristics can be transformed to a spatial representation. Here, the time-series ROCOF is transferred to a matrix **P** as

$$\boldsymbol{P} = \begin{bmatrix} P_{t(1)} & \cdots & P_{t(k)} & \cdots & P_{t(m)} \\ \vdots & & \ddots & & \vdots \\ P_{t(n-m+1)} & & \cdots & & P_{t(n)} \end{bmatrix} \tag{8}$$

where $P_{t(k)}$ is the converted pixel value at timestamp k, m is the number of points/timestamps at each row, n is the total number of points/timestamps in the image.

The ROCOF matrix is converted to a pixel image by

$$P_{t(k)} = \begin{bmatrix} r_{t(k)} \\ g_{t(k)} \\ b_{t(k)} \end{bmatrix} = \begin{bmatrix} \dfrac{ROCOF_{t(k)} - ROCOF_{\min}}{ROCOF_{\max} - ROCOF_{\min}} \times 255 \\ 0 \\ \dfrac{ROCOF_{\max} - ROCOF_{t(k)}}{ROCOF_{\max} - ROCOF_{\min}} \times 255 \end{bmatrix} \tag{9}$$

where, $r_{t(k)}$, $g_{t(k)}$, and $b_{t(k)}$ are the red, green, and blue strength at the timestamp k respectively. $ROCOF_{\max}$ and $ROCOF_{\min}$ are the maximum and the minimum ROCOF from $t(1)$ to $t(n)$ respectively. Here, a static value 0 is assigned to the green (g) position. The purpose is to make the 2 types of image differentiable. However, other values are possible as long as they can make the images differentiable for the CNN.

Some sample ROCOF images are shown in Figure 6-3. The images are of 20×20 size which contains 40-second frequency measurements. Note this paper uses 40-second data because the frequency excursion normally lasts for less than 10 seconds. This paper includes the 15-second pre-event and post-event data to construct better input images. From Figure 6-3, each event type has a unique spatial footprint that differentiates it from others. A generation trip has a blue belt while a load disconnection has a red belt due to the frequency dip and frequency spike. Similarly, looking at the gray-scale images, a generation trip has a gray belt in a black background while a load disconnection has a black belt in a gray background. It is seen from the constructed images, in both formats, the temporal information (time-series data) is preserved spatially (image).

RAS Image Construction

As seen in Figure 6-1 (b), angle shifts have rather obvious spatial differences. This paper calculates, aligns, and plots the relative angle shifts as 2D images. A relative angle is written as follows:

$$As_{t(k)}^{i} = VA_{t(k)}^{i} - VA_{t(k)}^{syn} \tag{10}$$

where Asit(k) is the angle shift of the ith device at timestamp k, VAit(k) is the voltage angle of the ith device at timestamp k, and VAsynt(k): the voltage angle of a synchronously rotating motor.

Figure 6-4 shows some sample RAS images of 100×100 size. As seen from Figure 6-4 (a), sequential RASs are very obvious when an event is involved, while from Figure 6-4 (c), synchronous RASs are observed when no event is involved. Similarly, the conclusion

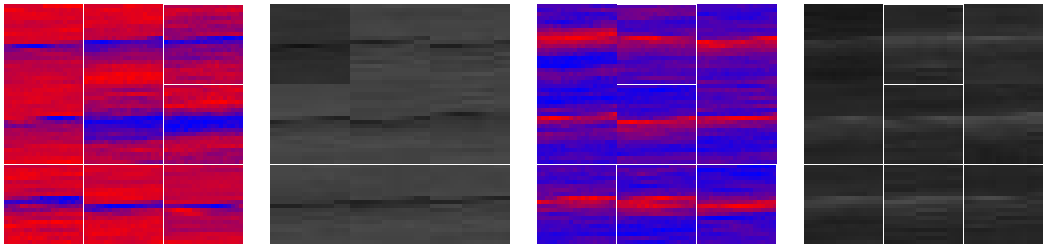
holds true for gray-scale images. Since the RAS signals contain obvious spatial characteristics, their gray-scale images reflect strong spatial characteristics as well.

Layers

A CNN mimics the structure of the brain visual cortex. A typical CNN consists of multiple stages. Usually, each stage consists of many layers, which are the convolution layer, the activation layer, the max-pooling layer, and the drop-out layer (optional). However, the last stage is usually composed of fully connected layers for classification purpose. Here, the function of each layer is explained as follows:

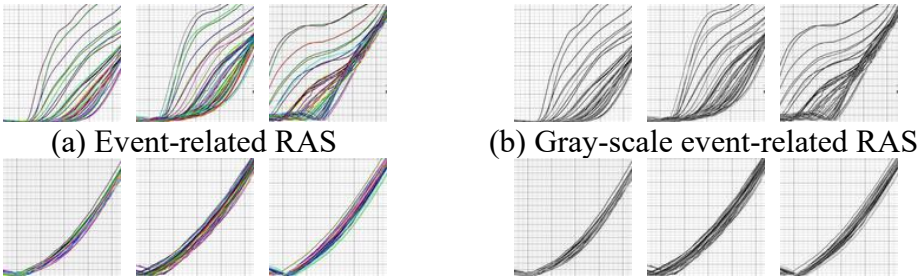
Convolution Layer: A convolution layer takes n_1 2D feature maps of size $n_2 \times n_3$ as the input. It transfers the input feature maps to m_1 2D feature maps of size $m_2 \times m_3$ using m_1 trainable kernels of size $l_1 \times l_2$. Each kernel detects a particular spatial character at every location on the input. Based on the sizes of the input images, this paper places 4 sequential convolution layers to reduce the size of the images gradually. Note 4 is specially tuned for the constructed images in this paper. On other occasion where the size of the picture is larger, more convolutional layers may be needed.

Activation Layer: An activation layer takes the 2D feature maps of size $m_2 \times m_3$, which are generated by the last convolution layer as the input. Traditionally, an activation layer is assigned a non-linear activation function. The purpose of the activation layer is to add non-linearity to the model, making it capable to compute any function. Some popular non-linear activations functions include sigmoid, tanh [78] and rectified linear unit (ReLU) [79]. Since ReLU is the most widely used activation function since negative inputs to the neuron are ignored. This paper uses ReLU as the activation function.



(a) GT color (b) GT gray-scale (c) LD color (d) LD gray-scale

Figure 6-3 Sample images



(a) Event-related RAS (b) Gray-scale event-related RAS

(c) Non-event RAS (d) Gray-scale non-event RAS

Figure 6-4 Sample RAS images

Max-pooling Layer: A pooling layer is to merge semantically similar features into one, thus it reduces the dimension of the feature maps. A max-pooling unit computes the maximum of a local path of units in one feature map (or in a few feature maps). This paper follows the classic design of the max-pooling layer [80]. It applies multiple max-pooling layers where the size of an image is reduced to a half.

Drop-out Layer (optional)*:* A drop-out layer [81] is applied at the end of each stage to prevent the CNN from overfitting. The drop-out layer ignores part of the neurons, thus it reduces the total feedback during the back propagation. This paper applies 2 dropout layers in each CNN model to reduce the speed of the learning process. Note, in this paper, 2 dropout layers are chosen based on the 4 convolution layers. If more convolution layers are introduced, more dropout layers may be needed.

Parameters

CNN parameters [75] are important values that decide its performance. CNN parameters include:

Kernel size: Kernel size is the most important parameter in a CNN model. Kernel size decides the first 2 dimensions of the convolutional output. A proper kernel size decides the spatial generalization capability of the current convolution layer. A proper kernel size typically depends on the size of the input.

Feature map number: Feature map number is another important parameter. It decides the last dimension of the convolutional output. A proper feature map number decides the number of local features to extract. A proper feature map number typically depends on the complexity of the input.

Stride: Stride is the number of pixels that the model skips between operations. For example, in a convolution layer, if stride=*n*, the kernel moves rightward by *n* pixels when the current convolution is done.

Zero-Padding: Padding is usually used to keep the input and output the same dimension. Padding improves the overall performance of a CNN by keeping information at the borders.

Input neuron number (fully-connected layer): Typically, the input neuron number matches the number of the flattened outputs of the former layer.

Output neuron number (fully-connected layer): The output neuron number should be carefully chosen. A proper number of output neurons preserves the information from the former layer yet makes the model efficient to train. Note the number of neurons at the last layer typically matches the number of image categories.

Figure 6-5 illustrates the structure of the CNN models, while Table 6-2 and Table 6-3 demonstrate the structure of the two CNN models that are used by this paper respectively, where ***k*** is the kernel size, ***f*** is the number of the feature maps, ***s*** is the stride, ***p*** is the zero-padding, ***in*** is the input neuron number, and ***out*** is the output neuron number.

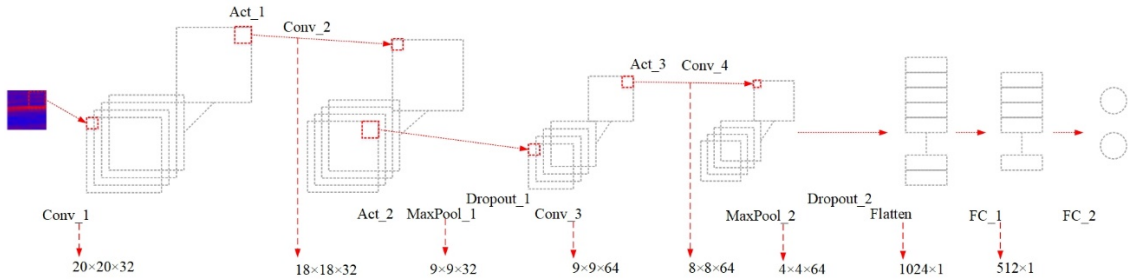
The output size is decided by the following function [76].

$$M_{out} = \frac{M_{in} - k + 2p}{s} + 1 \tag{11}$$

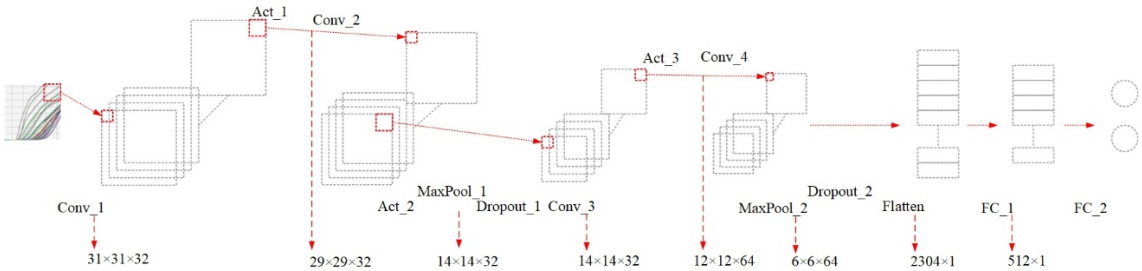
where M_{in} is the width/height of the input, and M_{out} is the width/height of the output.

Classifier fusion

A classifier fusion decides the final detection result by combining the results from multiple models. In this paper, the detection result relies on the outputs of the ROCOF_Net and the RAS_Net.



(a) ROCOF_Net



(b) RAS_Net

Figure 6-5 Proposed CNN Models

Table 6-2 CNN Structure for The ROCOF Data

Input	Operation	Output
20×20×3	Conv_1(k=2,f=32,s=1,p=1)	20×20×32
20×20×3	Act_1	20×20×32
18×18×32	Conv_2(k=3,f=32,s=1,p=0)	18×18×32
18×18×32	Act_2	18×18×32
18×18×32	MaxPool_1(k=2,s=2,p=0)	9×9×32
9×9×32	Dropout_1	9×9×32
9×9×32	Conv_3(k=2,f=64,s=1,p=1)	9×9×64
9×9×64	Act_3	9×9×64
9×9×64	Conv_4(k=2,f=64,s=1,p=0)	8×8×64
8×8×64	Act_4	8×8×64
8×8×64	MaxPool_2(k=2,s=2,p=0)	4×4×64
4×4×64	Dropout_2	4×4×64
4×4×64	Flatten	1024×1
1024×1	FC_1(in=1024, out=512)	512×1
512×1	Act_5	512×1
512×1	Dropout_3	512×1
512×1	FC_2(in=512, out=4)	2×1
4×1	Act_6	2×1

Table 6-3 CNN Structure for The RAS Data

Input	Operation	Output
100×100×3	Conv_1(k=10,f=32,s=3,p=0)	31×31×32
31×31×32	Act_1	31×31×32
31×31×32	Conv_2(k=3,f=32,s=1,p=0)	29×29×32
29×29×32	Act_2	29×29×32
29×29×32	MaxPool_1(k=2,s=2,p=0)	14×14×32
14×14×32	Dropout_1	14×14×32
14×14×32	Conv_3(k=3,f=64,s=1,p=1)	14×14×64
14×14×64	Act_3	14×14×64
14×14×64	Conv_4(k=3,f=64,s=1,p=0)	12×12×64
12×12×64	Act_4	12×12×64
12×12×64	MaxPool_2(k=2,s=2,p=0)	6×6×64
6×6×64	Dropout_2	6×6×64
6×6×64	Flatten	2304×1
2304×1	FC_1(in=2304, out=512)	512×1
512×1	Act_5	512×1
512×1	Dropout_3	512×1
512×1	FC_2(in=512, out=2)	2×1
2×1	Act_6	2×1

This paper uses the output from the ROCOF_Net to decide the possible event type of a ROCOF image. Then, it uses the output from the RAS_Net to rule out the falsely detected events. A lookup table is created to reflect the determination strategy as is shown in Table IV. Note for the RAS_Net output, 0 denotes the RAS signal that suggests an event, while 1 denotes otherwise. Therefore, events with confirmation on the RAS signal are classified as real events, otherwise they are classified as other conditions.

Figure 6-6 shows a diagram of the workflow of the event detection system. For data preprocessing, the system maintains an n-size queue, Q, of ROCOF values, $ROCOF_{\max}$, and $ROCOF_{\min}$. At each time instant k, the system pops the first ROCOF value out, pushes the newly calculated ROCOF value into the tail of the queue, and updates the $ROCOF_{\max}$, and $ROCOF_{\min}$. Afterwards, the system converts the ROCOF values via (8) and (9) into images. Here, to convert the ROCOF values into gray-scale images, an extra processing will be done via (6). Two event detection modules are shown in Figure 6-6. The ROCOF_Net and the RAS_Net run in parallel to provide fusion result and the system enters the OUTPUT state. Afterwards, the system returns to the START state and keeps running.

6.4 Results Analysis

This paper takes the confirmed generator trip (GT), load disconnection (LD), frequency ramping down (FR_down), and frequency ramping up (FR_up) events recorded by the FNET/GridEye system (Liu, et al., 2017) from January to December, 2018 to validate the proposed model. The histogram of the cases is shown in Table 6-5.

Table 6-4 Classifier Fusion Lookup Table

ROCOF_Net	RAS_Net	Decision
0	0	GT
1	0	LD
0	1	FR_down
1	1	FR_up

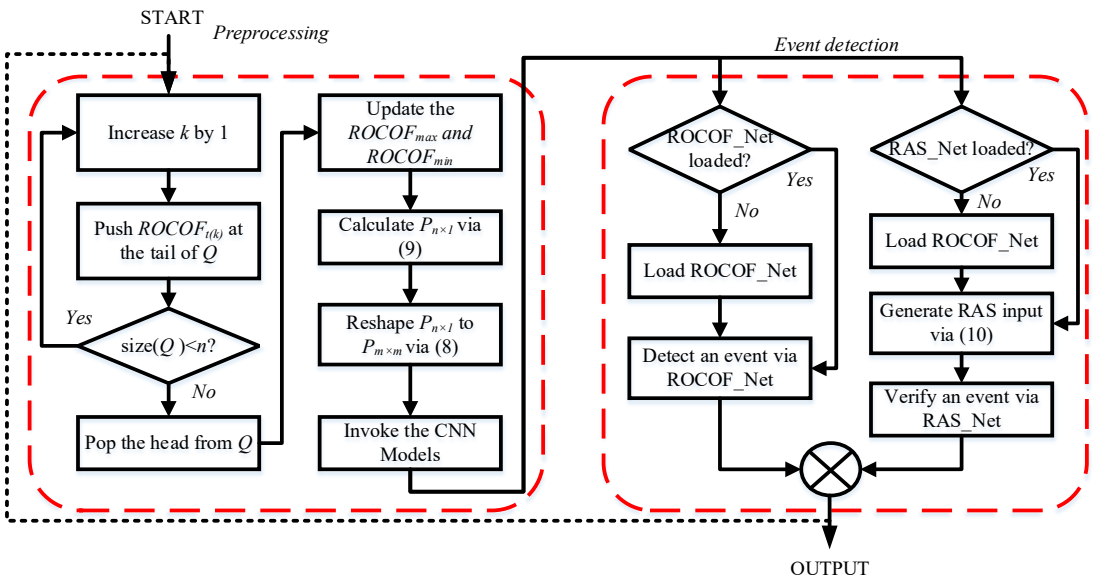


Figure 6-6 Deep learning-based event detection system diagram

For generality, this paper uses 5-fold cross validation to evaluate the performance of the model instead of splitting the data into fixed training and testing sets. First, the data is split into 5 pieces. Then, in each evaluation, this paper selects an unused piece as the validation set and packs the rest as the training set. Then, it reinitializes the model, trains and validates it using the current training and validation sets, until all pieces are used once as the validation set. In the training, this paper uses categorical loss as the loss function and stochastic gradient descent (SGD) as the update function, with learning rate as 0.001, weight decay as 10^{-6}, and momentum as 0.9.

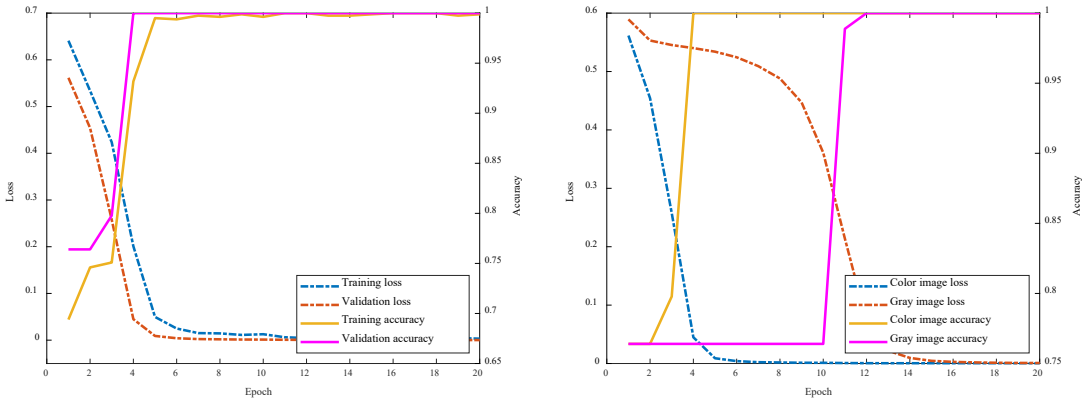
ROCOF_Net Standalone Evaluation

As is shown in Figure 6-7 (a), the training set converges at around 6th epoch, while the validation set converges at around 5th epoch. An interesting observation is the validation set always converges faster than the training set. This observation is due to the strong spatial characteristic constructed by (9). As for the accuracy, the average validation accuracy is 100%. The result proves the ROCOF_Net is efficient in classifying the two types of event.

As is shown in Figure 6-7 (b), using gray-scale images, the model converges slower than using color images. This is because the original constructed images (with RGB channels) contains more obvious spatial characteristics, while the gray-scale transformation blurs these characteristics. However, the eventual accuracy is 100% as well regardless of the slow convergence speed.

Table 6-5 Experimental Cases

Type	Region	Case#
GT	MRO	13
	NPCC	9
	RFC	55
	SPP	10
	SERC	39
	FRCC	10
LD	MRO	18
	NPCC	36
	RFC	66
	SPP	1
	SERC	84
	FRCC	3
FR_down	MRO	7
	NPCC	10
	RFC	37
	SPP	1
	SERC	33
	FRCC	2
FR_up	MRO	71
	NPCC	15
	RFC	168
	SPP	6
	SERC	150
	FRCC	32



(a) ROCOF_Net color image training (b) Comparison of color and gray images

Figure 6-7 ROCOF_Net evaluation results

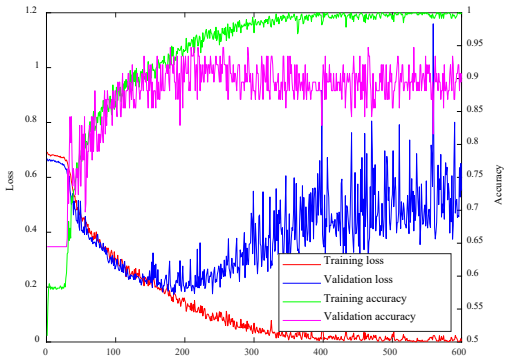
Note that even though the average validation accuracy of the ROCOF_Net is 100%, it does not mean the event detection accuracy has no error. This is because the ROCOF_Net only suggests an event type based on the constructed image.

In conclusion, both color images and gray images are good input candidates for the ROCOF_Net, although it converges faster using color images.

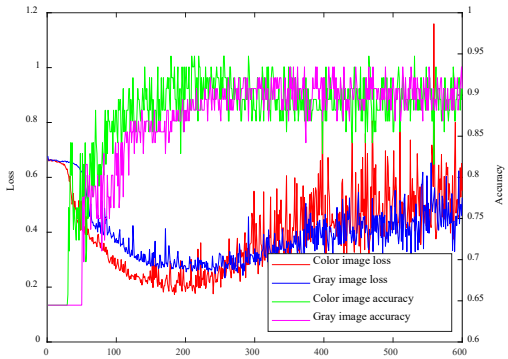
RAS_Net Standalone Evaluation

As shown in Figure 6-8 (a), for LD events, the training set converges after 400th epoch, while the validation set converges around 210th epoch, then diverges afterwards. The difference in the convergence performance is mainly caused by edge cases in LD training set and overfitting. The edge cases are those whose spatial characteristic resemble both an event and a ramping. Due to complex operational conditions and locations of disturbances, the manifestation of LD events is more complex. Therefore, some edge cases are introduced into the training set. As a result, when more edge cases are included, longer epochs are needed for the training set to converge. Since the CNN tries to generalize these edge cases, it is possible to get overfitted. In fact, Figure 6-8 (a) shows the overfitting causes the validation loss to diverge after 210th epoch. However, the rise of validation does not affect the validation accuracy much.

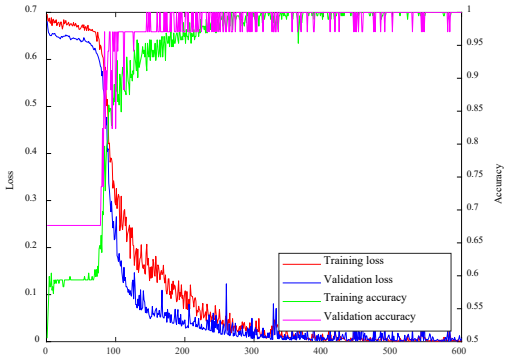
As shown in Figure 6-8 (a), the validation accuracy does not drop much even though the validation loss exceeds its initial value. Nonetheless, the overfitting issue still slightly deteriorates the performance on the validation set, as the best validation accuracy is observed at 210th epoch, where the validation loss reaches the minimum.



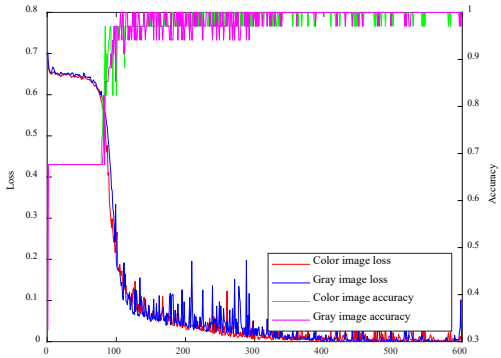
(a) RAS_Net color image LD



(b) RAS_Net comparison LD



(c) RAS_Net color image GT



(d) RAS_Net comparison GT

Figure 6-8 Evaluation Results

In conclusion, for LD events, the number of training epochs should be limited to avoid overfitting on the training set, otherwise it will detetioriate the overall performance of the proposed model.

Figure 6-8 (b) demonstrates the comparison of using color images and gray-scale images under LD events. As shown in Figure 6-8 (b), using gray-scale images, the evaluation loss becomes less chaotic than using color images. Upon convergence, the performance of using color images is slightly better than using gray-scale images. This is because the gray-scale images simplify the information from the color images, which causes some information loss. The impact of the information loss is two-fold. Firstly, it deteriorates the performance of the model upon convergence. On the other hand, it stabilizes the performance of the model. Eventually, the accuracy of using gray-scale images is similar to that of using color images. This is because the model eventually rules out the effect by the RGB channels, while it concentrates on the magnitudes of the RASs.

As seen from Figure 6-8 (c), for GT events, however, both the validation accuracy and loss are stable even if the model converges on the training set. The main reason is GT events can have clearer spatio-temporal characteristics than LD events, which avoids introducing too many edge cases. As opposed to the LD events, the validation set maintains convergence when the training set converges.

As seen from Figure 6-8 (d), using gray-scale images under GT events does not help much on improving the performance of the model. The model achieves similarly good performance using both images. However, a more stable validation loss is still observed when using gray-scale images.

Fusion Evaluation

In this paper, apart from accuracy, precision and recall are used to evaluate the fusion result. The definitions of these criteria are explained below:

Precision: the percentage of the real events out of the total events reported by the model, which is calculated by

$$pr = \frac{TruePositives}{TruePositives+FalsePositives} \tag{12}$$

Recall: the percentage of the real events out of the ground truth events, which is calculated by

$$rc = \frac{TruePositives}{TruePositives+FalseNegatives} \tag{13}$$

As seen from (12) and (13), the precision and the recall represent the false positive exclusion ability and the true positive inclusion ability respectively. Specifically, the *precision* represents how good the proposed model is at excluding false alarms. Other the other hand, *recall* represents how good the proposed model is at not missing true events.

This paper compares the performance of the ROCOF method [57], the frequency-only CNN [66], and the proposed model. It compares the average true positives, false positives, true negatives, and false negatives. Finally, as CNN is known to work well on large-volume datasets, the fusion evaluation considers a different amount of training data.

As seen from Table 6-6, on all conditions, the ROCOF method and the CNN (Frequency) model both achieve 100% in recall, which means they do not miss any events. Their superior recalls are brought by their inability to differentiate frequency rampings from disturbances.

Table 6-6 Precision, Recall, and Accuracy Comparison

Training Data	Perf	ROCOF		CNN (Frequency)		Proposed Model (Color)		Proposed Model (Gray)	
		GT	LD	GT	LD	GT	LD	GT	LD
10%	Prec.	0.67	0.40	0.67	0.40	0.91	0.77	0.91	0.77
	Rec	1	1	1	1	0.91	0.94	0.91	0.94
	Acc	0.67	0.40	0.67	0.40	0.88	0.77	0.88	0.77
	Cnv.	*N/A*	*N/A*	10~20	10~20	1300+	700+	1200+	800+
25%	Prec.	0.45	0.48	0.45	0.48	1	0.85	1	0.84
	Rec.	1	1	1	1	1	0.900	1	0.94
	Acc.	0.45	0.48	0.45	0.48	1	0.83	1	0.84
	Cov.	*N/A*	*N/A*	10~20	10~20	600~700	250~300	600~700	250~300
50%	Prec.	0.58	0.51	0.58	0.51	1	0.94	1	0.90
	Rec.	1	1	1	1	1	0.90	1	0.90
	Acc.	0.58	0.51	0.58	0.51	1	0.895	1	0.87
	Cnv.	*N/A*	*N/A*	10~20	10~20	400~600	200~250	400~600	200~250
75%	Prec.	0.49	0.53	0.49	0.53	1	0.91	1	0.94
	Rec.	1	1	1	1	1	0.86	1	0.90
	Acc.	0.49	0.53	0.49	0.53	1	0.855	1	0.895
	Cnv.	*N/A*	*N/A*	10~20	10~20	200~300	200~250	200~300	200~250
100%	Prec.	0.51	0.59	0.51	0.59	1	0.98	1	0.98
	Rec.	1	1	1	1	1	0.94	1	0.96
	Acc.	0.51	0.59	0.51	0.59	1	0.94	1	0.94
	Cnv.	*N/A*	*N/A*	10~20	10~20	150~200	180~220	150~200	180~220

This disadvantage deteriorates their precisions since many ramping cases are reported as disturbances. As seen from the figure, the precisions of these 2 methods are between 48.95% and 66.67% for GT events, and between 39.66% to 59.40% for LD events. For these two methods, their accuracy performances are only impaired by the false positive (false alarms). Therefore, their accuracies equal to their precisions for all cases. The evaluation results suggest that the false alarm challenges the credibility of both algorithms.

As opposed to them, the proposed model achieves superior performance on all criteria for both event types. As seen from Table 6-6, using 100% training data, the proposed model achieves 100% accuracy on GT events, and 94.42% accuracy on LD events. The superior performance on GT events is due to their clear spatio-temporal characteristics. However, for LD events, the accuracy and precision are slightly slower. This is because some LD events can have ambiguous spatio-temporal characteristics, which can resemble those of frequency ramping. The ambiguous spatio-temporal characteristics are mainly observed in the LD events captured under low load conditions. These LD events usually occur in early mornings [82], when the overall load is low. Under such conditions, even a small-magnitude LD event can cause the system frequency to drop drastically, which triggers the ROCOF_Net. However, its RAS signals are usually nonobvious because of its small magnitude. Therefore, these small-magnitude LD events can be recognized as frequency rampings, which deteriorate the recall rate (more false negatives). However, in practice, since their magnitudes are usually small, these false negative cases usually have little impact to the bulk power system reliability. In fact, they are too trivial to be considered in

balancing and frequency control regulations. Therefore, the performance of the proposed model is still acceptable.

Effect of training data amount

Since the performance of a deep learning model can rely on the number of training data, this paper uses different training set volumes in the evaluation as well. Figure 6-9 (a) and (b) demonstrate the impact of the number of training data on precision, recall, and accuracy.

As seen in Figure 6-9 (a), the number of training data has some but limited impact on GT events. This is because the spatio-temporal characteristics of GT events are generally clearer. As aforementioned, clearer spatio-temporal characteristics help reduce the number of edge cases. This makes the performance on the GT events stable and superior. However, as seen in Figure 6-9 (a), the performance of the model can still be deteriorated if the number of training samples is too small.

As opposed to GT events, Figure 6-9 (b) suggests that the number of training data has greater impact on LD events. This is because the spatio-temporal characteristics of LD events are more ambiguous, which results in more edge cases. However, to avoid false negative rate (missing event), the edge cases are labeled as true events in the training set. Therefore, when the number of training samples is smaller, the percentage of the edge cases can become larger. They lead to lower precision and accuracy, while keeping comparable recall. With the increase of the training set, the impact from edge cases becomes trivial. Therefore, the precision and accuracy increase with the training data size.

On the other hand, using smaller-size training data slows down the convergence speed on all cases. As Figure 6-9 (c) and (d) show, less training data results in more epochs for

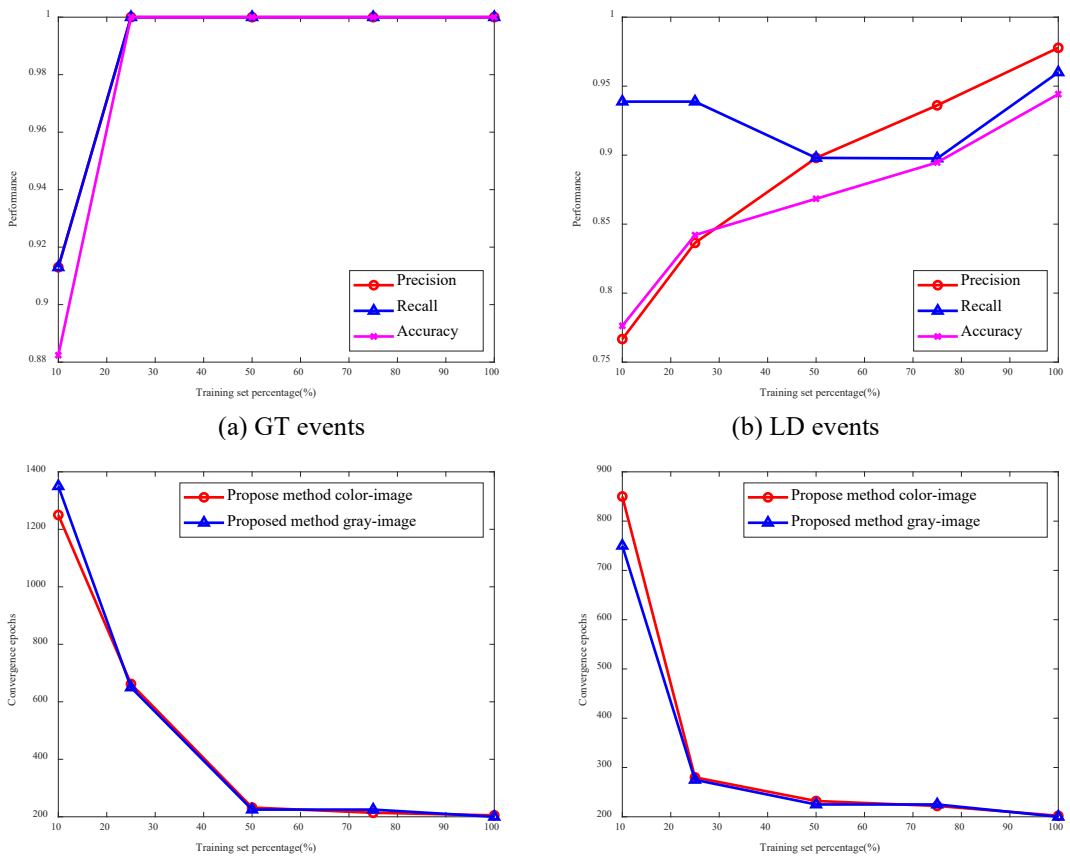
the proposed model to converge. This is because the proposed model needs more epochs to generalize the spatio-temporal features when the number of training samples is limited. However, this effect can be trivial since the training of the proposed model can be implemented offline.

Detection efficiency

This paper uses two NVIDIA GeForce GTX 1080 Ti GPU units as the hardware, TensorFlow 1.12.0 as the deep learning platform, and Windows 10 pro as the operating system. The proposed model is implemented as a micro-service via Flask [87] on an in-domain remote server.

The comparison of the detection time is demonstrated in Table 6-7. In Table 6-7, Detection time A represents the time overheads including data caching, network communication, etc. Meanwhile, detection time B represents the actual time of the algorithm logic. Total detection time represents the difference between the time when the event is detected and the time when the event emerges. The total detection time is the sum of the detection time A and the detection time B. Note, in this paper, 2 GPU units are used in parallel for the ROCOF_Net and the RAS_Net respectively. Furthermore, the initialization time of the GPU units is ignored since they only need to be initialized once when the model starts up.

As seen from Table 6-7, the ROCOF method is the fastest algorithm to detect disturbance events overall. The detection time A of the ROCOF method mainly comes from data caching.



(a) GT events (b) LD events

(c) GT events (d) LD events

Figure 6-9 Learning curve

Table 6-7 Detection Time Comparison

	ROCOF Method	CNN (Frequency)	Proposed Model (Color)	Proposed Model (Gray-scale)
Detection time A (ms)	2230	2346	2356	2356
Detection time B (ms)	100	113	120	115
Total detection time (ms)	2330	2459	2476	2471

The detection time of the ROCOF method is trivial, which indicates that it concludes the detection result whenever the data collection is done. On the other hand, the proposed model needs slightly more detection time A since it communicates with the remote server to transmit the data and generate the input. For detection time B, the proposed method is comparable to the ROCOF method. Finally, the total detection time of the proposed model is slightly longer than the ROCOF based method. However, this difference can be ignored in terms of real-time monitoring.

Therefore, the average running time of the proposed model could be controlled in micro-second level given the GPU keeps running. This makes the model suitable for some real-time control problems [83].

6.5 Discussion

Synchrophasor data in steady state conditions

Since the proposed model fully depends on synchrophasor measurements, the requirements of their data are discussed. This paper uses the frequency disturbance recorders (FDRs), whose reporting rate is 10Hz, with TVE less than 0.14% at steady state [84]. Many commercial PMUs are manufactured with higher reporting rates and lower TVEs. For example, an arbiter 1133A PMU can have 1-60 Hz reporting rate, with TVE less than 0.1% [85]. Some PMUs are proved to have < 0.1% on input magnitude and less than 0.0365% on phase angle (deg.) with 50Hz reporting rate [86]. Therefore, the proposed method may work with any PMUs which have report rate larger than 10Hz and TVE less than 0.14%. However, due to different operational conditions, the requirement of the data

resolution may change. In this paper, the data from the U.S. eastern interconnection was utilized to evaluate the model. However, for small systems such as ERCOT and QUEBEC, higher-resolution data will be necessary since the propagation of the traveling wave can be quicker. Moreover, since the steady state performance of commercial PMUs can vary from one manufacturer to another, some parameters in the proposed model, including the window size in image generation and hyper parameters in CNN, need adjustment as well.

Synchrophasor data in dynamic conditions

Unlike steady state, the actual dynamic performance of PMUs can vary greatly from one manufacturer to another. Practical issues including filter-related timestamp shift, GPS-induced time inconsistency, and in-consistent frequency measurements are common in commercial PMUs [89]. For the ROCOF data, PMUs from different manufacturers can report different values due to the difference in the estimation algorithm [89]. This difference can affect the total detection time of the proposed model. On the other hand, the RAS data can be affected by time shifts[89]. Since this paper considers measuring units from the same manufacturer, the ROCOF calculations are considered consistent. However, the time shift issue is common in the angle data, which can lead to inaccurate RAS calculations. Therefore, the timestamp shifts are detected and offset by calibrators [90] before the image generation. This is a required step in order to make the proposed model work, otherwise the time shift can cause chaotic RAS signals.

Training data

As discussed in Section IV, to achieve best detection accuracy, the proposed model needs relatively large training data to generalize the dynamic characteristics under various operational conditions. A potential limitation of the proposed model is it does not directly include operational conditions, such as real power, reactive power, system inertia, percentage of renewables, etc., as inputs. Operational conditions can impact the actual ROCOF and RAS signals during disturbances. For example, due to higher and higher penetration of distributed energy resources (DERs), the actual ROCOF and RAS manifestations under the DER-related disturbances can be different from conventional ones [88]. In fact, the performance of the proposed model can be further improved by incorporating these operational conditions as inputs. The proposed model can be initialized as multiple instances, where each instance is trained via data under similar operational conditions. Decision algorithms such as decision tree can be used to select appropriate instance for specific operational condition to get a more accurate result.

Implementation

The proposed model can be implemented in both online and offline modes. For online implementation, since the proposed model is implemented in TensorFlow 1.12, it can be integrated into the existing synchrophasor system via a socket-based software daemon or micro-service technology such as Flask [87]. Figure 6-10 demonstrates an example online implementation of the proposed model in FNET/GridEye. The proposed model is hosted via Flask as a microservice. It provides a likelihood estimation for the event report. Under the “Likelihood Estimation” section, it reports “Disturbance” or “Ramping” to help

operators understand if the detected event is a real disturbance. For offline implementation, the proposed model can be invoked periodically to classify the historical events. In conclusion, the proposed model can provide online detection results to assist real-time situation awareness as well as offline event analysis. However, since these implementation methods are based on distributed systems, they require a high-speed, secure, and reliable network transmission pipeline. The maintenance and troubleshooting of such a complex system can be challenging in practice.

6.6 Conclusion

This paper proposed a deep learning-based power system frequency event detection model. This paper shared important knowledge on building image inputs and CNN to detect the frequency disturbance events. This paper utilized the verified event data from the U.S. eastern interconnection to prove its feasibility in event detection. The exciting results presented in this paper allow power companies to foresee such model becoming a fundamental tool for situation awareness in bulk electric power system. The proposed model is of great significance in terms of helping the system operators know the condition of the system and decide the next necessary operations.

Limited by the confirmed event cases, this paper only collected the generator trips and the load disconnections from the power industry. Events including line trips and inter-area oscillations also have obvious ROCOF and RAS signals. In the future, more research will be conducted on including other disturbance events.

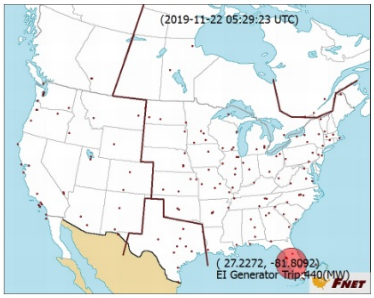
FNET Event Report

Basic Event Information

Event Date	Event Time	Event Type	Estimated Amount
2019-11-22	05:29:23 UTC	Generation Trip	440 MW
Point A	Point B	Point C	Point C Prime
60.0353 Hz	60.0196 Hz	60.0173 Hz	N/A Hz
MOD-027-1 Event	Inter Connection	Estimated Reliability Coordinator	Likelihood Estimation
NO	EI	FRCC	Disturbance
Estimated Event Location		Additional Location Information	
(27.2272, -81.8092)		near DeSoto County Plant power plant (FRCC) in (Arcadia,FL,34266).	

*Due to limited knowledge on WECC and ERCOT, the magnitude estimation may not be accurate. Please verify it before use.

Location Map



Frequency Plot of All FDRs

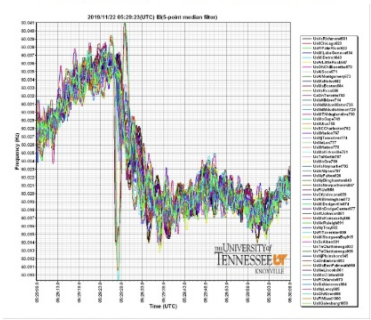


Figure 6-10 Real-time disturbance alarm

LIST OF REFERENCES

[1] A.G. Phadke and T. Bi, “Phasor measurement units, WAMS, and their applications in protection and control of power systems,” in Journal of Modern Power Systems and Clean Energy, vol. 6, no. 4, pp.619-629, July 2018.

[2] R. D. Quint, “Synchrophasors for Situational Awareness,” North American Electric Reliability Corporation, Atlanta, GA, USA, 2015.

[3] Y. Liu et al., "Wide-Area-Measurement System Development at the Distribution Level: An FNET/GridEye Example," in IEEE Transactions on Power Delivery, vol. 31, no. 2, pp. 721-731, April 2016.

[4] H. Yin, W. Yu, A. Bhandari, W. Yao and L. Zhan, "Advanced Universal Grid Analyzer Development and Implementation," 2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA), College Station, TX, USA, 2019, pp. 1-5.

[5] A. von Meier, E. Stewart, A. McEachern, M. Andersen, and L. Mehrmanesh, "Precision Micro-Synchrophasors for Distribution Systems: A Summary of Applications," in IEEE Transactions on Smart Grid, vol. 8, no. 6, pp. 2926-2936, Nov. 2017.

[6] IEEE Standard for Phasor Data Concentrators for Power Systems," in IEEE Std C37.247-2019 , vol., no., pp.1-44, 10 Sept. 2019.

[7] W. Wang, et al., “Information and Communication Infrastructures in Modern Wide-Area Systems,” in *Wide-area Power Systems Stability, Protection, and Security*. H. H. Alhelou, Berlin, Germany: Springer Nature, 2020.

[8] A. Armenia and J. H. Chow, "A Flexible Phasor Data Concentrator Design Leveraging Existing Software Technologies," in *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 73-81, June 2010.

[9] R. Pourramezan, Y. Seyedi, H. Karimi, G. Zhu and M. Mont-Briant, "Design of an Advanced Phasor Data Concentrator for Monitoring of Distributed Energy Resources in Smart Microgrids," in IEEE Transactions on Industrial Informatics, vol. 13, no. 6, pp. 3027-3036, Dec. 2017.

[10] Electric Power Group. "ePDC-enhanced Phasor Data Concentrator," electricpowergroup.com. http://www.electricpowergroup.com/epdc.html (accessed June. 7, 2020)

[11] Pingthings. "PingThings PredictiveGrid™- The platform for data science and AI on petabyte scale sensor data," www.pingthings.io. https://www.pingthings.io/platform.html (accessed June. 7, 2020)

[12] Grid Protection Alliance. "Grid Protection Alliance: Products", gridprotectionalliance.org. https://www.gridprotectionalliance.org/products.asp (accessed June. 7, 2020)

[13] W. Wang, C. Chen, w. yao, K. Sun, W. Qiu and Y. liu, "Synchrophasor Data Compression Under Disturbance Conditions via Cross-Entropy-Based Singular Value Decomposition," IEEE Transactions on Industrial Informatics (Early Access).

[14] Influxdata. "Propose-Built Open Source Time Series Database", influxdata.com. https://www.influxdata.com (accessed June 13, 2020)

[15] J. Zhao, G. Zhang, M. L. Scala, et al, "Enhanced Robustness of State Estimator to Bad Data Processing Through Multi-innovation Analysis," IEEE Trans. Ind. Informat., vol. 13, no. 4, pp. 1610-1619, 2017.

[16] Z. Lin, F. Wen, Y. Ding, et al, "Data-Driven Coherency Identification for Generators Based on Spectral Clustering," IEEE Trans. Ind. Informat., vol. 14, no. 3, pp. 1275-1285, 2018.

[17] W. Yu, W. Yao, X. Deng, et al, "Timestamp Shift Detection for Synchrophasor Data Based on Similarity Analysis between Relative Phase Angle and Frequency," IEEE Trans. Power Del (Early Access).

[18] W. Wang, J. Zhao, W. Yu, et al, "FNETVision: A WAMS Big Data Knowledge Discovery System," IEEE PESGM, pp. 1-5, Portland, USA, 2018.

[19] W. Yao, S. You, W. Wang, et al, “A Fast Load Control System Based on Mobile Distribution-Level Phasor Measurement Unit”, IEEE Trans. Smart Grid, vol. 11, no. 1, pp, 895-904, 2020.

[20] C. Chen, H. Yang, W. Wang, et al, “Harmonic Transmission Characteristics for Ultra-long Distance AC Transmission Lines Based on Frequency-Length Factor”, Electr. Pow. Syst. Res., (Early Access).

[21] C. Chen, X. Ma, H. Yang, W. Wang, and Y. Liu, “Study on the Power-Frequency Waves Distribution Characteristics for Half-Wavelength Transmission Lines Based on the Frequency-Length Factor,” Mathematical Problems in Engineering, vol. 2020, pp. 1-14.

[22] N. Tong, L. Chen, W. Wang, X. Lin and Z. Li, "Local-Measurement-Based High-Speed Protection for Half-Wavelength UHV Lines," IEEE Trans. Power Del. (Early Access).

[23] K. Sun *et al*., "VSC-MTDC System Integrating Offshore Wind Farms Based Optimal Distribution Method for Financial Improvement on Wind Producers," IEEE Trans. Ind. Appl., vol. 55, no. 3, pp. 2232-2240, May-June 2019.

[24] T. Li, Y. Li, M. Liao, W. Wang and C. Zeng, “A New Wind Power Forecasting Approach Based on Conjugated Gradient Neural Network,” Mathematical Problems in Engineering, vol. 2016, pp. 1-9.

[25] U.S. Department of Energy, "Summary of the North American SynchroPhasor Initiative (NASPI) Activity Area," U.S. Department of Energy, 2012

[26] IEEE Standard C37.118.1-2011, "IEEE Standard for Synchrophasor Measurements for Power Systems," 2011.

[27] S. Das and T. Singh Sidhu, "Application of Compressive Sampling in Synchrophasor Data Communication in WAMS," IEEE Trans. Ind. Informat., vol. 10, no. 1, pp. 450-460, 2014.

[28] K. Sayood, “Introduction to Data Compression” (Elsevier, 4th edn. 2012)

[29] R. Klump, P. Agarwal, J. Tate, et al, "Lossless Compression of synchro-nized phasor measurements," IEEE PESGM, Providence, USA, 2010.

[30] P. Top and J. Breneman, "Compressing Phasor Measurement Data," North American Power Symposium (NAPS), Manhattan, USA, 2013.

[31] S. Santoso, E. Powers and W. Grady, "Power Quality Disturbance Data Compression using Wavelet Transform Methods," IEEE. Trans. Power Del., vol. 12, no. 3, pp. 1250-1257, 1997.

[32] J. Ning, J. Wang, W. Gao, et al, "A Wavelet-Based Data Compression Technique for Smart Grid," IEEE Trans. Smart Grid, vol. 2, no. 1, pp. 212-218, 2011.

[33] E. Hamid and Z. Kawasaki, "Wavelet-based Data Compression of Power System Disturbances using the Minimum Description Length Criterion," IEEE Trans. Power Del., vol. 17, no. 2, pp. 460-466, 2002.

[34] J. Khan, S. Bhuiyan, G. Murphy and J. Williams, "Data Denoising and Compression for Smart Grid Communication," IEEE. Trans. Signal Inf. Process. Netw., vol. 2, no. 2, pp. 200-214, 2016.

[35] J. Khan, S. M. A. Bhuiyan, G. Murphy and M. Arline, "Embedded-Zerotree-Wavelet-Based Data Denoising and Compression for Smart Grid," IEEE Trans. Ind. Appl., vol. 51, no. 5, pp. 4190-4200, 2015.

[36] F. Zhang, L. Cheng, X. Li, Y. Sun, W. Zhao and W. Zhao, "Application of a Real-Time Data Compression and Adapted Protocol Technique for WAMS," IEEE. Trans. Power Syst., vol. 30, no. 2, pp. 653-662, 2015.

[37] L. Xie, Y. Chen and P. Kumar, "Dimensionality Reduction of Synchrophasor Data for Early Event Detection: Linearized Analysis," IEEE Trans. Power Syst., vol. 29, no. 6, pp. 2784-2794, 2014.

[38] P. Gadde, M. Biswal, S. Brahma, et al, "Efficient Compression of PMU Data in WAMS," IEEE Tran. Smart Grid, vol. 7, no. 5, pp. 2406-2413, 2016.

[39] J. Souza, T. Assis and B. Pal, "Data Compression in Smart Distribution Systems via Singular Value Decomposition," IEEE Trans. Smart Grid, vol. 8, no. 1, pp. 275-284, 2017.

[40] North American Electric Reliability Corporation, "Balancing and Frequency Control" (North American Electric Reliability Corporation, pp. 17-40, 2011.

[41] P. T. Boer, D. Kroese, S. Mannor, et al, “A Tutorial on the Cross-Entropy Method,” Annals of Operations Research, 2005, vol. 34, no. 1, pp. 19-67.

[42] W. Wang et al., "Frequency Disturbance Event Detection Based on Synchrophasors and Deep Learning," IEEE Trans. Smart Grid (Early Access).

[43] "PSS®E--high-performance transmission planning and analysis software," [Online]. Available: https://new.siemens.com/global/en/products/energy/services/transmission-distribution-smart-grid/consulting-and-planning/pss-software/pss-e.html, [Accessed: 23-Jul-2019].

[44] X. Deng, D. Bian, D. Shi, W. Yao, L. Wu and Y. Liu, "Impact of Low Data Quality on Disturbance Triangulation Application Using High-Density PMU Measurements," IEEE Access, vol. 7, pp. 105054-105061, 2019.

[45] Y. Liu et al., "Recent developments of FNET/GridEye — A situational awareness tool for smart grid," CSEE Journal of Power and Energy Systems, vol. 2, no. 3, pp. 19-27, 2016.

[46] F. W. Bernal, J. Wold, R. Concepcion and J. Budai, "A Method for Correcting Frequency and RoCoF Estimates of Power System Signals with Phase Steps,"

2019 North American Power Symposium (NAPS), Wichita, KS, USA, 2019, pp. 1-6

[47] X. Deng, H. Li, W. Yu, et al, “Frequency Observations and Statistic Analysis of Worldwide Main Power Grids using FNET/GridEye,” IEEE PESGM, Atlanta, USA, 2019.

[48] P. Wright, P. Davis, K. Johnstone, et al, "Field Measurement of Frequency and ROCOF in the Presence of Phase Steps," IEEE Trans. Instrum. Meas., vol. 68, no. 9, pp. 1688 - 1695, 2019.

[49] W. Wang, W. Yao, C. Chen, X. Deng and Y. Liu, "Fast and Accurate Frequency Response Estimation for Large Power System Disturbances using Second Derivative of Frequency Data," IEEE Trans. Power Syst. (Early Access).

[50] “Batch Writes,” *Optimize Writes to InfluxDB*. [Online]. Available: https://v2.docs.influxdata.com/v2.0/write-data/best-practices/optimize-writes/. [Accessed: 01-Apr- 2020].

[51] North American Electric Reliability Corporation. Frequency Response Standard Background Document, 2012.

[52] W. Wang, W. Yao, C. Chen, X. Deng and Y. Liu, "Fast and Accurate Frequency Response Estimation for Large Power System Disturbances Using Second Derivative of Frequency Data," IEEE Trans. Power Syst. vol. 35, no. 3, pp. 2483-2486, May 2020.

[53] “Fuel Mix Report: 2007 – 2020,” *Generation* [Online]. Available: http://www.ercot.com/gridinfo/generation [Accessed: 22-Mar-2021].

[54] Electricity Consumers Resource Concil (ELCON), "The Economic Impacts of the August 2003 Blackout," 2004.

[55] W. Wang, J. Zhao, W. Yu and Y. Liu, "FNETVision: A WAMS Big Data Knowledge Discovery System," *2018 IEEE Power & Energy Society General Meeting (PESGM)*, Portland, OR, 2018.

[56] W. Yao, S. You, W. Wang, *et al*., "A Fast Load Control System Based on Mobile Distribution-Level Phasor Measurement Unit," *IEEE Trans. Smart Grid*, Early access.

[57] T. Xia, H. Zhang, R. Gardner, J. Bank, J. Dong, J. Zuo, Y. Liu, L. Beard, P. Hirsch, G. Zhang and R. Dong, "Wide-area Frequency Based Event Location Estimation," *2007 IEEE Power Engineering Society General Meeting*, Tampa, FL, 2007.

[58] J. E. Tate and T. J. Overbye, "Line Outage Detection Using Phasor Angle Measurements," *IEEE Trans. Power Syst.,* vol. 23, no. 4, pp. 1644-1652, 2008.

[59] S. You, J. Guo, G. Kou, Y. Liu and Y. Liu, "Oscillation Mode Indetification Based on Wide-area Ambient Measurements using Multivariate Empirical Mode Decomposition," *Electric Power System Research,* vol. 134, no. 2016, pp. 158-166, 2016.

[60] I. Kamwa, A. K. Pradhan and G. Joos, "Robust Detection and Analysis of Power System Oscillations Using the Teager-Kaiser Energy Operator," *IEEE Trans. Power Syst.,* vol. 26, no. 1, pp. 323-333, 2011.

[61] T. Bi, X. Song, J. Wu and Q. Yang, "Novel Method for Distrubance Indentification in Power Systems," *2006 IEEE Power Engineering Society General Meeting*, Montreal, Quebec,Canada, 2006.

[62] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh and G. Stefopoulos, "Missing Data Recovery by Exploiting Low-Dimensionality in Power System Synchrophasor Measurements," *IEEE Trans. Power Syst.,* vol. 31, no. 2, pp. 1006-1013, 2016.

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

[64] Y. Liu, S. You, W. Yao, Y. Cui, L. Wu, D. Zhou, J. Zhao, H. Liu and Y. Liu, "A Distribution Level Wide Area Monitoring System for the Electric Power Grid-FNET/GridEye," *IEEE Access,* vol. 5, pp. 2329-338, 2017.

[65] Y. Song, W. Wang, Z. Zhang, H. Qi and Y. Liu, "Multiple Event Detection and Recognition for Large-Scale Power Systems Through Cluster-Based Sparse Coding," *IEEE Trans. Power Syst.,* vol. 32, no. 6, pp. 4199-4210, 2017.

[66] V. Miranda, P. A. Cardoso, R. Bessa and I. Decker, "Through the looking glass: Seeing events in power systems dynamics," *International Journal of Electrical Power & Energy Systems,* vol. 106, pp. 411-419, 2019.

[67] Y. He, G. J. Mendis and J. Wei, "Real-Time Detection of False Data Injection Attacks in Smart Grid: A Deep Learning-Based Intelligent Mechanism," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2505-2516, Sept. 2017.

[68] Y. Ju, G. Sun, Q. Chen, M. Zhang, H. Zhu and M. U. Rehman, "A Model Combining Convolutional Neural Network and LightGBM Algorithm for Ultra-Short-Term Wind Power Forecasting," *IEEE Access*, vol. 7, pp. 28309-28318, 2019.

[69] S. Motepe, A. N. Hasan and R. Stopforth, "Improving Load Forecasting Process for a Power Distribution Network Using Hybrid AI and Deep Learning Algorithms," *IEEE Access*, vol. 7, pp. 82584-82598, 2019.

[70] Y. Du, F. Li, J. Li and T. Zheng, "Achieving 100x Acceleration for N-1 Contingency Screening With Uncertain Scenarios Using Deep Convolutional Neural Network," *IEEE Trans. Power Syst.,* vol. 34, no. 4, pp. 3303-3305, July 2019.

[71] K. Cai, W. Cao, L. Aarniovuori, H. Pang, Y. Lin and G. Li, "Classification of Power Quality Disturbances Using Wigner-Ville Distribution and Deep Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 119099-119109, 2019.

[72] North American Electric Reliability Corporation, "Balancing and Frequency Control," North American Electric Reliability Corporation, Princeton, NJ, 2011.

[73] P. Markham and Y. Liu, "Electromechanical speed map development using FNET/GridEye frequency measurements," in *2014 IEEE PES General Meeting | Conference & Exposition*, National Harbor, MD, 2014.

[74] P. Gupta, R. S. Bhatia and D. K. Jain, "Active ROCOF Relay for Islanding Detection," *IEEE Trans. Power Del.,* vol. 32, no. 1, pp. 420-429, 2017.

[75] I. Goodfellow, Y. Bengio and A. Courville, "Convolutional Networks," in *Deep Learning*, Cambridge, MA, USA, MIT Press, 2016, pp. 326-339.

[76] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM,* vol. 55, no. 10, pp. 78-87, 2012.

[77] Radiocommunication Sector of ITU, "Recommendation ITU-R BT.601-7," International Telecommunication Union, 2011.

[78] Y. LeCun, K. Kavkcuoglu and C. Farabet, "Convolutional networks and applications in vision," *2010 IEEE International Symposium on Circuits and Systems*, Paris, France, 2010.

[79] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *27th International Conference on International Conference on Machine Learning*, Haifa, Isreal, 2010.

[80] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proc. of The IEEE,* pp. 1-46, 1998.

[81] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[82] D. Zhou *et al*., "Distributed Data Analytics Platform for Wide-Area Synchrophasor Measurement Systems," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2397-2405, Sept. 2016.

[83] J. Undrill, “Primary Frequency Response and Control of Power System Frequency, ” LBNL, Berkely, CA, Tech. Rep. LBNL-2001105, Feb. 2018.

[84] J. A. Culliss, “A 3rd Generation Frequency Disturbance Recorder: A Secure, Low Cost Synchrophasor Measurement Device,” Ph.D. dissertation, Dept. EECS., Univ. of Tenn., Knoxville, TN, 2015.

[85] *Model 1133A Phasor Measurement Specifications*, Arbiter Systems, Paso Robles, CA, 2019, pp. 1-1.

[86] G. Y. Yang, K. E. Martin and J. Østergaard, "Investigation of PMU Performance Under TVE criterion," *2010 5th International Conference on Critical Infrastructure (CRIS)*, Beijing, 2010, pp. 1-7.

[87] Pallets. (2010). Welcome to Flask--Flask Documentation 1.1.x. Pallets [Online]. Available: https://flask.palletsprojects.com/en/1.1.x/

[88] NERC, “1200 MW Fault Induced Solar Photovoltaic Resource Interruption Disturbance Report,” NERC., Atlanta, GA, June. 2017.

[89] Z. Huang, J. F. Hauer and K. E. Martin, "Evaluation of PMU Dynamic Performance in Both Lab Environments and under Field Operating Conditions," *2007 IEEE Power Engineering Society General Meeting*, Tampa, FL, 2007, pp. 1-6.

[90] W. Yu, W. Yao, X. Deng, Y. Zhao and Y. Liu, "Timestamp Shift Detection for Synchrophasor Data Based on Similarity Analysis between Relative Phase Angle and Frequency," *IEEE Trans. Power Del*. Early Access

[91] X. Deng, H. Li, W. Yu, W. Wang, and Y. Liu, “Frequency Observations and Statistic Analysis of Worldwide Main Power Grids using FNET/GridEye”, *2019 IEEE PESGM*, Atlanta, GA, 2019, pp.1-5.

[92] W. Wang, *et al*., “Frequency Disturbance Event Detection Based on Synchrophasors and Deep Learning,” IEEE Trans. Smart Grid, in press, 2020.

[93] W. Wang, W. Yao, C. Chen, X. Deng, Y. Liu, “Fast and Accurate Frequency Response Estimation for Large Power System Disturbances using Second Derivative of Frequency Data” IEEE Trans. Power Syst., vol. 35, no. 3, pp. 2483-2486, 2020.

[94] W. Wang, J. Zhao, W. Yu, and Y. Liu, “FNETVision: A WAMS Bid Data Knowledge Discovery System,” *2018 IEEE PESGM*, Portland, OR, 2018, pp. 1-5.

[95] Y. Cui, W. Wang, P, Fuhr, M, Roderiguez, and Y. Liu, “Spatio-Temporal Synchrophasor Data Characterization for Mitigating False Data Injection in Smart Grids,” *2019 IEEE PESGM*, Atlanta, GA, 2019, pp.1-5.

[96] W. Yao, S. You, W. Wang, X. Deng, Y. Li, L. Zhan, and Y. Liu, “A Fast Load Control System Based on Mobile Distribution-Level Phasor Measurement Unit”, vol. 11, no. 1, pp. 895-904, Jan. 2020.

[97] North America SynchroPhasor Initiative. “Synchrophasor technology fact sheet,” 2014. [Online]. Available: www.naspi.org/sites/default/files/reference_documents/33.pdf?fileID=1326. [Accessed Oct. 1st, 2014]

[98] B. Yang, K. V. Katsaros, W. K. Chai and G. Pavlou, "Cost-Efficient Low Latency Communication Infrastructure for Synchrophasor Applications in Smart Grids," in *IEEE Systems Journal*, vol. PP, no. 99, pp. 01-11.

[99] A. Armenia and J. H. Chow, "A Flexible Phasor Data Concentrator Design Leveraging Existing Software Technologies," in IEEE Transactions on Smart Grid, vol. 1, no. 1, pp. 73-81, June 2010.

[100] R. Pourramezan, Y. Seyedi, H. Karimi, G. Zhu and M. Mont-Briant, "Design of an Advanced Phasor Data Concentrator for Monitoring of Distributed Energy Resources in Smart Microgrids," in *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3027-3036, Dec. 2017.

[101] A. Derviskadic, P. Romano, M. Pignati and M. Paolone, "Architecture and Experimental Validation of a Low-Latency Phasor Data Concentrator," in *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2885-2893, July 2018.

[102] M. P. Andersen, S. Kumar, C. Brooks, A. von Meier and D. E. Culler, "DISTIL: Design and implementation of a scalable synchrophasor data processing system," *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Miami, FL, 2015, pp. 271-277.

[103] M. Edwards, A. Rambani, Y. Zhu, “Design of Hadoop-based Framework for Analytics of Large Synchrophasor Datasets” in Procedia Computer Science, vol. 12, pp. 254-258, 2012.

[104] T. Li, Y. Li, M. Liao, W. Wang, and C. Zeng, “A new wind power forecasting approach based on conjugated gradient neural network” Mathematical Problems in Engineering, vol. 2016, pp. 1-8, 2016.

[105] K. Sun, K. Li, J. Pan, Y. Liu, and Y. Liu, “An Optimal Combined Operation Scheme for Pumped Storage and Hybrid Wind-Photovoltaic

Complementary Power Generation System", Applied Energy, vol. 242, pp. 1155-1163.

[106] NERC, "1200 MW Fault Induced Solar Photovoltaic Resource Interruption Disturbance Report," NERC, Atlanta, GA, June. 2017.

[107] W. Wang, *et al*., "Advanced Synchrophasor-based Application for Potential Distributed Energy Resources Management: Key Technology, Challenge and Vision," 2020 IEEE IAS I&CPS Asia, Weihai, China, 2020.

[108] W. Wang, *et al*., "Advanced Synchrophasor-based Application for Potential Distributed Energy Resources Management: Key Technology, Challenge and Vision," 2020 IEEE IAS I&CPS Asia, Weihai, China, 2020.

[109] IEEE Standard C37.118.1-2011, "IEEE Standard for Synchrophasor Measurements for Power Systems," 2011.

[110] W. Yu, W. Yao, X. Deng, Y. Zhao and Y. Liu, "Timestamp Shift Detection for Synchrophasor Data Based on Similarity Analysis between Relative Phase Angle and Frequency," *IEEE Trans. Power Del*. Early Access

[111] Oracle. "MySQL," 2020. [Online]. Available: mysql.com. [Accessed: Jan. 29, 2020]

[112] Grid Protection Alliance. "openHistorian," 2020. [Online]. Available: https://github.com/GridProtectionAlliance/openHistorian, [Accessed: Jan. 29, 2020]

[113] T. H. Cormen, C. E. Leiserson, R. L. Rivest, et al, "Introduction to Algorithms", 3rd ed., Cambridge, MA, USA: MIT Press, 2009, pp.191-212.

[114] E. O. Schweitzer, D. Whitehead, G. Zweigle, K. G. Ravikumar and G. Rzepka, "Synchrophasor-based power system protection and control applications," *2010 Modern Electric Power Systems*, Wroclaw, 2010, pp. 1-10.

VITA

Weikang Wang received his B.E. degree in computer science from North China Electric Power University in 2016. He started his Ph.D. study at the University of Tennessee, Knoxville, in August 2016, with anticipated graduation in July, 2021.

His research interests include wide-area monitoring, bulk power system awareness, machine learning, and big data analytics. He has authored one book chapter, over eight journal papers and six conference papers in IEEE Transactions on Smart Grid, IEEE Transaction on Power Systems, IEEE Transactions on Power Delivery, IEEE Transactions on Industrial Informatics, IEEE Transaction on Instrumentation and Measurements, Electric Power System Research, IET Generation, Transmission, and Distribution, IEEE Power and Energy Society General Meeting, etc.