



January 2018

# Investigation Of Multi-Criteria Clustering Techniques For Smart Grid Datasets

Mitch J. Campion

Follow this and additional works at: <https://commons.und.edu/theses>

---

## Recommended Citation

Campion, Mitch J., "Investigation Of Multi-Criteria Clustering Techniques For Smart Grid Datasets" (2018). *Theses and Dissertations*. 2182.

<https://commons.und.edu/theses/2182>

This Thesis is brought to you for free and open access by the Theses, Dissertations, and Senior Projects at UND Scholarly Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of UND Scholarly Commons. For more information, please contact [zeinebyousif@library.und.edu](mailto:zeinebyousif@library.und.edu).

INVESTIGATION OF MULTI-CRITERIA CLUSTERING TECHNIQUES FOR  
SMART GRID DATASETS

Mitch J Campion

Bachelor of Arts in Mathematics, Concordia College-Moorhead, MN

A Thesis

Submitted to the Graduate Faculty

University of North Dakota

in partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota

May

2018

© 2018 Mitch Campion

This thesis, submitted by Mitch Campion in partial fulfillment of the requirements for the Degree of Master of Science from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

---

Prakash Ranganathan, Ph.D., Chairperson

---

Hossein Salehfar, Ph.D.

---

Saleh Faruque, Ph.D.

This thesis meets the standards for appearance, conforms to the style and format requirements of the Graduate School of the University of North Dakota, and is hereby approved.

---

Grant McGimpsey  
Dean of the School of Graduate Studies

---

Date

## PERMISSION

Title: Investigation of Multi-Criteria Clustering Techniques for Smart Grid Datasets

Department: Electrical Engineering

Degree: Master of Science

In presenting this thesis in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis work or, in his absence, by the chairperson of the department or the dean of the Graduate School. It is understood that any copying or publication or other use of this thesis or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my thesis.

Mitch Campion

May 2018

## Table of Contents

PERMISSION.....	iv
LIST OF FIGURES.....	viii
LIST OF TABLES.....	x
ACKNOWLEDGEMENTS.....	11
ABBREVIATIONS.....	12
ABSTRACT.....	14
<b>Chapter 1. Introduction.....</b>	<b>16</b>
1.1 Motivation.....	16
1.2 Background.....	17
1.3 Thesis Outline.....	19
1.4 Thesis Contributions.....	20
1.5. Publications.....	21
<b>Chapter 2. Methodology.....</b>	<b>24</b>
2.1 Data Mining and Smart Grid.....	24
2.1.1 Data Clustering.....	27
2.1.2 Time-series Clustering.....	30
2.2 Clustering Algorithms.....	32
2.2.1 k-means.....	32
2.2.2 k-medians.....	32
2.2.3 Density Based Spatial Clustering for Applications with Noise (DBSCAN).....	33
2.2.4 Hierarchical Clustering (hclust).....	33
2.2.5 k-Shape Algorithm.....	35
2.2.6 Partitional DTW.....	36
2.3. Forecasting for Power Systems and Smart Grid Applications.....	36
2.3.1. Forecasting Algorithms.....	40
2.3.2. ARIMA.....	40
2.3.3. Seasonal time-series decomposition.....	41
2.4 Smart Grid Modeling Using Graph Theory.....	41
2.4.1 Degree and Eccentricity.....	44
2.4.2 Betweenness Centrality.....	44
2.5 Graph Clustering.....	45
2.5.1 Graph Cluster Modularity Index.....	46
2.5.2 Girvan-Newman (GN) Algorithm.....	47
2.5.3 Nearest Generator (NG) Clustering.....	48
2.5.4 Two-Stage Graph Clustering Method.....	49
2.6 R Software and Programming Language.....	51
2.7 Electric vehicle (EV) considerations for power grid.....	51

<b>Chapter 3. Clustering Analytics for Streaming PMU Datasets .....</b>	<b>57</b>
3.1. Summary .....	57
3.2 Methods .....	57
3.2.1. Background .....	57
3.2.2 Streaming Phasor Datasets .....	58
3.2.3. Methodology .....	61
3.3 Results .....	61
3.3.1. Streaming frequency data .....	62
3.3.2. Streaming Voltage Data .....	64
3.3.3 Streaming Current Data .....	66
3.4 Interpretation and Discussion .....	68
3.5 Conclusions .....	70
<b>Chapter 4. Betweenness Centrality-based Identification of Critical Buses and Decomposition of Microgrids in IEEE Test-Bus Systems .....</b>	<b>72</b>
4.1. Summary .....	72
4.2 Methods .....	72
4.2.1. Background .....	72
4.2.2 Identification of Critical Nodes .....	73
4.2.3 Node Removal Methodology and Normalized Expected Impedance Distance .....	74
4.2.4 Power System Decomposition and Microgrid Specifications .....	76
4.2.5 Economic Dispatch Formulation .....	78
4.2.6 Methodology .....	82
4.3 Results .....	83
4.3.1 CBI evaluation .....	83
4.3.2 Discussion and interpretation of CBI and node removal .....	87
4.3.3 Results of graph clustering, modularity, and economic dispatch .....	88
4.4. Discussion and Interpretation .....	96
4.5 Conclusions .....	98
<b>Chapter 5. Investigation of Time-Series Clustering for Demand Profile Classification and Improved Load Forecasting .....</b>	<b>100</b>
5.1 Summary .....	100
5.2 Methods .....	100
5.2.1. Background .....	100
5.2.2 Understanding the smart meter data .....	102
5.2.3 Methodology .....	103
5.3 Results .....	104
5.3.1 Time-series clustering results .....	104

5.3.2 Forecasting Results .....	107
5.4 Discussion and Interpretation .....	109
5.4.1 Clustering Evaluation .....	109
5.4.2 Forecasting evaluation .....	110
5.5 Conclusions.....	110
<b>Chapter 6. Results and Future Directions .....</b>	<b>112</b>
6.1 Summary .....	112
6.2 Future directions.....	113
BIBLIOGRAPHY .....	115
APPENDIX I – R CODE FOR CLUSTERING IN PMU DATASETS.....	125
APPENDIX II – R CODE FOR CLUSTERING BASED MICROGRID DECOMPOSITION .....	132
APPENDIX III – R CODE FOR CLUSTERING SMART METER DATA AND LOAD FORECASTING.....	138



LIST OF FIGURES

Figure 1. Raw plot of sepal length vs sepal width of flowers observed..... 27

Figure 2. Example clustering scheme on flower sepal data with 3 clusters. .... 28

Figure 3. Comparison of DTW and Euclidean distance metrics for time-series.. 31

Figure 4. An example of a hclust dendrogram from PMU data with cutoff points on the y-axis ..... 35

Figure 5. Simple graph where  $V(G)=\{1,2,3,4,5\}$  and  $E(G)=\{1-2,2-4,2-5,3-4,3-5,4-5\}$ . ..... 42

Figure 6. Two stage graph clustering process visualization ..... 50

Figure 7. Global popularity growth of electric vehicles..... 55

Figure 8. Streaming frequency data with anomalies randomly inserted ..... 59

Figure 9. Anomalies inserted into streaming voltage magnitude data from PMU60

Figure 10. Anomalies randomly inserted into PMU current magnitude data. .... 60

Figure 11. k-means, DBSCAN, and k-median algorithms applied to 10 minutes streaming frequency data ..... 62

Figure 12. Run-time comparison for clustering algorithms on frequency data.... 63

Figure 13. Hclust centroid linkage density, violin, and box plots..... 63

Figure 14. k-means, DBSCAN, and k-medians and DI comparison for voltage data ..... 64

Figure 15. Run time comparison for the algorithms applied to streaming voltage data ..... 65

Figure 16. Hclust centroid linkage method: density plot, violin plot, boxplot ..... 65

Figure 17 a) k-means, b) DBSCAN, c) k-medians and d) DI comparison..... 66

Figure 18. Hclust centroid method: density plot, violin plot, box plot ..... 67

Figure 19. Run time comparison for current magnitude data..... 67

Figure 20. CBI and microgrid decomposition methodology flow chart..... 82

Figure 21. NEGD for all metrics in 118 bus removal..... 83

Figure 22. NEID comparison for all metrics in 118 bus removal ..... 84

Figure 23. NEID comparison for top 5 metrics in 118 bus removal..... 85

Figure 24. NEGD comparison for most influential indices of 300 bus removal ... 85

Figure 25. NEGD comparison for all metrics of 300 bus removal ..... 86

Figure 26. NEID comparison for most influential indices of 300 bus removal .... 86

Figure 27. 118-bus system clustered with admittance-weighted GN algorithm. B.) 118 bus system clustered with length-weighted GN. C.) 118 bus system clustered with two stage NG+GN. D.) 300 bus system clustered with admittance-weighted GN.....	91
Figure 28. Example decompositions for 300 bus system. A.) 300 bus system clustered with length-weighted GN. B.) 300 bus system clustered two stage NGGN.....	92
Figure 29. Yearly charging hours of the 200 EVs in the smart meter data .....	102
Figure 30. Power consumption for charging of each EV in one year. ....	103
Figure 31. Centroid daily demand profiles from 4-shape clustering for households w/ EVs .....	105
Figure 32. Centroids of daily household demand found by 3-shape clustering for households w/out EVs .....	106
Figure 33. Traditional Forecasting scheme for households w/ EVs .....	107
Figure 34. Traditional forecasting method applied to households w/out EVs ...	108
Figure 35. Proposed forecasting method that uses time-series clustering applied to households w/ EVs .....	108

LIST OF TABLES

Table 1. The results of clustering scheme displaying how many of each species belong to each of 3 separate clusters..... 28

Table 2. Battery specifications for EVs and PHEVs available in 2018..... 52

Table 3. Definition of MGRs..... 78

Table 4. IEEE 118 bus system with modularity scores for each clustering algorithm as well as MGRs ..... 89

Table 5. Modularity and MGRs for IEEE 300 bus system for GN algorithms..... 89

Table 6. IEEE 118 bus decompositions by algorithm and zone..... 93

Table 7. Bus assignments for each GN algorithm for 300 bus system ..... 93

Table 8. Economic dispatch cost results for 118 bus system ..... 95

Table 9. Economic dispatch cost results for 300 bus system ..... 95

Table 10. Generator rating/load ratio of IEEE 118 clusters ..... 96

Table 11. Generator rating/load ratio of IEEE 300 bus clusters..... 96

Table 12.CVIs for k-shape clustering of households w/ EVs ..... 104

Table 13. CVIs for DTW clustering for households w/ EVs..... 104

Table 14. Scaled CVIs to compare across indices for DTW clustering of households w/ EVs ..... 105

Table 15. CVIs for k-shape clustering of households w/out EVs ..... 105

## ACKNOWLEDGEMENTS

I would like to thank my wife, Setareh, for her patience and encouragement in this process. I would also like to thank my parents, John and Kristin as well as my siblings Matthew, Mason, and Mikayla for supporting me throughout my life. I am thankful for all of them.

I would like to express my sincere gratitude to my advisor, Dr. Prakash Ranganathan for his continuous guidance, support and advice which he had provided and the opportunity to pursue this degree and research. I'm very fortunate to have him as my advisor.

I am thankful to the members of my committee: Hossein Salehfar and Saleh Faruque, for their guidance and instruction as professors. I would also like to acknowledge the role of the Department of Electrical Engineering at University of North Dakota (UND), Grand Forks, North Dakota for providing me the opportunity to study and conduct research.

My thanks also goes to my friends and colleagues here in Grand Forks for supporting me both academically and socially.

Above all, I am grateful to God for giving me the grace, capability, and the opportunity to pursue this academic work and supports me all areas of life.

## ABBREVIATIONS

PMU	Phasor Measurement Unit
DI	Dunn's Index
DBSCAN	Density Based Spatial Clustering of Applications with Noise
BC	Betweenness Centrality
DTW	Dynamic Time Warping
EV	Electric Vehicle
CVI	Cluster Validity Index
WAMS	Wide Area Management System
SBD	Shape Based Distance
ARIMA	Autoregressive Integrated Moving Average
AR	Autoregressive
I	Integrated
MA	Moving Average
SIL	Silhouette index
DB	Davies Boulin index
DB*	modified Davies Boulin index
CH	Calinski-Harabasz index
SF	Score Function
COPI	Context-Independent Optimality and Partiality Index
Hclust	Hierarchical Clustering
RTU	Remote Terminal Unit
SCADA	Supervisor Control and Data Acquisition
GN	Girvan-Newman
BCGC	Betweenness Centrality Graph Clustering
NG	Nearest Generator
NEID	Normalized Expected Impedance Distance
NEGD	Normalized Expected Geodesic Distance
MGR	Microgrid Rules
NGGN	Nearest Generator Girvan Newman

ED	Economic Dispatch
L-GN	Length weighted Girvan Newman
A-GN	Admittance weighted Girvan Newman
SBD	Shape Based Distance
AMPL	Algebraic Mathematical Programming Language
MAED	Multi-Area Economic Dispatch
IT	Information Technology
DoE	Department of Energy

## ABSTRACT

The processing of data arising from connected smart grid technology is an important area of research for the next generation power system. The volume of data allows for increased awareness and efficiency of operation but poses challenges for analyzing the data and turning it into meaningful information. This thesis showcases the utility of clustering algorithms applied to three separate smart-grid data sets and analyzes their ability to improve awareness and operational efficiency.

Hierarchical clustering for anomaly detection in phasor measurement unit (PMU) datasets is identified as an appropriate method for fault and anomaly detection. It showed an increase in anomaly detection efficiency according to Dunn Index (DI) and improved computational considerations compared to currently employed techniques such as Density Based Spatial Clustering of Applications with Noise (DBSCAN).

The efficacy of betweenness-centrality (BC) based clustering in a novel clustering scheme for the determination of microgrids from large scale bus systems is demonstrated and compared against a multitude of other graph clustering algorithms. The BC based clustering showed an overall decrease in economic dispatch cost when compared to other methods of graph clustering. Additionally, the utility of BC for identification of critical buses was showcased.

Finally, this work demonstrates the utility of partitional dynamic time warping (DTW) and k-shape clustering methods for classifying power demand profiles of households with and without electric vehicles (EVs). The utility of DTW time-series

clustering was compared against other methods of time-series clustering and tested based upon demand forecasting using traditional and deep-learning techniques. Additionally, a novel process for selecting an optimal time-series clustering scheme based upon a scaled sum of cluster validity indices (CVIs) was developed. Forecasting schemes based on DTW and k-shape demand profiles showed an overall increase in forecast accuracy.

In summary, the use of clustering methods for three distinct types of smart grid datasets is demonstrated. The use of clustering algorithms as a means of processing data can lead to overall methods that improve forecasting, economic dispatch, event detection, and overall system operation. Ultimately, the techniques demonstrated in this thesis give analytical insights and foster data-driven management and automation for smart grid power systems of the future.



## Chapter 1. Introduction

The following sections serves to outline the proposed focus of work to satisfy the requirements of an M.S. Electrical Engineering. Specifically, this thesis work demonstrates and analyzes the utility of clustering algorithms for 3 distinct applications of smart grid datasets. The methods proposed provide novel algorithms, analysis, and implementations of algorithms for smart grid control and software applications. The algorithms analyzed have application in software development for smart grid automation and real-time situational awareness. The motivation, outline, contributions, and resultant publications from this thesis work are described in this introduction.

### 1.1 Motivation

The power grid is a critical infrastructure and the industrial backbone to the operation of any society. The United States power grid has a well-established record of reliability. However, the reliability of the power grid has had the unintended consequence of causing power systems technology to be slower to adapt with new technologies that have been embraced by other industries. As consumer demand and government incentives for smart devices and renewable energy has increased, this trend has begun to change. The power grid has begun to adapt and is becoming a "smart grid". One challenge posed by new smart grid technologies is coordinating and analyzing the mass amount of data that these devices generate. The specific challenge that this thesis examines, is turning the data provided by smart grid technologies into meaningful information to improve the operation of the power system. This thesis examines a small sector of smart

grid technologies and focuses on the algorithms that can aid in an automated or semi-automated decision-making process and increase the efficiency of grid operations.

## 1.2 Background

The term, “smart grid” is a term used in connection with ways to update and automate the functioning of the conventional power grid. According to the U.S. Department of Energy in [1], smart grid generally refers to "a class of technologies that modernize utility electricity delivery systems and bring them in line with the 21st century." More specifically, smart grid is the integration of remote control, automation, internet-of-things technologies, sensor networks, renewable energy sources, two-way digital communications, and data science methodologies into power systems. Though these technological advances show great promise, they also provide challenges.

The U.S power system has a long-established track record of reliability, but the integration of smart grid technologies will usher a new era of optimal operation, increased reliability, environmental consciousness, and increased efficiency. These types of technologies have been utilized in numerous other industries but have been slow to integrate into the field of power systems, partially due to the established reliability of the current system. Some of the key hardware technologies for smart grid initiatives examined in this thesis include phasor measurement units (PMUs) and smart-meters. These technologies have been developed for the tasks of utility area system management in the form of Wide Area Management Systems (WAMSs).

WAMSs are a nexus of software and sensor networks that allow real-time interaction with power system components. WAMSs are a crucial component to a smart grid because they provide an interface to monitor and manage grid operations. WAMSs rely heavily on technologies with two-way communication capabilities. PMUs and smart meters play vital roles in WAMS as they provide time-tagged data of crucial grid parameters including electricity demand, voltage, current, phase angles, and more. Without knowing the real-time status of the grid, a power system cannot be managed efficiently. PMUs and smart meters supply data which is interfaced in the software of a WAMS and allows for system operators to more efficiently manage the system.

The time-tagged measurements from these systems can be used for many power system applications such as state estimation, load forecasting, fault detection, microgrid operations, economic dispatch, and much more. The challenge posed by these new technologies is coordinating and utilizing the mass amount of data that they make available. Many of these metering technologies can provide data in volumes from 30 to 120 samples per second.

The data provided are meaningful, but the volume and scope presented by power systems applications makes the analysis and utilization of the data a challenging task. To make informed decisions based upon grid data or mitigate the risk of costly and dangerous grid failures, informative methods of power systems analysis need to be developed. This thesis work provides applications of data mining and applied mathematics techniques to investigate, analyze, and understand ambiguous data and connective topology of power systems

components. The methods investigated in this work efficiently utilize the data that can be gathered from smart grid technologies. Specifically, this work focuses on applications of data clustering that tangibly improves economic dispatch, grid anomaly detection, and load forecasting.

These technologies will increase operational efficiencies, allow for more consumer interaction with power consumption, and increase real-time situational awareness capabilities of utilities. Additionally, they will help with the integration of renewable energy sources that are beginning to penetrate the grid. More generally, these technologies represent a dramatic shift in the power systems industry. Smart grid is a broad term and encompasses many next generation power systems technologies. The scope of this thesis focuses on smart grid technologies that provide challenges in the realm of data analytics, so a full and complete discussion of smart grid technologies is beyond the scope of this work.

### 1.3 Thesis Outline

This thesis is organized as follows. **Chapter 2** contains methodological review and is divided into three main sections. The first section provides technical background on clustering algorithms applied to PMU datasets. The second section presents technical background on betweenness centrality, graph theory, and graph clustering in power systems context. The third section provides technical review for time-series clustering, forecasting, and energy demand forecasting. **Chapter 3** outlines the methodology and examines the datasets for a case study that investigates the efficacy of clustering algorithms to detect anomalous data in streaming PMU datasets. **Chapter 4** outlines the methodology and examines the

case study performed in performing graph clustering on IEEE test beds and observing the effect these algorithms have on economic dispatch. **Chapter 5** examines the datasets that were used in a time-series clustering paradigm and outlines the methodology for examining time-series clustering as a data processing step for smart-meter demand forecasting. Finally, **Chapter 6** summarizes the results and provides conclusions as well as directions for future work.

#### 1.4 Thesis Contributions

The following are the three objectives of this research work.

**Objective 1:** Evaluate and propose a clustering algorithm to detect anomalies in streaming PMU data.

To accomplish objective 1, the following tasks were performed

**Task 1:** Conducted literature review on existing automated techniques to detect anomalous data from PMU data.

**Task 2:** Investigated multiple automated algorithms for anomaly detection in PMU datasets.

**Task 3:** Evaluated the efficacy of the algorithms based upon a cluster validity index (CVI), Dunn Index (DI).

**Objective 2:** Develop a multi-criteria clustering method that efficiently decomposes a larger grid into potential microgrids

To accomplish this objective, following tasks were carried out:

**Task 4:** Conducted literature review on clustering methods for network graphs and the application with microgrids.

**Task 5:** Investigated multiple graph theory-based clustering algorithms and compared their efficiency using a Multi-Area Economic Dispatch (MAED) formulation in a case study with IEEE 118 and 300 bus systems to discover which algorithm resulted in the lowest dispatch cost for each bus system.

**Objective 3:** Investigate the effect of clustering time-series as a processing step to improve conventional time-series forecast methods for smart meter load forecasting.

To accomplish this objective, following tasks were carried out:

**Task 6:** Conducted literature review for clustering algorithms and forecasting techniques nuanced for time-series data

**Task 7:** Evaluated clustering techniques according to CVIs and devised a method of technique selection while comparing the accuracy of clustering-based forecasts with traditional forecasting techniques that do not use clustering

## 1.5. Publications

Journals/Book Chapter:

1. **Mitch Campion**, P. Ranganathan, and S. Faruque, “**A Review and Future Directions of UAV Swarm Communication Architectures**,” *Journal of Unmanned Vehicle Systems*, 2018. In Review.
2. Prakash Ranganathan, Kendal Nygard, **Mitch Campion**, Arun Nair. “**Decomposition of Microgrids in Large-Scale Electric Test Beds for Economic Dispatch Optimization.**” In: “*Distributed Linear Programming*

*Models in a Smart Grid.*" Power Electronics and Power Systems. Springer. Print. 2017

3. Erwan Olivo, **Mitch Champion**, and Prakash Ranganathan. "**Data Compression for Next Generation Phasor Data Concentrators (PDCs) in a Smart Grid.**" *Journal of Information Security* 07.05 (2016): 291-96.

Conference Publications:

1. Arun Nair, **Mitch Champion**, David Hollingworth, Prakash Ranganathan, "**Investigation of PJM Day-Ahead Load Forecasting for Economic Dispatch,**" *Proc. Of IEEE Electro Information Technologies Conference (EIT 2018)*, Rochester Hills, MI, 2018. In Review
2. M. Pozniak, J. Schwalb, **M. Champion**, J. Englund, E. Vettel, M. Nehring, P. Ranganathan, "**Next Generation Counter-UAS Platform Through UAV Swarms,**" *Proc. Of IEEE Vehicular Technology Conference (VTC 2018)*, Chicago, IL, 2018. In Review.
3. **Mitch Champion**, Prakash Ranganathan. "**Identification of Critical Buses Based on Betweenness Centrality in a Smart Grid**", *Proc. Of IEEE Electrical Power and Energy Conference (EPEC) 2017*, Saskatoon, SK, Canada, 2017
4. **Mitch Champion**, Martin Pozniak, Calvin Bina, Prakash Ranganathan, Naima Kaabouch, and Mark Boetl. "**Predicting West Nile Virus Occurrences in North Dakota Using Data Mining Techniques**", *Proc. of*

*Future Technologies Conference (FTC 2016)*, CA, USA, San Francisco.

N.p.: SAI Conferences, 2016.

5. Justin Pagel, **Mitch Champion**, and Prakash Ranganathan, "**Clustering Analytics for Streaming Smart Grid Datasets**", *Power Systems Conference 2016 (PSC 2016)*, March 8-11, 2016, Clemson, SC



## Chapter 2. Methodology

### 2.1 Data Mining and Smart Grid

As the complexity of the grid grows for a two-way communication between generation and consumers, a large focus is placed on integrating devices with highly capable sensors to more effectively interact and observe the status of the grid in real-time. These types of devices allow operators and even users to gain actionable intelligence pertaining to the operation of the grid and appliances that are connected to it. These devices can collect massive amounts of data. To turn this mass amount of data into meaningful information, data mining methods are necessary.

A concise definition for the umbrella term “data mining” is: “the process of discovering and extracting useful patterns in large data sources [2].” Data mining methods exist at an intersection and aggregation of many different fields including mathematics, statistics, computer science, and computational science. The practice of data mining is usually falls under the umbrella of data science. Data mining techniques consist of useful analytical tools including statistical analysis, clustering algorithms [3], predictive modeling algorithms [4], supervised and unsupervised learning, and many more [4]. Data mining is related to and is often a foundation for artificial intelligence and machine learning. The number of possible applications for data mining techniques is innumerable. Some notable applications of data mining that have been influential in modern society include image recognition techniques used by popular social media applications such as Facebook [5] and Snapchat [6], pattern recognition algorithms used by online retailers to suggest items to shoppers [7], and of course, as I am proposing, use in the modern power grid [8]. Because the practice of data mining is comprised of a large conglomeration of techniques

and has vast application, an accepted definition of data mining can vary slightly depending upon the source.

There are a large volume of existing literatures pertaining to methods of processing data, data mining, and data analysis for smart grid applications [8], [9]. The problem of big data and the promise of turning large volumes of data into operable intelligence is well-known in the field of smart grid. There is a volume of preliminary works and a few small private corporations building business models centered on data science methods and consulting for utility companies. Additionally, some of the larger energy companies have divisions within their Information Technology (IT) departments that specialize in big data management and analysis. This section will review some of the key literatures regarding big data and smart grid power systems.

A key literature that provides thorough foundation for this research topic is [10]. It is a book published in 2016 by the “National Academy of Engineers.” The work outlines in detail many avenues of analytical research foundations for the next generation smart grid. The work was published jointly with the Department of Energy (DoE) and National Academic Press. The work serves as a reference to preliminary works but especially is helpful in outlining the challenges associated with smart grid and the specific areas of mathematics, computational science, and data mining that may lead to breakthrough in these challenges. Some of the priorities listed include data-driven models of the electric grid, data-driven approaches for improving planning, operations, maintenance, and decision-making protocols, machine-learning models for hazard modeling, and visualization methods for complex data and systems [10]. This document also provides a thorough background on many data mining or mathematical methods that are proposed

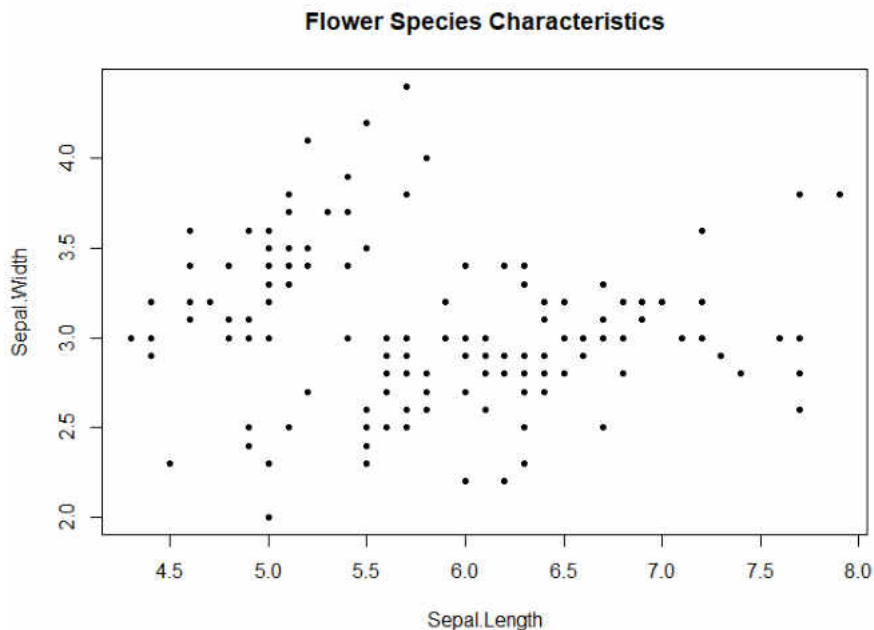
to meet these challenges. This work was compiled by leading experts in industry and academia and lays a foundation for focus areas of research topics, including this thesis.

In [11] a software framework is proposed that makes use of data clustering methods to provide system operators with enhanced situational awareness. The work outlined by [12] and proposes the use of spanning trees to classify data from smart grid devices. Additionally in [13], [14] general overviews of applications and assessments of clustering methods for power systems data is analyzed. Clustering has also been investigated in determining grid topology.

Research works [15]–[17] investigate different methods of organizing grid topology based on graph clustering methods. Data mining and analytical techniques have been useful in detection of critical components in power systems as proposed in [18]–[22]. Many of the methods for detecting critical nodes combine the use of data mining and graph theory. Thus, by combining these methods with power systems data can aid identifying which sections of the power grid are most central and vulnerable to cascading failures. There is very limited work carried in this research area. A number of publications focus on the development of novel energy management systems that make use of data mining and computational techniques [11], [23], [24]. There is abundant literature showcasing data mining and machine learning methods in demand, price, and electricity forecasting [25]–[30]. The utilization of data mining methods shows great promise in power systems and there is a plethora of applications for smart grid datasets.

### 2.1.1.1 Data Clustering

Cluster analysis is the task of grouping a set of objects or data points in such a way that objects in a group are more like each other than to objects contained in other groups. The purpose of clustering is to get an improved understanding of the associations that exists in a dataset [31]. Clustering algorithms have been applied to provide classification of and intelligent insights from otherwise ambiguous data. In general, the purpose of clustering is to obtain an improved understanding of the input group or dataset.



**FIGURE 1. RAW PLOT OF SEPAL LENGTH VS SEPAL WIDTH OF FLOWERS OBSERVED**

A simple example of an application of clustering would be classifying species of flower based upon sepal widths and sepal lengths of flowers observed in a garden. If there are 3 known species a clustering algorithm with an output of 3 clusters can be applied and the accuracy of the scheme can be tested. An example plot of the flower pedal data with and without clustering is shown below in Figures. 1 and 2 respectively. The accuracy of the clustering scheme is observed by table 1. By comparing the plots

from Figures 1 and 2, it is observed that there are three distinct clusters of flower species based upon the sepal width and lengths. The centroids (average values) of the 3 clusters are plotted as bold points in Figures. 2. The specific data points are identified by the clustering scheme and the accuracy of the scheme is shown in table 1. Further discussion on clustering algorithms is discussed in the literature review section.

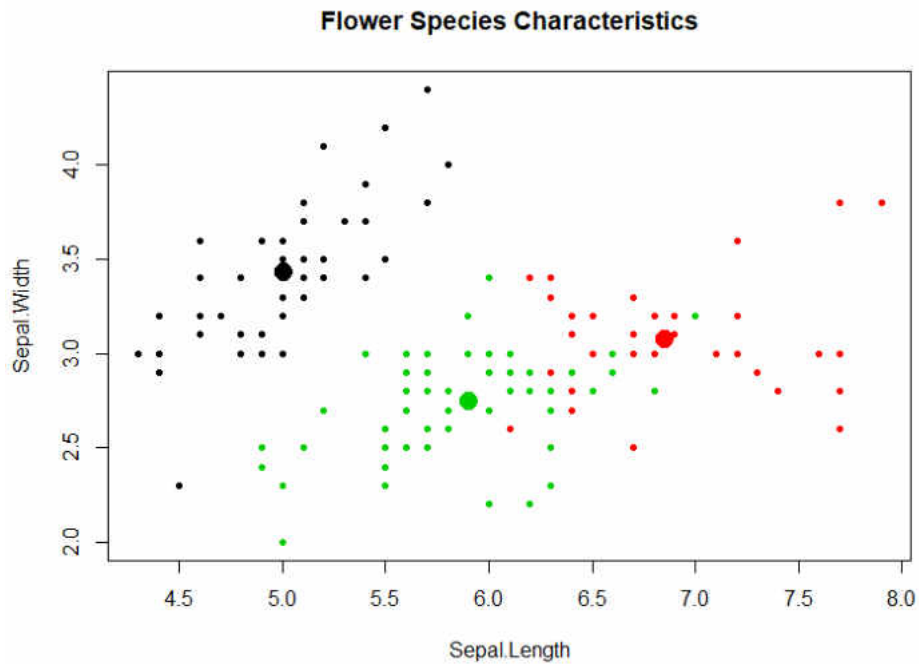


FIGURE 2. EXAMPLE CLUSTERING SCHEME ON FLOWER SEPAL DATA WITH 3 CLUSTERS.

Species\Cluster #	1	2	3
Setosa	50	0	0
Versicolor	0	2	48
Virginica	0	36	14

TABLE 1. THE RESULTS OF CLUSTERING SCHEME DISPLAYING HOW MANY OF EACH SPECIES BELONG TO EACH OF 3 SEPARATE CLUSTERS.

Clustering can be supervised or unsupervised. Supervised clustering means that the output of a clustering algorithm can be trained or verified empirically. Unsupervised

clustering is a more ambiguous task where the output is not well understood and cannot be explicitly verified. Clustering algorithms are applied to a wide variety of different tasks, issues, and fields of study. Time-series clustering is a variation of clustering with modifications due to specific considerations of time-series data. The most important aspect to any clustering scheme or algorithm is how the distance between the objects of clustering are defined. Clustering is traditionally applied to an ' $n$ ' dimensional data set. In traditional ' $n$ ' dimensional datasets, distances between clustering objects can be defined by traditional distance metrics such as Euclidean and Manhattan distance depending upon the application. To cluster time-series data different metrics such as shape-based distance (SBD) and DTW are appropriated [32], [33].

To evaluate clustering algorithms, cluster validity indices (CVIs) are computed. CVIs are a quantitative method to evaluate the output of unsupervised clustering schemes. They can also be used for supervised schemes, however, the CVIs discussed in this work are typically used for unsupervised schemes due to the nature of unsupervised clustering. The basic premise of CVIs is to quantitatively evaluate how compact and well separated from one another the clusters resulting a clustering scheme are. They are especially helpful when comparing and evaluating multiple clustering schemes to analyze relative efficiencies. Common CVIs include: Dunn's Index (DI) [34], Silhouette (Sil) [35], Davies-Bouldin (DB) [36], modified Davies-Bouldin (DB\*) [36], score function (SF) [37], Calinski-Harabasz (CH) [38], and context-independent optimality and partiality indices (COPI) [39]. Research shows that even the insight from cluster evaluation criteria are somewhat ambiguous and no single cluster validation index is

necessarily better than another [40]. This work uses a combination of DI, DB, DB\*, SF, CH, and COPI analyzed to evaluate time-series clustering schemes of smart meter data.

This work specifically uses DI for evaluation of clustering in PMU datasets. DI identifies sets of clusters that are compact with a small variance between members of the cluster yet distinctly separated from other clusters [34]. Ideally, average values of the separate clusters are distinctly separated from one another, but the internal cluster variances are small. A higher CVI indicates better clustering. This is important in evaluating the significance of the efficiency of the clustering algorithms [34], [41].

### 2.1.2 Time-series Clustering

With the increase in deployment of smart meters, the ability to analyze individual residential energy consumption is becoming possible. This data is stored and analyzed as time-series data. The challenge posed by residential usage is that for any given utility, there are many households to serve. The challenge of forecasting for each household is tedious, yet important for optimal system management. One methodology that meets this challenge is time-series clustering. Time-series clustering has been shown to be an effective method to extract information from time-series databases for the purposes of pattern discovery.

Time-series clustering poses unique considerations compared to traditional data clustering. Euclidean distance is the most widely used distance metric for general clustering schemes. The Euclidean distance between two time-series,  $X$  and  $Y$ , both of length  $m$  is calculated using equation 1.

#### ( 1 ) EUCLIDEAN DISTANCE

$$ED = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$$

Though Euclidean distance is a common metric, it is limited by its simplicity. Specifically, the Euclidean distance between time-series is prone inaccuracies in comparing similar time-series that have shifts in the time domain. To address this weakness, a technique called dynamic time warping (DTW) was introduced. DTW accounts for the Euclidean weakness by allowing for non-linear and elastic alignments of time-series based on localized characteristics rather than rigid point to point distances. DTW detects non-linear alignments of time series by establishing an  $m$ -by- $m$  matrix,  $\mathbf{M}$ , with the Euclidean distance between any two points of  $X$  and  $Y$ . A warping path,  $\mathbf{W} = \{w_1, w_2, w_3, \dots, w_n\}$  where  $n \geq m$ , is established that defines a mapping between  $X$  and  $Y$ . This path can be computed on matrix  $\mathbf{M}$  with dynamic programming and the formula for DTW is a minimization described by equation 2.

( 2 ) DYNAMIC TIME WARPING

$$DTW(X, Y) = \min \sqrt{\sum_i^k w_i}$$

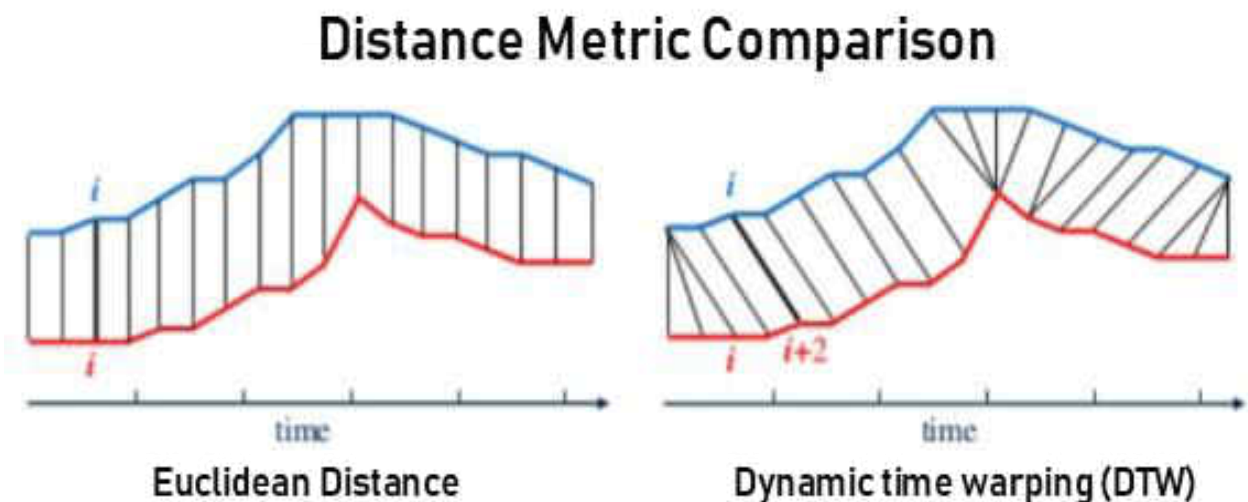


FIGURE 3. COMPARISON OF DTW AND EUCLIDEAN DISTANCE METRICS FOR TIME-SERIES



A visual example of the difference between Euclidean and DTW distance metrics is illustrated by figure 3. In figure 3, the non-linear mappings of DTW from one time-series to another are demonstrated in comparison to the linear mappings of Euclidean distance. While many of the same types of clustering algorithms can be deployed for time-series data, an altering of the distance metric to DTW is appropriate given the unique considerations for time series.

## 2.2 Clustering Algorithms

### 2.2.1 k-means

The k-means algorithm is a well-known and popular clustering algorithm which was first proposed by Lloyd [42]. The goal of this algorithm is to minimize the variability within clusters and maximize the variability between different clusters. The use of k-means requires determining the number of clusters that are desired. For applications where it is unclear on the number of desired clusters, the utility of this algorithm can be limited. k-Means functions by initially assigning all data points into random clusters and computing the centroids of those clusters. After this task, each data point is assigned to the centroid that is closest (or most similar) to. The algorithm then repeats several iterations until no changes are made in the assignment of data points [31].

### 2.2.2 k-medians

The k-medians algorithm is a variation of k-means [43]. The effective difference between them is that k-means minimizes Euclidean distance of each point to a cluster centroid while k-medians minimizes inter-cluster Manhattan distance. The Manhattan distance is the sum of the differences of all the corresponding data points in a cluster [44]. k-Medians, like k-means, also requires the input for number of desired clusters.

### 2.2.3 Density Based Spatial Clustering for Applications with Noise (DBSCAN)

DBSCAN [45] is another clustering algorithm applied to openPDC data. DBSCAN finds core samples of high density and expands clusters from them. Theoretically, this algorithm is effective for data that contains clusters of similar density and that have some associated noise. To apply DBSCAN, two parameters are needed. The first parameter is a positive number, *eps*. The second parameter is a natural number, *minPoints*. If the number of points within a distance, *eps*, from a starting point is greater than *minPoints*, then these points will be clustered together. The algorithm then recursively builds by checking all the new points to find out if there are several points greater than the *minPoints* value within a distance, *eps*. After all the points have been added to the cluster, a new arbitrary point is picked, and the process is repeated. If that arbitrary point has fewer than points than *minPoints* within the distance *eps*, it is considered a noise point. The goal in choosing proper values for *eps* and *minPoints* was to maximize the value of a CVI.

### 2.2.4 Hierarchical Clustering (hclust)

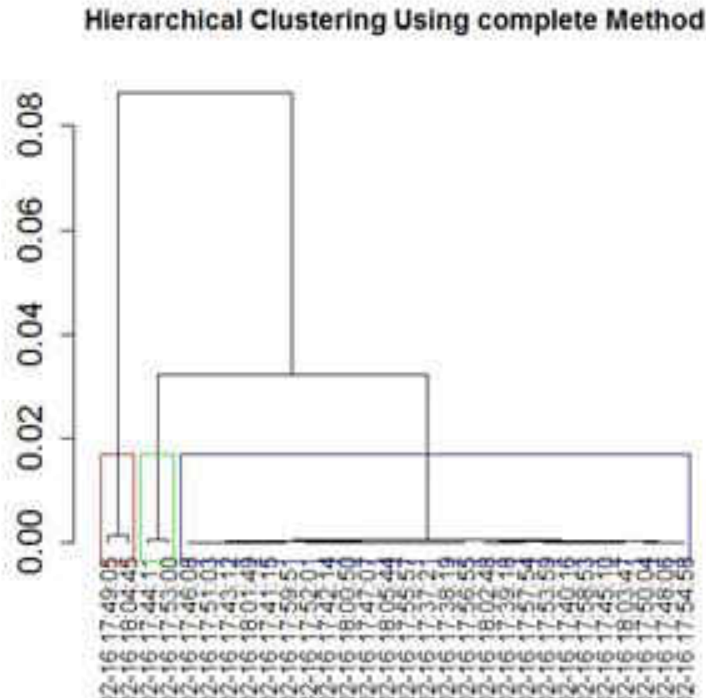
Hierarchical clustering (hclust) [31], [44] is a set of algorithms that group data by creating a cluster tree called a dendrogram. An example dendrogram is shown in figure 4. There are two different types of hierarchical clustering: agglomerative and divisive [44]. In this thesis, agglomerative clustering is used. Agglomerative hclust initializes each data point as an individual cluster and then proceeds step by step to merge the closest pairs of clusters until there exists only one cluster. One of the main advantages of hclust is that there is no need to specify the number of clusters. If a specific number of clusters is desired, an hclust tree can be cut at a desired level. The cutoff values can be selected such that one can theoretically analyze any step of the hclust algorithm. The y-axis of the

dendrogram shown in figure 4 shows the potential cutoff values. In figure 4 a cutoff value of .019 was used. 3 clusters are observed at this cutoff. For this project, the cutoff values are selected according to the maximization of Dunn's index.

In general, a smaller number of clusters produces a larger Dunn's index. By viewing an hclust dendrogram one can infer the number of clusters that exist at each cutoff value. By process of viewing the dendrogram, cutoff values were selected that associated with smaller numbers of clusters to observe maximum values of Dunn's index. Typically, hclust is difficult to use on large datasets like openPDC due to the requirement of a distance matrix describing the dissimilarity between each point of data. For large datasets, this matrix can require a large amount of memory. This also causes the dendrogram to be very convoluted as shown in Figure 4. Other visualization techniques have been adopted for this thesis to analyze hclust data.

A difficult choice of hclust is to define the distances between clusters [31], [42]. Euclidean distance is the most common method to use as a measure of dissimilarity. However, there are several different methods to measure the Euclidean distance, or any distance metric could theoretically be used (Manhattan distance etc...). These methods are called clustering linkage criteria. Some cluster linkage criteria methods are single-link, complete link, average link, centroid link, and Ward's method. Single-link distance between clusters is computed as the distance between the two closest elements of the clusters. The complete-link distance between clusters is computed by the distance between the most distant elements of the clusters. The average-link distance between clusters is computed by the distance between the average of all pairwise distances between clusters. The centroid-link distance between clusters is computed by the

distance between the centroids of the two clusters. Ward's Method to define the distance between clusters is computed by the difference between the variance of the two clusters.



**FIGURE 4. AN EXAMPLE OF A HCLUST DENDROGRAM FROM PMU DATA WITH CUTOFF POINTS ON THE Y-AXIS**

2.2.5 k-Shape Algorithm

k-Shape [32] is a relatively new time-series clustering algorithm proposed in 2015. K-Shape is a time-series clustering paradigm for time series classification, shape extraction, and analysis. k-Shape is similar to k-means clustering as it is a partitional clustering algorithm and requires a user-defined input to determine the 'k' number of clusters to be defined. It is also similar algorithmically in that it contains an iterative procedure and a refinement phase. The centroids of each cluster in k-shape are found using cross correlation measures. In the assignment phase of the algorithm, shift invariance is enabled through a distance metric called shape-based distance (SBD) as opposed to a Euclidean distance metric that is used in k-means. SBD is a distance metric based on coefficient normalized cross-correlation of time-series and is an appropriate

metric to extract the shape of a time series when analyzing its 'distance' or similarity to another time-series [32]. SBD works by z-normalizing time series and determining distance based on cross-correlation. SBD is used to update cluster memberships by calculating the time-series centroids and by defining the clustering of each time-series data into the cluster with the nearest centroid. SBD is a defining characteristic of the k-shape algorithm, and is a central contribution to the work in [32].

#### 2.2.6 Partitional DTW

Partitional DTW is an algorithm without a proper name. Partitional DTW is simply a clustering algorithm that uses a partitional process and DTW as its distance metric. Partitional clustering algorithms are also called "center-based" clustering algorithms. Partitional clustering algorithms define cluster centers which are called centroids. Then, a partitional algorithms will assign data objects to the centroid that each data object is closes to according to the defined distance metric [46]. As an example, k-means is a partitional algorithm but traditionally uses Euclidean distance as the defining metric to determine the distance between clustering objects. Partitional DTW is essentially the k-means algorithm but instead of Euclidean distance, DTW is used as the distance metric. This method was employed as a time-series clustering method due to the ability of DTW to perform clustering on time-series data [31].

### 2.3. Forecasting for Power Systems and Smart Grid Applications

Forecasting is the process of using mathematical modeling to predict a future event [4]. Forecasting is a necessary and important function virtually in any industry. In the case of electric utilities, the importance is magnified. While other industries can store their products as a buffer against inaccurate forecasting, the magnitude of electrical energy

provided by power companies cannot yet be effectively stored in mass quantities. Because of this, power must be delivered as soon as it is generated. As a result, utility companies are increasingly required to develop formal load forecasting models to support their decisions about operation, planning, and maintenance.

Just as in other industries, electricity price depends on the equilibrium between the supply and demand. Balancing the supply and demand of power is a delicate task and predicting it ahead of time is even more challenging. Because forecasting is such a challenging task, high importance is placed on models that can provide accurate results. For this reason, utility companies have directed their attention toward forecasting and invest considerable resources to the task. Further discussion of forecasting methodologies, applications, and algorithms is found in the literature review and methodology sections of this thesis. This work proposes time-series clustering as a processing step to a forecasting scheme and observes its effect on forecast accuracy.

There is no single forecasting that can satisfy all the needs of a utility. A common practice is to use the different techniques for different purposes. With so many applications, it is unrealistic to establish a single forecasting technique to apply to every problem. The classification of different forecasts not only depends upon the business needs, but also on the other factors that drive the electricity consumption.

In architecture and engineering it is often stated that “Form follows function.” This means that the design of an object or product arises from how that object will be used. This is also true in the realm of forecasting. A single type of forecasting doesn’t satisfy the needs of all forecasting problems. Different types of forecasting are needed because of drastically different situations in which forecasting might be used. In the case of electric

load forecasting, different methods of forecasting can be divided into the following categories:

- Very short-term load forecast: ranges from few minutes to few hours.
- Short term load forecast: ranges from one day to two weeks.
- Medium term load forecast: ranges from two weeks to three years.
- Long term load forecast: ranges from three to five years.
- Fine-grain interval forecasts: Forecasts that predict values for data in fine-grained intervals (seconds, minutes)
- High granularity forecasts: Forecasts that predict values for high granularity (days, weeks, months)
- Time series forecasts: Forecasts of time-series data
- Single variable forecasting: Forecasts that use only a single variable.
- Multivariate forecasting: Forecasts that use variables effecting the target forecasting variable.
- Application specific forecasts: market, price, and demand are all examples of different applications of forecasting for power companies.

There are many different types of forecasting for many different types of applications. However, one thing that all forecasting schemes have in common is the desire to provide high accuracy. To measure a forecast accuracy, forecast accuracy metrics must be defined. There are many forecast accuracy metrics, however a few well-known and utilized metrics are Mean Absolute Percentage Error (MAPE), Mean Absolute Deviation

(MAD), and Mean Squared Deviation (MSD). MAPE, MAD, and MSD were all utilized in tandem this work to define forecast accuracy. Each metric has its own strengths that through analysis of all three metrics, a clear understanding of a forecast accuracy can be obtained. These metrics are defined as follows:

**( 3 ) MAPE**

$$MAPE = \frac{\frac{\sum|Actual-Forecast|}{Actual} * 100\%}{n}$$

**( 4 ) SUM OF SQUARED DEVIATION**

$$SSD = \sum|Actual - Forecast|^2 = \sum|Error|^2$$

**( 5 ) MEAN SQUARED DEVIATION**

$$MSD = \frac{\sum(Actual - Forecast)^2}{n} = \frac{SSD}{n}$$

Where  $n$  is the number of observations. In all cases a lower metric indicates more accurate forecasting. All three metrics compare a forecasted value, with an actual observed value to determine accuracy. MAPE is a good metric to quantify how good a prediction is on average and displays as a percentage, which is easy to interpret. SSD and MSD are more sensitive to high errors of individual observations due to the squared term in the formula. Because of this sensitivity, SSD is a good metric to analyze how consistent the accuracy of a forecast is. Over a period of point forecasts, if one point in the forecast is very inaccurate compared to the corresponding observed value, the squared term will cause the MSD or SSD to have a high value. For this reason, SSD and MSD are good metrics to determine forecast consistency. Forecasts that contain low MAPE, SSD, and MSD are accurate, highly desirable, and difficult to obtain. A framework that simultaneously analyzes both MAPE, MSD, and SSD is appropriate in this work.



### 2.3.1. Forecasting Algorithms

A variety of models are used for different types of forecasting purposes. As such, a myriad of mathematical models have been implemented for load forecasting. Some types of models that have been utilized include methods such as Autoregressive Integrated Moving Average (ARIMA), exponential smoothing, Loess, support vector machines, Neural Networks, and hybrid combinations of multiple algorithms. Forecasting algorithms are designed with mathematical formulations that require them to be applied appropriately. For example, neural networks are most effectively applied to situations with high volumes of data [47]. Partial least squares regression, is most effectively applied to situations where there are many regressor variables that may have collinearity [4], [48], [49]. ARIMA is a flexible algorithm and is appropriately applied to time-series data, particularly time-series that display, trend, seasonality, and cyclical natures. The work in [29] investigates ARIMA for day-ahead spot price forecasting. Additionally, [50] provides a great resource overviewing ARIMA, exponential smoothing (ES), and other statistical forecasting algorithms. The authors in [25] and [51] use a hybrid model and neural networks for electricity demand forecasting. This work specifically focuses on the implementation of a loess filter followed by ARIMA forecasting. This method has been demonstrated as an effective method for day-ahead load forecasting. A brief description of these algorithms follows.

### 2.3.2. ARIMA

ARIMA is a time-series forecasting algorithm that is based on three components of the time series on which it is applied to. The three components are the “autoregressive” component (AR), the “integrated” component (I), and the “moving average” component (MA). A non-seasonal ARIMA model is classified as an “ARIMA (p,d,q)” model, where

- **p** is the number of autoregressive terms
- **d** is the number of non-seasonal differences needed for stationarity, and
- **q** is the number of lagged forecast errors in the prediction equation

Consider the ARIMA (p,d,q) is expressed as:

### ( 6 ) ARIMA POLYNOMIALS

$$\phi(B)(1-B)^d X_t = \theta(B) Z_t$$

Where  $\phi$ ,  $\theta$  are the  $p^{th}$  and  $q^{th}$  degree polynomials;  $d^{th}$  is a non-negative differencing operation. It is often a case that stochastic processes may not have a constant level so they inhere homogeneous behaviors over time. If d is a non-negative integer,  $X_t$  is said to be an ARIMA (p,d,q) processes id  $(1-B)^d X_t$  is an ARMA (q,p) processes [52].

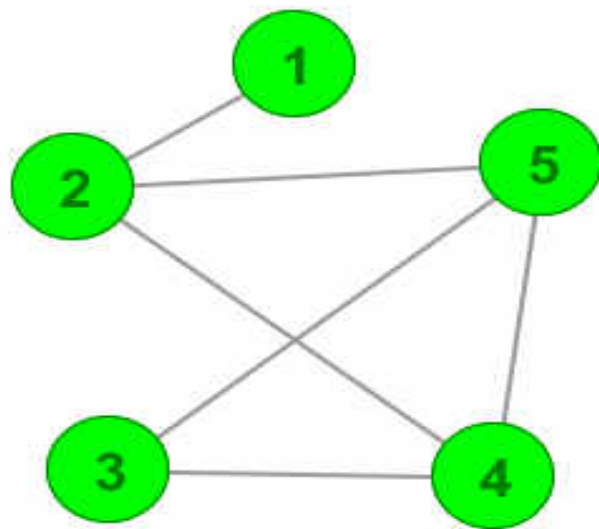
#### 2.3.3. Seasonal time-series decomposition

Time series decomposition is a technique to observe seasonality and trend patters from within a time series. There are many ways to decompose a time series into its seasonal and trend components. The simplest way to do this is using simple moving averages of varying window sizes. This work utilizes the loess [53] algorithm to accomplish the decomposition as a pre-processing stage to an ARIMA forecasting scheme [54].

#### 2.4 Smart Grid Modeling Using Graph Theory

A graph is a mathematical structure that represent pairwise relationships or connections between objects [55]. A graph is a set of vertices (*nodes*) that are connected edges (*lines*). Graph theory is a branch of mathematics that studies connections and relationships between objects using graphs [55]. Graphs can also be called networks. In

this thesis, graph and network are used interchangeably. Due to the inter-connectedness between various objects (*e.g.*, *circuit breakers, feeders, transmission lines, sources, and loads*) in a power grid, these grids can be intuitively represented as a graph. Formally, a graph's notation is given as a pair of sets,  $G = (V, E)$ , where  $G$  is the graph,  $V$  is the set of vertices, and  $E$  is the set of edges that are formed by the vertex pairs. A vertex set is a concatenated list of the name for each vertex in a graph and is denoted by  $V(G)$ .



**FIGURE 5. SIMPLE GRAPH WHERE  $V(G)=\{1,2,3,4,5\}$  AND  $E(G)=\{1-2,2-4,2-5,3-4,3-5,4-5\}$ .**

An edge list is a concatenated list of connected vertices in a graph and is denoted by  $E(G)$ . As an example, Figure 5 shows a graph with  $V(G) = \{1,2,3,4,5\}$  and  $E(G) = \{1 - 2, 2 - 4, 2 - 5, 3 - 4, 3 - 5, 4 - 5\}$ . Figure 5 displays a simple un-directed graph. An un-directed graph is one where the graph's edges are bidirectional [55]. For the purposes of this work, bus-system graphs are un-directed graphs. The following subsections describe criteria related to graph theory that can be used to determine grid-decomposition structures.

The raw graph topology of a given graph does not provide any functional information about the actual system that the graph represents. In terms of a power-grid

system, there is information about the buses and transmission lines within the grid that must be considered in order to adequately model the system. This modeling can be accomplished via the use of vertex and edge weights. Weights are simply a numeric value that is assigned to graph objects to convey some functional information about the graph or a specific graph object. A common example of an edge weight is assigning a numerical value to an edge that corresponds to the length of that edge. In this work, the notation for an edge weight is  $W_{m,n}$ , where  $m$  and  $n$  are vertices in  $V(G)$  such that  $W_{m,n}$  is the weighted value for the edge that connects bus  $m$  to bus  $n$ .

In this thesis, four metrics are considered for edge weights in a smart-grid power-transmission system. These metrics are i) *topological weight*, ii) *admittance*, iii) *impedance*, and iv) *line-length weights*. The topological weight assumes that  $W_{m,n} = 1 \forall m,n \in E(G)$ . The topological weighting metric captures the trivial topological connections of the graph and displays no bias toward certain network objects. The admittance-based edge weights are determined based upon calculating the transmission-line admittance. For this metric, admittance weight is given by  $W_{m,n} = \frac{1}{|R_{m,n} + jX_{m,n}|}$ , where  $R_{m,n}$  is the resistance of the transmission line that connects bus  $m$  to bus  $n$  and  $X_{m,n}$  is the line's reactance. Impedance weight is the inverse of the admittance weight. Length-based weighting assigns  $W_{m,n}$  equal to the transmission line's length. For this work, IEEE 57, IEEE 118, and 300-bus test systems were used. Approximations of the transmission-line length were calculated according to the method outlined in [56]. This method first converts the per-unit reactance value to the actual value using an assumed  $S_{base} = 100MVA$  and  $V_{base} = 135kV$ . The length of the line is then calculated, assuming a conversion factor of  $.7\Omega$  per mile.

These weights are static weights in that they are constant for a given power system. Other works have considered similar static metrics as well as dynamic metrics that include power flow [15]. The static edge weights can be interpreted such that strongly connected vertices are more likely to be clustered together. Topological weights represent a network's true connectivity. Admittance/impedance weights reveal the internal electrical structure based on the network's electrical distance [15], [17].

#### 2.4.1 Degree and Eccentricity

Degree and Eccentricity are two attributes that are defined for every vertex in a graph. Degree is defined as the number of vertexes that are incident to a specific vertex in a graph [57]. In other words, degree is simply the number of nodes that are connected to a given node. Eccentricity of a graph vertex is the maximum graph distance between the defined vertex and any other connected vertex in the graph [55]. Degree and eccentricity are well known attributes of graph objects, and their importance relating to power systems topology was studied in this thesis.

#### 2.4.2 Betweenness Centrality

Betweenness centrality (BC) [58] is an index that quantifies a vertex or edge's centrality in a network. In order to understand BC, the graph-theory concept of shortest paths needs to be understood. The shortest-path problem [59] is a common concept in the study of graph theory. The problem is defined by finding the path between two given vertices in a graph such that the sum of the edge weights of the path's constituent edges is minimized. A path in an un-directed graph is denoted by  $P = \{v_m, v_1, v_2 \dots v_n\}$ , where  $P$  is the path and  $v_m, v_n$  are vertices in graph  $G$  that are contained in the path from  $v_m$  to  $v_n$ .

A more formal definition of BC is the number of shortest paths from all vertices in a graph to all other vertices in the graph that pass through a particular object [58], [60], [61]. Betweenness centrality can be calculated for vertices or edges. Either of these calculations indicates how central, connectively important, or “highly traveled” a particular edge or vertex is within a graph. This metric is of importance for a smart-grid transmission system due to the ability to quantify vertices or edges that are of high connective importance to the network. Buses and/or transmission lines with relatively high BC may be more likely to cause cascading problems in the event of a failure that bus or line.

## 2.5 Graph Clustering

“Graph clustering” is a term with several aliases, depending upon the application. In general, graph clustering, network-community detection, graph partitioning, and graph decomposition are different aliases by which similar processes are occurring. These aliases all mean to discover community relationships between nodes within a graph. These “communities” are characterized by relatively dense interconnections with relatively sparse connections between groups. Graph-clustering algorithms are designed to identify and to quantify where these community structures exist within a graph.

Graph clustering algorithms perform similar functions to data clustering algorithms, the only difference is the type of data or objects that the algorithms are applied to. In the case of graphs, clustering algorithms detect dense connections of nodes within a large graph or network [62]. This thesis examines clustering algorithms applied to power-transmission systems to form power-zone community structures that, for intents of analysis, are designated as microgrids.

Several algorithms perform graph clustering based on BC. Betweenness centrality graph clustering (BCGC) makes use of BC to identify key objects in a graph and define community structures around those objects. Another way of thinking about BCGC is a quantification of the likelihood that an edge is between community structures in a graph. BCGC algorithms make use of this betweenness metric functional by using it to distinguish community structures in a graph. A notable algorithm for betweenness centrality clustering is the Girvan-Newman (GN) algorithm [60], [63] and is discussed in more depth in the literature review chapter.

#### 2.5.1 Graph Cluster Modularity Index

A method for quantifying the strength of a graph-clustering is necessary to quantitatively understand how well a graph decomposition is clustered. Consequently, there is need for graph modularity [64], and [65]. The modularity index measures the strength of dividing a graph into clusters. The cluster decompositions with high modularity scores have dense connections between the vertices within clusters but sparse connections between the vertices in other clusters. Modularity is a CVI for graph clustering schemes. The calculation of graph-cluster modularity allows for quantitative optimization of a graph-clustering scheme. The calculation of modularity first involves constructing matrix  $e$  with dimensions  $k \times k$ ; element  $e_{ij}$  is the fraction of all graph edges that link vertices in cluster  $i$  to vertices in cluster  $j$ . Conversely, the trace of this matrix,  $Trace(e) = \sum_i e_{ii}$ , is the fraction of edges in the graph that connect vertices in the same cluster. The trace has a maximum of  $Trace(e) = 1$ . In an efficient graph-clustering scheme, the trace is, ideally, near to 1. While this number is important, it fails to signify

any information about connections to a clustering scheme's intercluster structure [31], [61], [64].

The modularity index goes another step by including inter-cluster connections. Here, modularity defines a row sum,  $a_i = \sum_j e_{ij}$ , that represents the fraction of edges that connect to the vertices in cluster  $i$ . Regarding these values, modularity is calculated by:

**( 7 ) MODULARITY INDEX**

$$Q = \sum_i (e_{ii} - a_i^2) = \text{Trace}(e) - \| e^2 \|$$

where  $\| x \|$  indicates the sum of the elements for matrix  $x$ . This value measures the fraction of the graph's edges that connect vertices of the same cluster minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices. This metric essentially compares the connections of one scheme to the same scheme with the same number of random connections. If the number of within-cluster edges is no better than random, then  $Q = 0$ . The maximum value of  $Q$  is 1. Numbers near 1 indicate a stronger cluster structure. In practice, values for networks typically fall between 0.3 and 0.7. Higher values are considered to be rare [31], [61], [64] [65].

2.5.2 Girvan-Newman (GN) Algorithm

The GN algorithm detects communities or clusters within a graph by iteratively removing edges from the original network graph. After the removal of edges, the remaining connected components of the network graph are the communities. The GN algorithm removes edges based upon the betweenness index of each edge. Removing edges with high betweenness is a method of separating community structures within a graph from one another. The steps of the GN are as follows:



1. The betweenness of all edges within a graph are calculated.
2. The edge with the highest betweenness is removed.
3. The betweenness of all edges affected by the removal of this edge are then recalculated
4. Repeat starting from step 2 until a desired cutoff has been obtained [60], [63].

The stopping point or cutoff of the algorithm can be determined in terms of iterations, a desired betweenness, an optimality of graph modularity [64], when a desired number of clusters has been formed, or when there are no more edges to be removed. The algorithm is somewhat similar to agglomerative hierarchical clustering algorithms in that in a step by step manner, the algorithm decomposes the original graph. This step by step decomposition can be viewed as a dendrogram like hclust.

#### 2.5.3 Nearest Generator (NG) Clustering

Nearest-generator clustering is a simple graph clustering method utilized in this thesis specifically due to the power systems' requirements. The nearest-generator method is appropriately named because the algorithm functions by assigning each bus in the system to a cluster defined by the generator to which it is nearest according to a desired edge-weight metric, such as transmission line length or impedance. This method was developed for a few reasons. The first reason is the trivial logic of assigning a demand bus to the generator to which it is nearest. The second reason this method was developed was that it is an efficient way to ensure that the cluster decompositions follow the microgrid rule of containing at least one generator. The authors know of no graph-clustering algorithm that, by default, would cluster the bus system in a way where each cluster would contain at least one generator.

#### 2.5.4 Two-Stage Graph Clustering Method

To adjust for the scalability of bus systems as they get larger, a two-stage method of graph clustering was applied. Generally, as the bus system's size increases, the number of clusters formed by a graph-clustering algorithm will also increase. As an example, when betweenness-centrality clustering is applied to the IEEE 300-bus system, 14 clusters result as the scheme with optimal modularity using this algorithm. To decrease the number of clusters while still respecting the optimal modularity and improving the certainty that each cluster contains generation and load, a two-stage clustering method was adopted. The general process of a two-stage method is as follows: A graph-clustering algorithm is applied to a desired network graph with the algorithm's stoppage criterion being set to optimal modularity. Once the algorithm has computed a community structure, the structure's topology converges. A converged community structure essentially treats the output memberships of a graph-clustering algorithm as new graph vertices. As a simple example, Figure 3 contains nine vertices. A graph-clustering algorithm is applied, resulting in three microgrids as shown by the three clusters in the first-stage method.

A converged graph of this community structure assumes that each community of vertices' output from a graph-clustering algorithm is a single vertex in a new representative graph. Additionally, the edges between communities are the only edges considered in a converged graph. Figure 3 contains a flow chart that describes the two-stage process. The application of the first clustering algorithm results in a 3-microgrid system by clustering the 9-bus system. From this converged graph, an additional graph-clustering algorithm is applied, thus becoming a two-stage method. A second-stage algorithm is applied to the converged clusters from the first-stage method, resulting in a

final cluster formation that only contains two clusters that represent and fully contain the original nine vertices.

Deploying a two-stage method allows for important and desired results to be achieved. One consequence of a two-stage method is that the overall number of clusters can be reduced when the system is large. Another important characteristic of two-stage clustering is that desirable attributes of multiple clustering algorithms can be considered in a single clustering scheme. As an example, in this thesis, an important combination of nearest-generator clustering and betweenness-centrality clustering are used in a two-stage method.

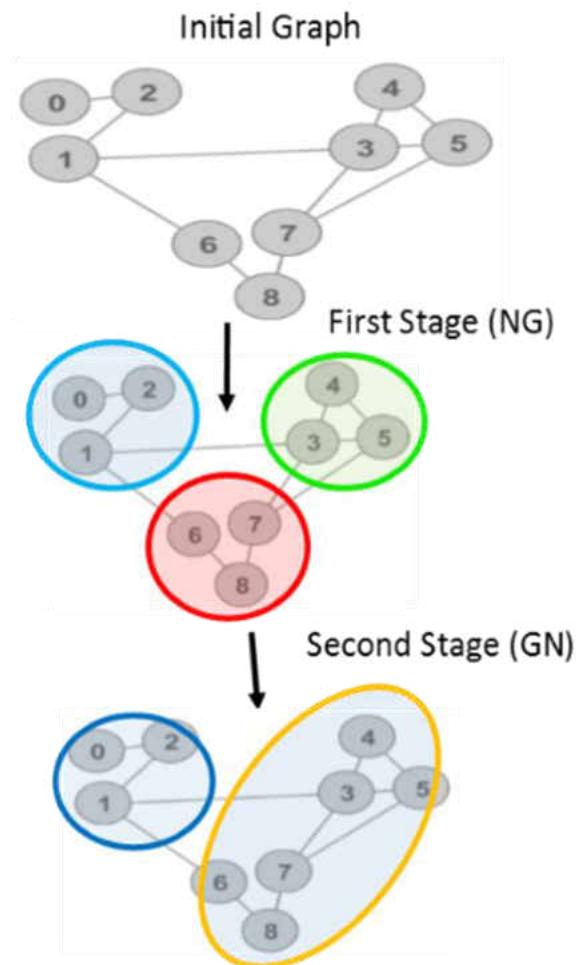


FIGURE 6. TWO STAGE GRAPH CLUSTERING PROCESS VISUALIZATION

## 2.6 R Software and Programming Language

R is an open source software programming language. The R environment is an integrated suite for calculation, graphical display, and data manipulation [66]. R was initially written by Robert Gentleman and Ross Ihaka of the statistics department at the University of Auckland. Today, R is a result of contributions from users all over the world. There are thousands of user-developed packages available with countless functions and capabilities available on the Comprehensive R Archive Network (CRAN). R was used extensively in this thesis work. Packages for forecasting, clustering, statistical analysis, and graph theory were utilized to conduct analysis and simulations for the work of this thesis. R code for each part is provided in the appendices.

## 2.7 Electric vehicle (EV) considerations for power grid

As popularity of EVs grows considerations must be made for accommodating the charging of these vehicles. Electric vehicles rely on the power grid to charge their batteries. Plug in hybrid electric vehicles are vehicles that have a combination of a combustion engine and a battery to provide power, thus the name “hybrid.” These vehicles are less reliant on the grid as they tend to have smaller powered batteries as compared to full EVs. EV’s can be a relatively large load in the electricity grid. If the charging is unmanaged it can be affect the electric grid negatively [67]. Some EV batteries can exceed 100kWh in size and are a considerable load to the grid when plugged in for charging. Uncoordinated charging of many of these vehicles could cause negative effects including transformer overload, harmonic distortion, and increased voltage deviation of the power system. Therefore, it is essential to better understand the impact of electric

vehicles on the grid [67]. Table X shows all the EVs and PHEVs available for purchase in the USA in 2018 and their associated battery specifications.

**TABLE 2. BATTERY SPECIFICATIONS FOR EVS AND PHEVs AVAILABLE IN 2018**

<b>Brand</b>	<b>Model</b>	<b>Battery kWh</b>	<b>Peak power kW</b>	<b>Peak Power hp</b>
Audi	A3 Sportback	8.8	75	150
BMW	330e	7.6	65	180
BMW	530e	9.4	70	184
BMW	530e	9.4	70	184
BMW	740e	9.2	80	255
BMW	i3	21.6	125	
BMW	i3	33.2	125	
BMW	i3 Rex	33.2	125	34
BMW	i8	7.1	96	231
BMW	X5	9	80	240
Cadillac	CT6	18.4	149	335
Chevrolet	Bolt EV	60	150	
Chevrolet	Volt	18.4	111	101
Chrysler	Pacifica	16		248
Fiat	500e	24	83	

Ford	C-Max Energi	7.6	88	141
Ford	Focus Electric	33.5	107	
Ford	Fusion Energi	7.6	88	141
Honda	Clarity	25.5	120	
Honda	Clarity PI	17	135	
Hyundai	IONIQ	28	88	
Hyundai	Sonata	9.8	50	154
Karma	Revero	21.4	301	260
Kia	Optima	9.8	50	154
Kia	Soul EV-e	27	81.4	
Kia	Soul EV	34	81.4	
Mercedes	B-Class	36	132	
Mercedes	C350e	6.2	60	241
Mercedes	GLE550e	8.8	85	329
Mercedes	S550e	8.7	80	329
MINI	Cooper SE	7.6	65	136
Nissan	Leaf	30	80	
Nissan	Leaf40	40	110	

Porsche	Cayenne	10.8	70	333
Porsche	Panamera	14.1	100	330
Porsche	Panamera Turbo	14.14	100	550
Smart	fortwo	17.6	60	
Tesla	Model 3			
Tesla	Model 3 LR			
Tesla	Model S 75	75	235	
Tesla	Model S 75D	75		
Tesla	Model S 100D	100		
Tesla	Model S P100DL	100		
Tesla	Model X 75	75		
Tesla	Model X 100D	100		
Tesla	Model X P100DL	100		
Toyota	Prius Prime	8.8	68	
Volkswagen	e-Golf	35.8	100	
Volkswagen	e-Golf SE	24.2	85	
Volvo	XC60	10.4		
Volvo	XC90	9.2	64	

Environmental concerns, security and supply of oil, and the increased use of intermittent renewable electric power sources in power grids are all factors that are increasing the focus on plugin hybrid electric-vehicles (PHEV) and Electrical vehicles (EV). Both PHEV's and EV's can assist in shifting the personal transportation sector away from fossil fuels and in providing balancing services to the electricity grid. EV's and PHEV's have potential to reduce greenhouse gas emission and thereby contribute towards improvement of global warming and hence many researchers are working on integration of this technology into the grid.

Global sales of electric vehicles for the year 2017 through August were over 649,000 units, 46% higher than the same period of 2016 [68]. In the Unites States of America, the 2017 3<sup>rd</sup> quarter finished with a sales increase of 30% compared to the same period of 2016. Over 142,000 plug-in vehicles have been delivered so far, and 62% of them are pure electric vehicles. The plugin share of the total light vehicles market is now, 1.1% compared to 0.9% in 2016 [68]. The charging of PHEV's and EV's can be a relatively large load in the electricity grid. If the charging is unmanaged it can be affect the electric

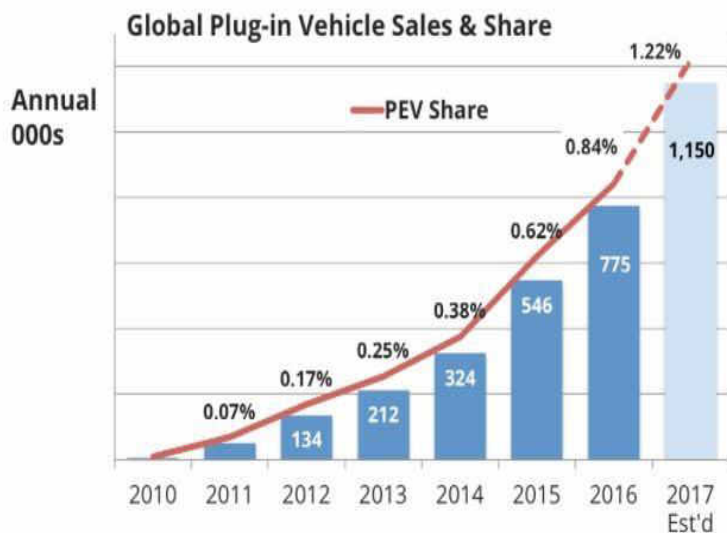


FIGURE 7. GLOBAL POPULARITY GROWTH OF ELECTRIC VEHICLES



grid negatively. These impacts include transformer overload, harmonic distortion and increased voltage deviation. Therefore, it is essential to study the impact of electric vehicles on the grid [67]. Bar graph of global plug-in vehicles sales& shares for years 2010-17 is shown in figure 7. In the presence of new technologies such as smart meters, renewable energy sources, distributed generation, and integration of electric vehicles, new paradigms and methods for load forecasting need to be developed. This work proposes a paradigm using time-series clustering given available smart meter data for households with EVs.

## Chapter 3. Clustering Analytics for Streaming PMU Datasets

### 3.1. Summary

This chapter aligns directly on the topic of grid situational awareness, anomaly detection, and algorithm development for WAMS software. This chapter analyzes the efficacy of clustering algorithms applied to streaming PMU phasor data (voltage, current, and frequency). The ability to accurately and efficiently cluster streaming phasor data allows for real-time detection and classification of grid anomalies. Existing clustering algorithms (k-means, k-medians, DBSCAN, and h-clust) were compared, and the efficacy of each algorithm in clustering anomalous data was analyzed. The clustering algorithms were implemented using R. The utility and effectiveness of hierarchical clustering (hclust) for anomaly detection in phasor measurement unit (PMU) datasets was demonstrated by comparing it against other well-known clustering algorithms. Hclust showed an increase in anomaly detection efficiency according to Dunn Index (DI) and improved upon run-times of well-known techniques such as Density Based Spatial Clustering of Applications with Noise (DBSCAN).

### 3.2 Methods

#### 3.2.1. Background

Situational awareness of modern power systems is becoming increasingly important as the complexity of grid systems grow [69]. Wide Area Management Systems (WAMS) are being developed by upgrading the existing power grids to enhance the abilities of the grids. Synchrophasors are units that can measure various parameters such as voltage, current, and frequency of the lines at a sampling rate of 30 to 120 samples per second [70]. These synchrophasors play a vital role in managing the WAMS because the system can be managed only if the operators know the status of the grid. The time-

tagged measurements from the synchrophasors can be used for many power system applications such as State Estimation (SE) [71]–[73], Load Forecasting (LF) [74], fault detection, micro- grid operations [75]–[77], etc. Using synchrophasor data, a voltage stability assessment technique has been proposed in [78]. An algorithm has been developed to detect and locate the faults on the transmission lines using the phasor data in [79].

A remote terminal unit (RTU) or supervisory control and data acquisition (SCADA) system can provide around 30 samples for 5 minutes, while the same number of samples is provided by the synchrophasor in one second at its slowest sampling rate. The difference in data frequency between traditional SCADA technologies is critically important to situational awareness. With a higher volume of data, more informative analysis of grid operation can be made. Even though synchrophasors provide power system information at a large sampling rate, they can be useful only if the operators can utilize the data to make decisions or manage the system.

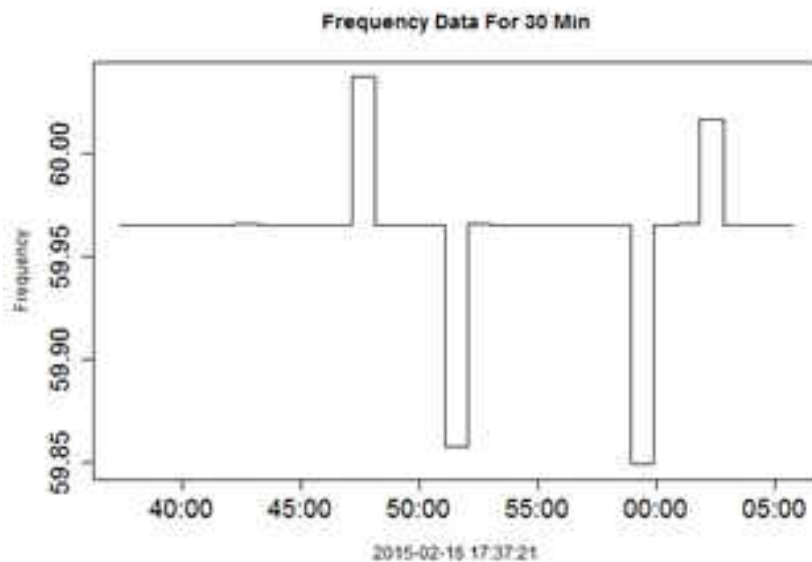
In [11] a software framework is proposed that makes use of data clustering methods to provide system operators with enhanced situational awareness. This work is a foundational work related to the work proposed by this chapter. The authors in [11] identify DBSCAN as an effective algorithm to detect anomalies in PMU datasets. This chapter expands on this work by comparing DBSCAN with other known clustering algorithms to clearly identify which algorithm is most appropriate for PMU data.

### 3.2.2 Streaming Phasor Datasets

The datasets used for clustering analysis contain 10 minutes of streaming voltage, current, and frequency data from a PMU operated by the Tennessee Valley Authority

obtained through open phasor data concentrator (openPDC). The openPDC synchrophasor collects data at a rate of 30 samples per second. This corresponds to roughly 18,00 data points for observation in just 10 minutes of operation.

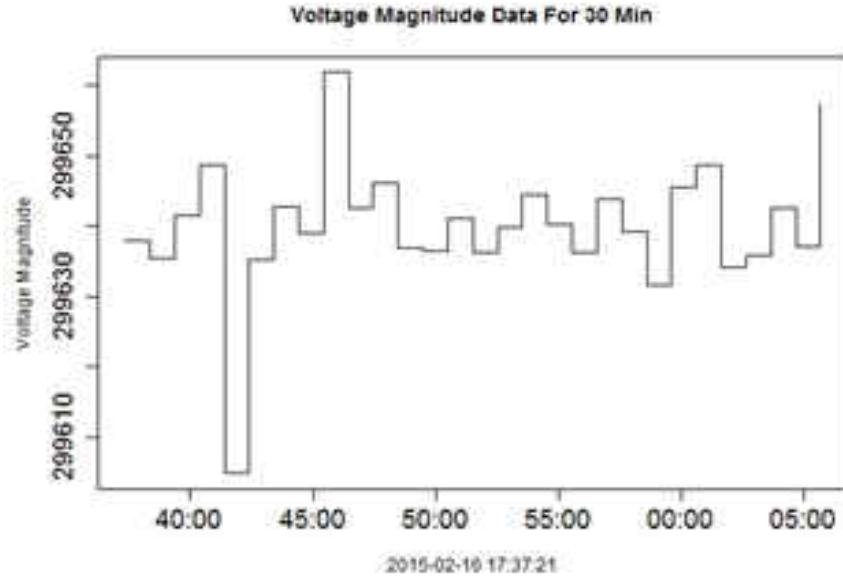
Figure 1 shows a stream of 30 minutes of phasor frequency data with randomly inserted faults as a plot of 1 minute moving averages. Although 30 minutes of data are shown in figure 8, datasets were trimmed to 10 minutes for clustering analysis due to computation time and CPU storage constraints. In each case 100 anomalous data points were inserted using the same random insertion protocol previously described. The timestamps containing anomalous data points are observable as they deviate significantly relative to a constant stream of frequency during the time interval shown in figure 8.



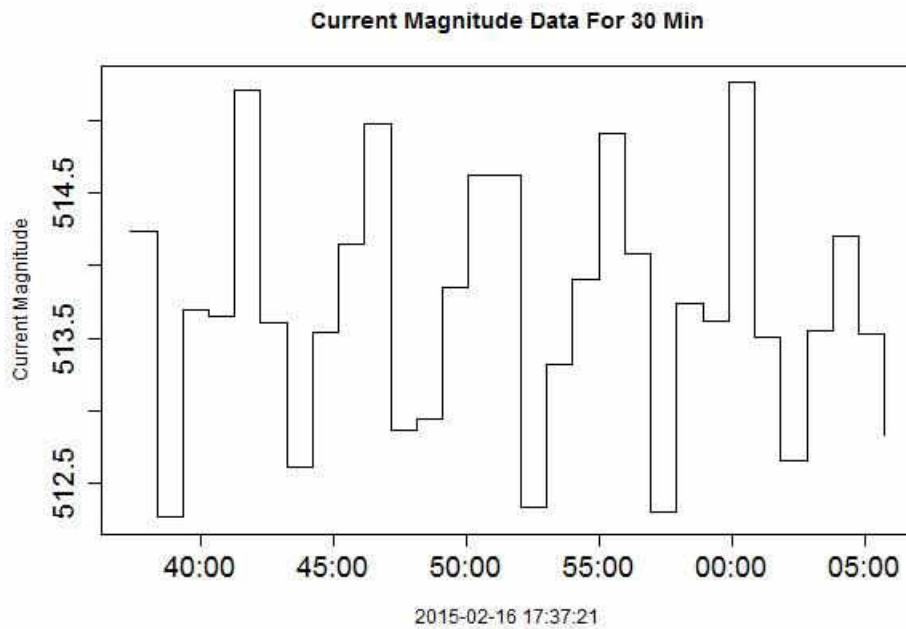
**FIGURE 8. STREAMING FREQUENCY DATA WITH ANOMALIES RANDOMLY INSERTED**

Similarly, figure 9 shows anomaly inserted voltage magnitude data, and figure 10 shows anomaly inserted current magnitude data. These data are manipulated data from an actual PMU. Voltage and current magnitudes are more variable than frequency in a

transmission, so the anomalies inserted in frequency data are more noticeable by inspection than in the voltage and current data.



**FIGURE 9. ANOMALIES INSERTED INTO STREAMING VOLTAGE MAGNITUDE DATA FROM PMU**



**FIGURE 10. ANOMALIES RANDOMLY INSERTED INTO PMU CURRENT MAGNITUDE DATA.**

### 3.2.3. Methodology

To observe the ability of clustering algorithms to cluster and detect problematic data, fault data were randomly inserted into the datasets. An insertion of 100 anomalous data points was placed in 1 to 4 randomly selected segments of each of the streaming datasets. The 100 points are roughly equivalent to 3 seconds of data. To each type (voltage magnitude, current magnitude, and frequency) of streaming phasor data, each of the clustering algorithms (k-means, k-medians, DBSCAN, and hclust) were applied. For algorithms where number of clusters to output was user-defined, a maximization of DI was used to determine optimal number of clusters. In the case of hclust, cutoff values were also chosen based upon a built-in optimization of DI. For DBSCAN, a trial and error approach was used to observe the values of *eps* and *minPoints* that maximized Dunn's index.

Then, for each algorithm on each data type, the DI was computed to quantitatively measure the efficiency of the clustering scheme on that data relative to the other algorithms. Additionally, the run times of each algorithm on each data type were analyzed. All parameters of each algorithm on each type of data were compared to evaluate which algorithm performed clustering of PMU data most efficiently.

### 3.3 Results

Results for the chapter entitled "Clustering Analytics in Streaming PMU Datasets" are discussed in this subchapter. Clustering schemes were applied to frequency, current, and voltage and compared by analyzing DI and run-times of each scheme.

### 3.3.1. Streaming frequency data

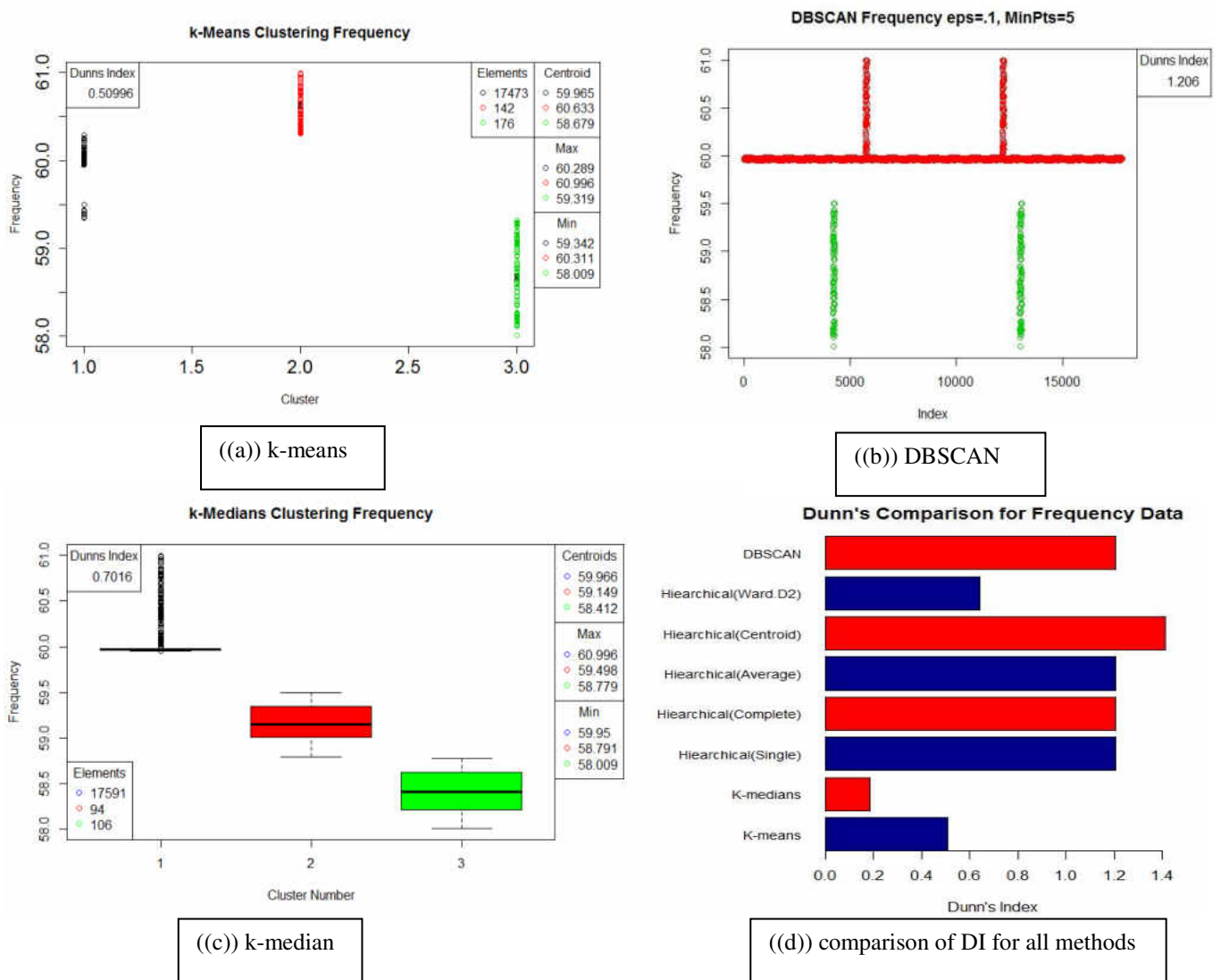


FIGURE 11. K-MEANS, DBSCAN, AND K-MEDIAN ALGORITHMS APPLIED TO 10 MINUTES STREAMING FREQUENCY DATA

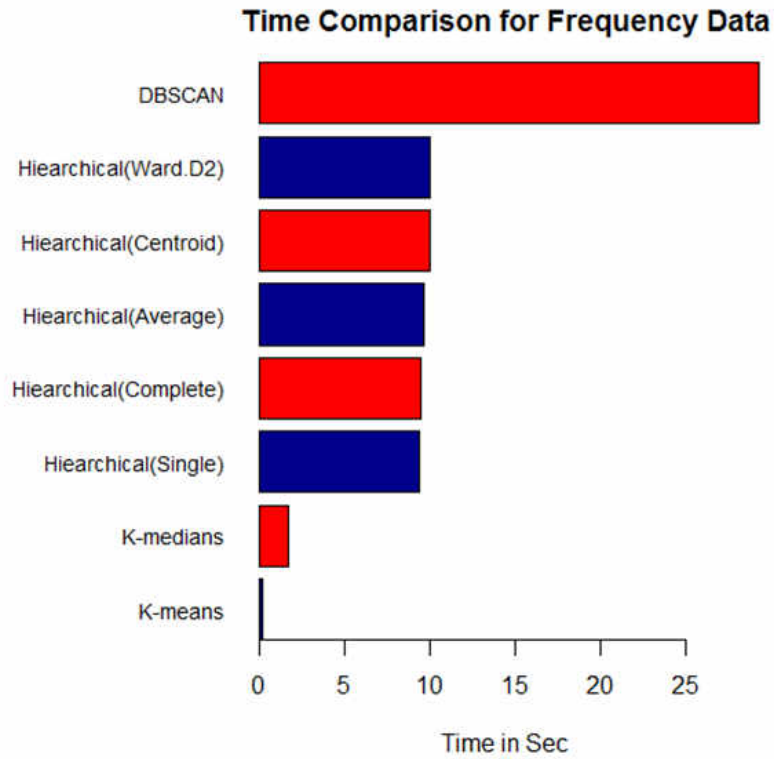


FIGURE 12. RUN-TIME COMPARISON FOR CLUSTERING ALGORITHMS ON FREQUENCY DATA

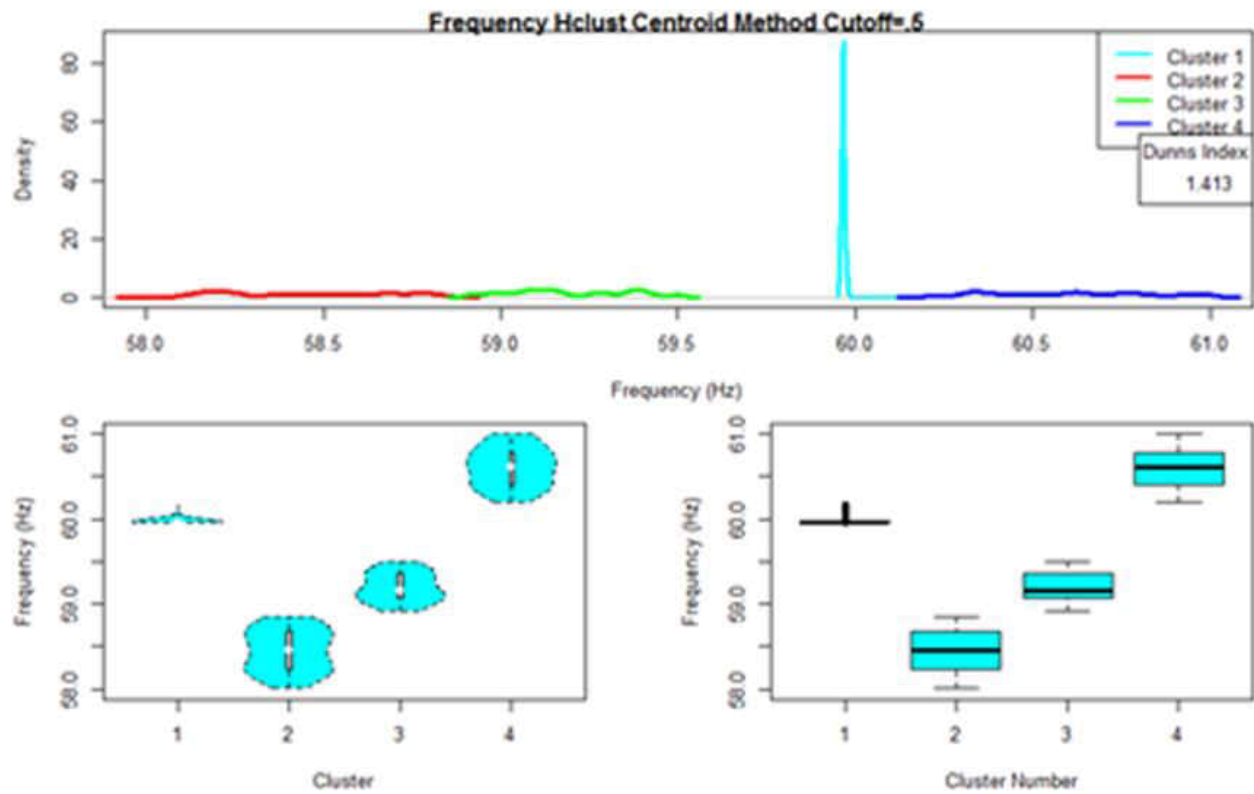


FIGURE 13. HCLUST CENTROID LINKAGE DENSITY, VIOLIN, AND BOX PLOTS



Figure 11 displays visualizations of the clustering schemes of streaming frequency data as well as a bar chart comparing the DI of each method. Figure 13 shows hclust centroid method density, violin, and box plots to better understand the distribution of data points in each cluster. This specific cluster was analyzed because it showed the highest value of DI. Figure 12 displays a bar chart of the run time required for each algorithm.

### 3.3.2. Streaming Voltage Data

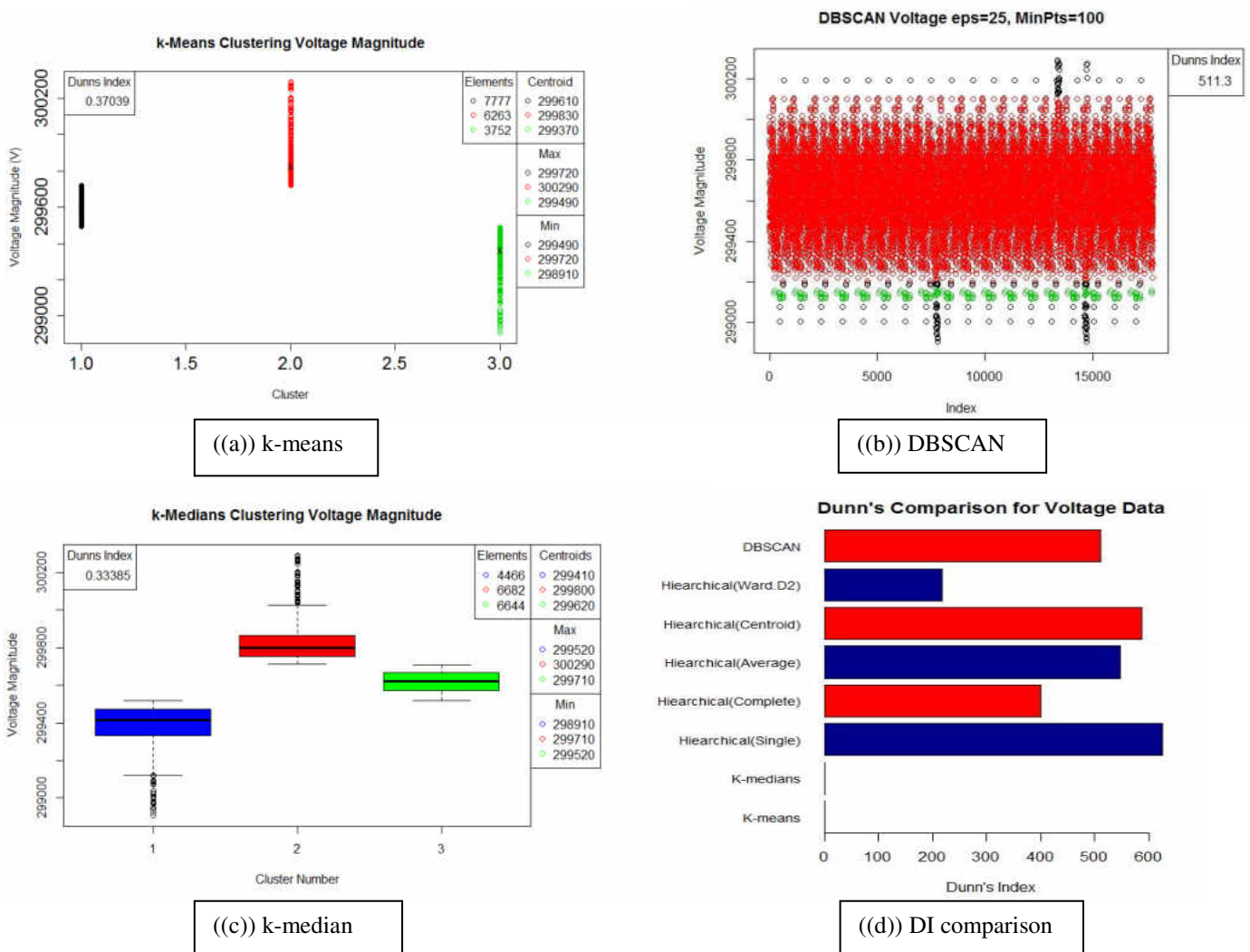


FIGURE 14. K-MEANS, DBSCAN, AND K-MEDIANS AND DI COMPARISON FOR VOLTAGE DATA

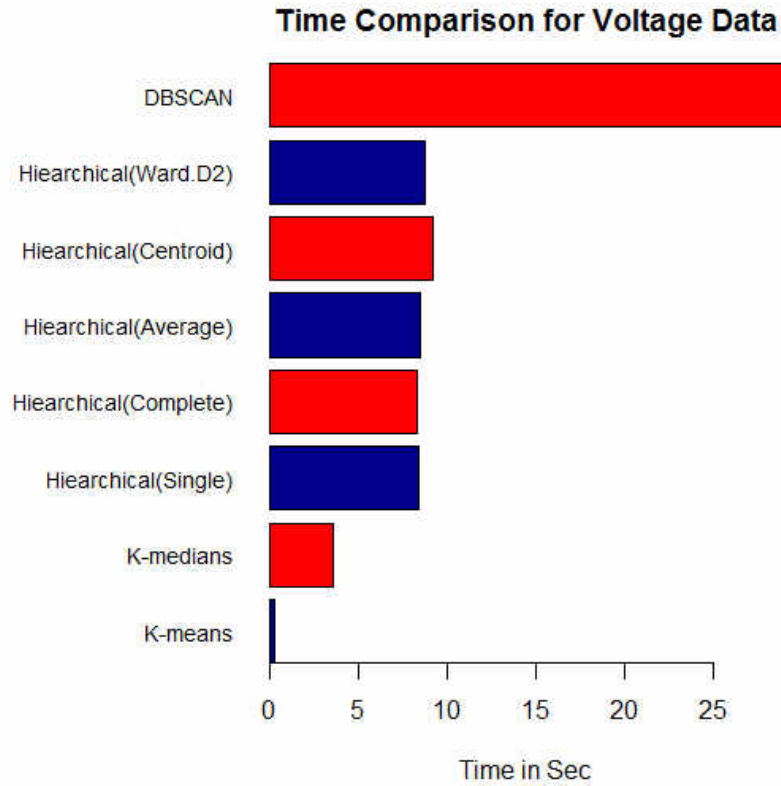


FIGURE 15. RUN TIME COMPARISON FOR THE ALGORITHMS APPLIED TO STREAMING VOLTAGE DATA

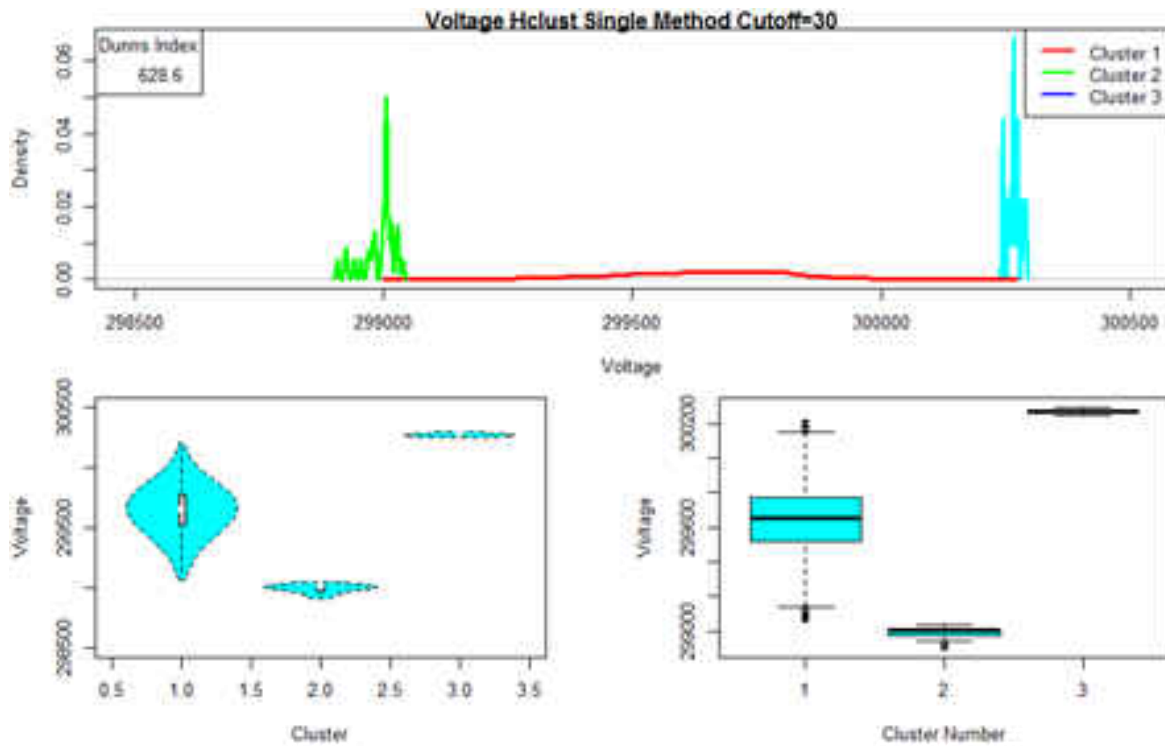


FIGURE 16. HCLUST CENTROID LINKAGE METHOD: DENSITY PLOT, VIOLIN PLOT, BOXPLOT

Figure 14 displays visualizations of the clustering schemes of streaming voltage data as well as a bar chart comparing the DI of each method. Figure 16 shows hclust centroid method density, violin, and box plots to better understand the distribution of data points in each cluster. This specific cluster was analyzed because it showed the highest value of DI. Figure 15 displays a bar chart of the run time required for each algorithm.

### 3.3.3 Streaming Current Data

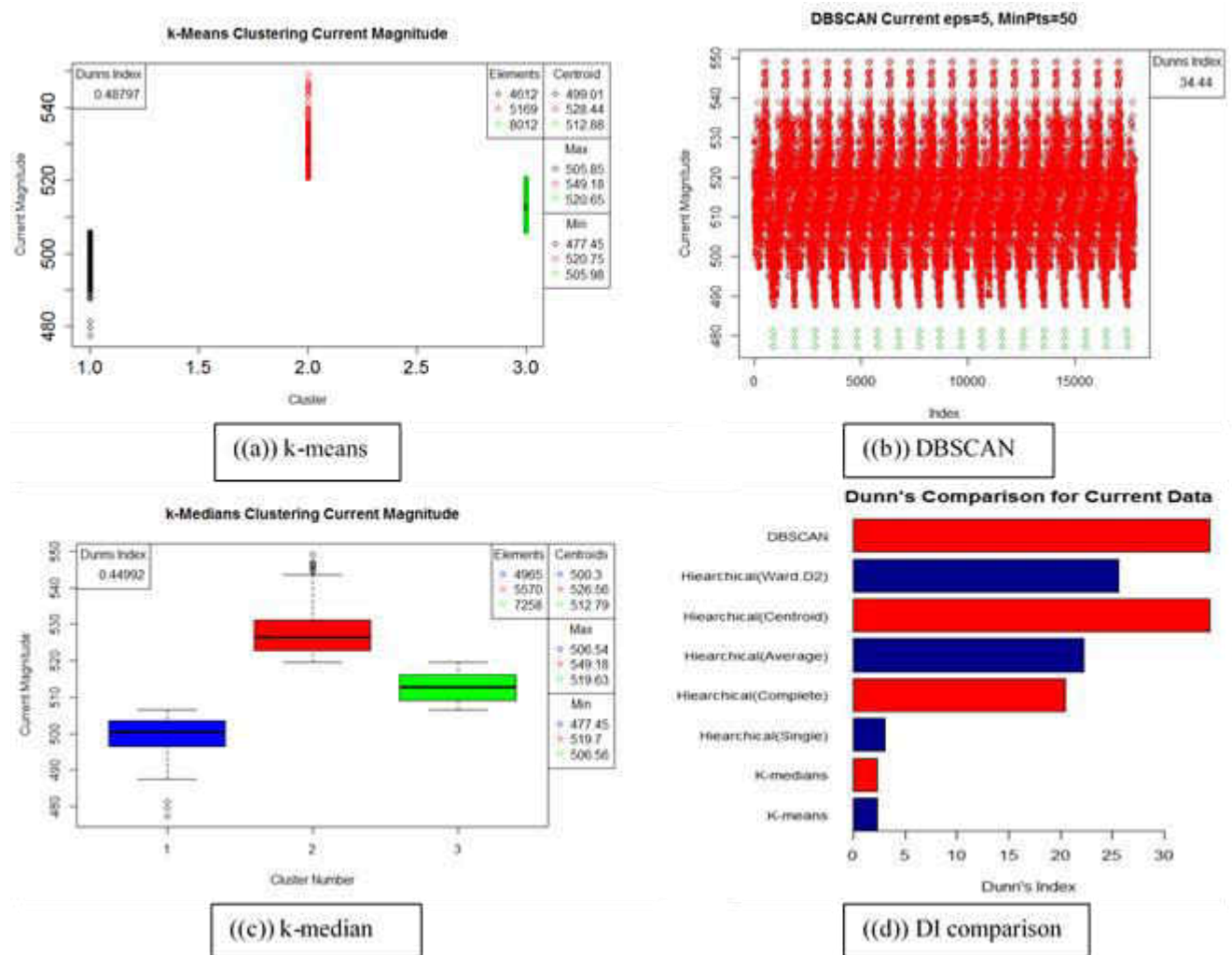


FIGURE 17 A) K-MEANS, B) DBSCAN, C) K-MEDIANS AND D) DI COMPARISON

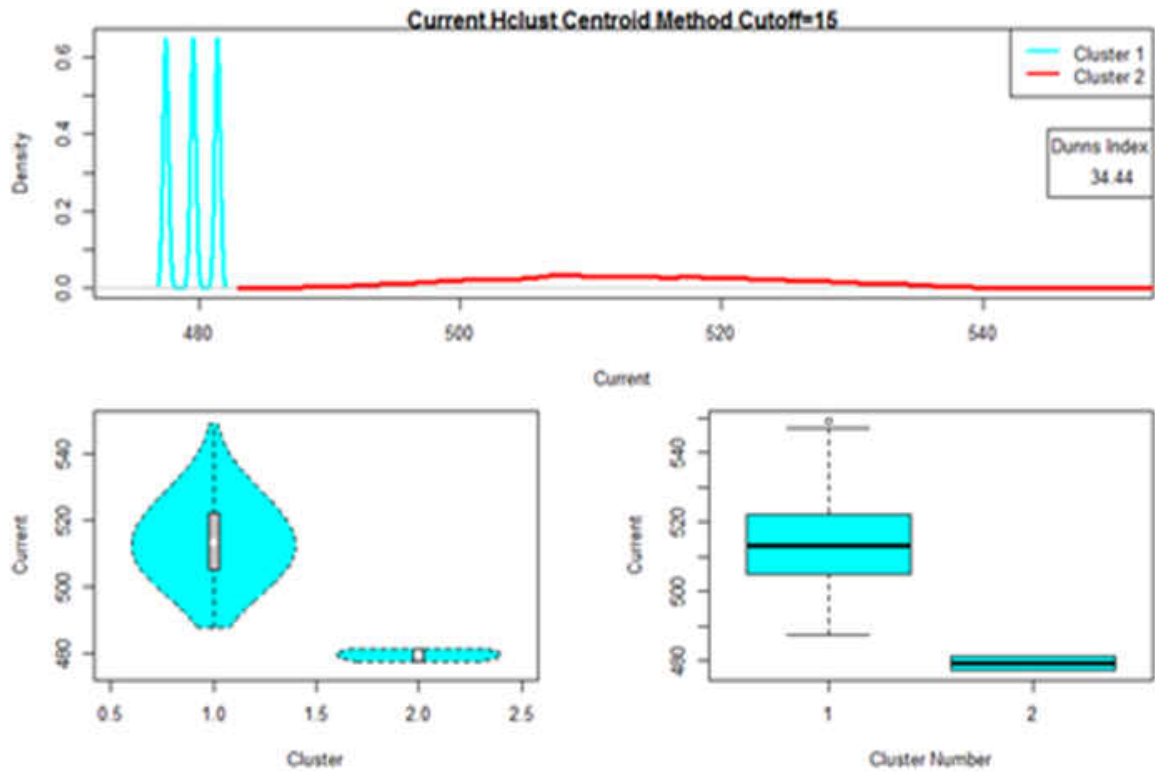


FIGURE 18. HCLUST CENTROID METHOD: DENSITY PLOT, VIOLIN PLOT, BOX PLOT

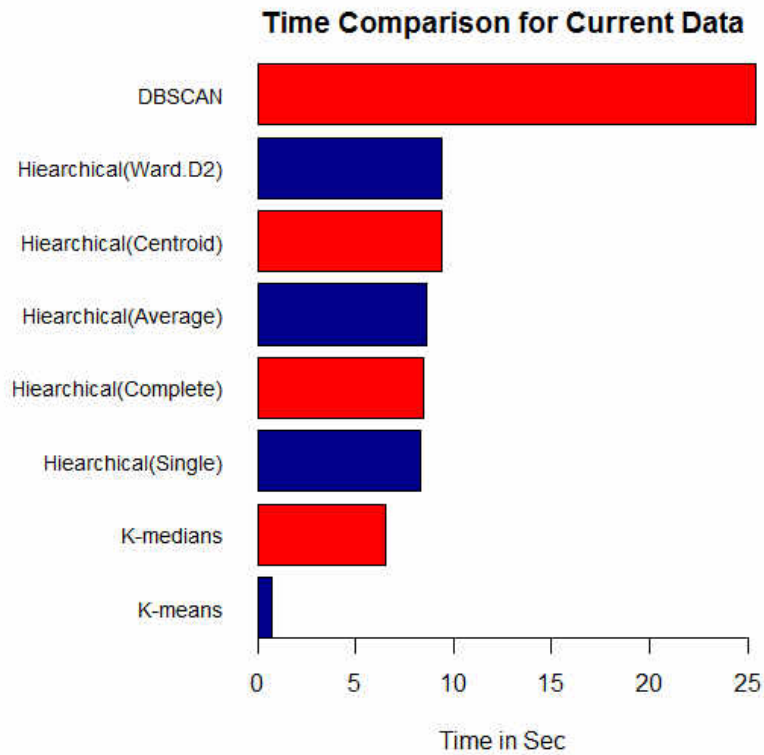


FIGURE 19. RUN TIME COMPARISON FOR CURRENT MAGNITUDE DATA

Figure 17 displays visualizations of the clustering schemes of streaming current data as well as a bar chart comparing the DI of each method. Figure 18 shows hclust centroid method density, violin, and box plots to better understand the distribution of data points in each cluster. This specific cluster was analyzed because it showed the highest value of DI. Figure 19 displays a bar chart of the run time required for each algorithm.

### 3.4 Interpretation and Discussion

For frequency data, hclust centroid method performed efficiently. This algorithm yielded the highest Dunn's index of 1.413. Hclust Ward's method yielded an index of 0.6418 while single, complete, and average methods each yielded indices of 1.206 but the box, violin, and density plots of these methods are not shown to avoid redundancy. Hclust, regardless of linkage method generally performed more efficiently than k-means and k-median algorithms.

Hclust single linkage performed clustering of voltage data very efficiently. This algorithm yielded the highest Dunn's index of 626.6. Centroid method yielded an index of 587.1 while single, complete, and average methods each yielded indices between 218.6 and 399.8. Density plot shows peaks of anomalous voltages at 299kV and 300.25kV. For voltage data, hclust generally performed more efficiently than k-means and k-medians.

Hclust single centroid again performed clustering of data efficiently. For current data, this algorithm yielded the highest Dunn's index of 34.44. DBSCAN also yielded an index of 34.44, while hclust single, complete, and average methods each yielded indices between 20.48 and 25.59. Hclust again performed more efficiently than k-means and k-medians regardless of linkage method. Density plot shows peaks of anomalous current around 480A. Violin plot shows an approximately normal distribution for current data

representing currents under normal operating conditions. A cluster with another approximately normal distribution segmented current values with low current magnitudes representing abnormal operation or low current conditions.

Computation run time of each algorithm were compared to examine their feasibility for real-time use. The computation time can play a role for feasible use for larger datasets. They also play a role when real-time decisions need to be made by system operators. Ideally, computation run time should be small enough that immediate responses can be coordinated to grid anomalies. If run-times are excessive there will be a delay in any decision-making response.

The computation run times for the three types of data were nearly identical. The deviation in computation time for a given algorithm often differs less than  $\pm 10\%$  between the three parameters. DBSCAN consistently shows a larger computation run time. Hclust computation run times were consistently about one third of the DBSCAN computation times at approximately 10 seconds. The k-means and k-medians algorithms consistently performed in 5 seconds or less. For computation run times, k-means clustering consistently performs very quickly relative to the other algorithms. It often performed clustering in  $\leq 3$  seconds. Hierarchical clustering, although it is slower in computation time than k-means, was consistently efficient. The time to perform hclust was not of concern for openPDC datasets of 10 minutes as the algorithm usually performed within 10 to 20 seconds. There is no concern for hierarchical clustering computation or RAM storage capabilities for openPDC datasets containing  $\leq 10$  minutes. Run time computations were conducted on a desktop computer with an Intel i5-4670k processor, 16GB RAM, and a z87-g41 MSI pc-mate motherboard. Capabilities of this system were exceeded when

attempting clustering on datasets containing 30 minutes of data. The system used for these computations is a capable system. It is significant to note that just 30 minutes of data from one PMU causes computational issues with this system. In order to implement these computational techniques, considerable processing systems are desirable.

### 3.5 Conclusions

This chapter introduced the application of clustering complex phasor data for openPDC datasets. k-Means, k-medians, DBSCAN, and hierarchical clustering algorithms were implemented using R statistical software. Distance metrics of hierarchical clustering were observed and consistently performed clustering more efficiently w.r.t Dunn's index than k-means, k-medians, and DBSCAN clustering algorithms. In particular, it was observed that centroid hclust performed efficiently for frequency and current magnitude data. The single-link metric of hclust performed most efficiently for voltage data. At the expense of optimizing Dunn's Index, it is possible that some compromises are made in terms of the representation of the data. For certain data, a smaller number of clusters results in a higher Dunn's Index. While this may be clustered efficiently, there may be some information to be gleaned from a larger number of clusters as certain types of data activity may be captured and highlighted more distinctly. In addition, the current method of selecting hclust cutoff values using trial and error can be taxing, inefficient, and possibly inaccurate.

Experimental results on parameters such as frequency, voltage, and current demonstrate the novelty and effectiveness of the application of hierarchical clustering. The results indicate that the hierarchical clustering with single linkage distance metric

is a good choice for sudden surge or sag values. On the other hand, the average distance metric is less sensitive to outliers and can detect small deviations in parameters.

Overall, hierarchical clustering is an efficient and effective set of algorithms for analyzing streaming phasor data. Dunn's indices consistently show efficient clustering performance and computation run times are feasible for practical use. A scheme that incorporates the use of hclust algorithms is recommended for application to real-time smart grid situational awareness to aid in anomaly detection and decision-making protocols. This will aid system operators in both detecting and troubleshooting potential issues in the grid.



## Chapter 4. Betweenness Centrality-based Identification of Critical Buses and Decomposition of Microgrids in IEEE Test-Bus Systems

### 4.1. Summary

The ability to identify critical structures, groups of buses, or critical hardware components is a key topic for power systems management. Applications of identification of critical components aids decision making with reference to maintenance, operations, and planning. This chapter proposes BC-based methods for critical component analysis in power systems. Specifically, BC is identified as a critical metric to identify individual buses that are important to transmission through a grid. This identification is extended a modified Girvan-Newman (GN) based BCGC algorithm to identify microgrid cluster formations from within smart grid networks. Methods proposed in this work use concepts from graph theory and network theory to model clusters of microgrids. Modules of smart grid are modeled using graphs with vertices and edges representing buses and transmission lines respectively. Specifically, load, batteries, generator, and relay buses were represented by graph vertices and transmissions between them considered as graph edges. Metrics of determining critical buses were analyzed based upon multiple criteria. BC was demonstrated to be effective in determining critical buses as well as defining community structures for microgrid determination within a larger scale power system.

### 4.2 Methods

#### 4.2.1. Background

A power grid can be decomposed into regions or areas using partitioning, splitting, and clustering methods informed from analysis provided by graph theory. Determination of regional community structures within a smart grid is an important task for optimal

management of the resources. The decomposition of power system is not a novel concept, but there is very limited research on how to decompose a grid or how to evaluate the effect of decomposition of micro grids for economic dispatch. Similar concepts date back to the 1950s [80]. The earliest work involving grid decomposition focused on the development methods for breaking large systems into smaller subsystems in order to make complex analysis or computations simpler [80].

More recent works have focused on identifying power-network zones within a grid [17], spectral clustering of power grids [15], and assessing grid reliability based on topological metrics [81]. In [15], hierarchical spectral-clustering methods were used for power-grid decomposition, and [17] used electrical distance quantification as a parameter for dividing a bus system into microgrid-like zones. To our knowledge, there is no paper that has considered an approach similar to the one proposed by this chapter. The metrics of betweenness centrality (BC) and two-stage clustering brings novelty to grid decomposition approaches. An efficient grid decomposition and microgrid utilization has become important in the 21<sup>st</sup> century, as smart grid technologies evolves. Optimal grid decomposition will play an important role in uncertainty quantification, contingency planning, resource allocation, optimal power flow, cascading failure protection, integration of renewable power sources to the next-generation smart grid [10].

#### 4.2.2 Identification of Critical Nodes

This work follows an analytical procedure similar to the method proposed in a highly regarded computational science work found in [82], but applies to concept to analysis of a power system. First, the test bus system was modeled using graph theory concepts. Next, a variety of indices were formulated. These indices were coined as critical

bus indices (CBI) that attempted to quantify the importance of buses in the test system. CBIs were formulated using indices that arise from topological analysis combined with functional information of power systems. The CBIs were formulated and examined to see which indices best provided meaningful information about the criticality of individual buses in a system relative to the other buses. Each of the indices were formulated such that the larger the index, the more critical the bus is according to that index. The indices are explained in table X.

11 indices were formulated. One index was simply assigning the degree ( $D$ ) of the bus as an index. Another index that was formulated ( $B$ ) was the normalized impedance weighted BC. Another index was normalized BC multiplied by the demand ( $NB_d$ ) of each bus. A fourth index was a normalized BC multiplied by degree multiplied by demand of each bus ( $NDB_d$ ). Finally, the last index was random, where buses were randomly assigned for removal.

#### 4.2.3 Node Removal Methodology and Normalized Expected Impedance Distance

To evaluate how effective these indices are in determining the importance of the buses, bus removal in descending order of each index was performed. Indices can be created with many metrics, but it is important that an index quantifies something meaningful. Bus removal analysis simulates the topological disruption that is caused to a graph by removing buses from the system and examining the effect that the removal has on the connectivity of the system. By comparing the removal of buses in descending order of each index to random removal of buses, the quantification of importance of each index can be analyzed. Node removal analysis is a method to determine which types of indices provide topological meaning to a connected system.

To quantify the disruptivity of removing a bus, analysis of normalized expected geodesic distance (NEGD) and normalized expected impedance distance (NEID) was conducted. In both cases the disruptivity of the 5 removal indices was compared against a removal of random buses in random order. NEGD is defined in [82] and NEID is directly related to it. NEGD is a metric that quantifies the connective distances of nodes in a graph. It is the average geodesic distance that would be expected to be traveled through when traveling from node  $i$  to node  $j$  in a graph. NEGD is given by (8). NEID is similar to NEGD, except distance between nodes (buses) is defined by the impedance of the edges (transmission lines) between the buses instead of defining distance geodesically. The equation for NEID is given by (9). To compare indices, normalization was conducted. The equation for normalization is given by (10).

**( 8 ) NORMALIZED EXPECTED GEODESIC DISTANCE**

$$NEGD = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{v_i, v_j}}{\frac{n(n-1) * E}{2}}$$

**( 9 ) NORMALIZED EXPECTED IMPEDANCE DISTANCE**

$$NEID = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n Z_{v_i, v_j}}{\frac{n(n-1) * E}{2}}$$

**( 10 ) NORMALIZATION OF CBI**

$$NCBI_i = \frac{CBI_i - \min(CBI)}{\max(CBI) - \min(CBI)}$$

In equations 8-10,  $n$  is the number of buses in the system,  $d_{v_i, v_j}$  is the geodesic distance between nodes  $i$  and  $j$ ,  $E$  is the eccentricity [83] (largest geodesic distance in the graph),  $Z_{v_i, v_j}$  is the impedance in the transmission line between nodes  $i$  and  $j$ . CBI is the

critical bus index being normalized (B, Bd, D, etc..), and  $NCBI_i$  is the normalized CBI of bus  $i$  following a standard normalization formula.

After node removal analysis for all indices was conducted, the most effective indices for quantifying criticality of buses were determined. After the most effective index was determined, strategies that employ this index were developed to decompose the test systems into microgrids. For the decomposition of a large system into smaller scale network communities or microgrids to be effective, the buses with highest importance should be given special consideration in decomposition. Based on the node removal CBI analysis, graph clustering algorithms based upon strong indices were deployed for system decomposition. Ultimately betweenness centrality was a demonstratively effective index.

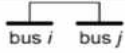



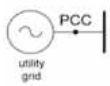
#### 4.2.4 Power System Decomposition and Microgrid Specifications

This chapter aligns on the topic of using graph clustering algorithms to determine microgrid like structures within a power system. Large scale grid decomposition is not always necessary in determining microgrids because microgrids are often built independently of a large-scale grid system. However, should the situation arise, where the desire to logically decompose a large power grid into microgrids is necessary, this work demonstrates the utility of graph clustering algorithms for the specific purpose of defining these microgrids, which can more loosely be called power zones. Graph clustering algorithms can be applied to any network graph. However, because the network graphs represent power system bus systems, specific considerations must be made. When applying graph clustering techniques to IEEE bus systems special attention was paid to whether the results of the clustering algorithms formed logical clusters per

power systems micro grid definitions. The definition of a micro grid varies somewhat depending upon the utility. For this work, a formal definition of a micro grid followed from the definition provided in [84].

This definition defined the “microgrid rules” (MGR’s) of a microgrid system by specifying the types of buses necessary to be contained in a micro grid as well as general power flow constraints. A table outlining the micro grid rules is found in table 2. I define the units of micro grid contains components listed in Table 3, where,  $P_i(t)$  = active power, injected from the bus into the grid (positive for generators, negative for loads);  $Q_i(t)$  = the reactive power, injected into the grid;  $V_i(t)$  = the voltage magnitude of the bus;  $\delta_i(t)$  = the phase angle of the voltage  $V_i$ . Buses are denoted with the running index,  $i$ . As outlined in table 3, a micro grid rule (MGR) was defined as a unit system containing at least one source of power generation, a non-zero load bus, and a bus containing power storage capability. When graph clustering algorithms were applied to the bus systems, special attention was paid to whether the grid decomposition formations of the clustering scheme followed the MGR’s. The specification on generator type is important in the utility of a micro grid.

**TABLE 3. DEFINITION OF MGRS**

Unit	Symbol	Constraints
power line		Constraints: $I_{i,j,x}(t) < I_{i,j,max}$
load		Constraints: $P_{i,x}(t) = -P_{L,ix}(t)$ (fixed) $Q_{i,x}(t) = -Q_{L,ix}(t)$ (fixed) $V_{ix,min} \leq V_{ix}(t) \leq V_{ix,max}$ Free variable: $V_{ix}(t), \delta_{ix}(t)$
Generators (renewable or conventional)		Constraints: $P_{ix}(t) = +P_{g,ix}(t)$ (fixed), $Q_{i,x}(t) = \bar{q}(t)$ (fixed), $V_{ix,min} \leq V_i(t) \leq V_{ix,max}$ Free variable: $V_{ix}(t), \delta_{ix}(t)$
Storage device / relay		$E_{ix}(t)$ = stored energy or state of charge(SOC) Typical constraints: $V_{ix}(t) = V_{s,i}$ (fixed) $0 \leq E_i(t) \leq E_{i,max}$ $-P_{i,rated} \leq P_i(t) \leq +P_{i,rated}$ State equation (one phase) $\frac{d(E_i)}{dx} = f_i(P_i, E_i)$ Free variable: $P_{i(t)}, E_i(t), \delta_i(t), Q_i(t)$
Point of common coupling		The point of coupling is indexed as bus1,i=1 Constraints: $\delta_{ix}(t) = 0$ $V_{l,x}(t) = -V_{in,x}(t)$ (fixed) $P_{lx,min} \leq P_{l,x}(t) \leq P_{lx,max}$ $Q_{lx,min} \leq Q_{l,x}(t) \leq Q_{lx,max}$ Free variable, $P_l(t), Q_l(t)$

4.2.5 Economic Dispatch Formulation

To see the impact of decomposition structures on its cost, an economic-load dispatch model (ED) is applied to these IEEE test systems. ED is a method to schedule the generator outputs with respect to its load demands to operate the power system most

economically. In other words, the main objective is to allocate the optimal power generation of different units at the lowest possible cost while meeting all system constraints [19]. The economic-load dispatch is performed in a multi-generator system in order to schedule the generators to satisfy the loads in the system that are subjected to generator and transmission-line limits. In a power system, minimizing the operation cost is very important and therefore, ED was used as an effective way to evaluate the different clustering techniques.

The clustering techniques divide the bus system into different zones, or areas and applying economic dispatch to such a system is known as Multi-Area Economic Dispatch (MAED). The aim of MAED problems is to minimize the power-generation cost while satisfying the system's load demand subject to the generation and line-flow constraints. The fuel cost for generating unit  $i$  (in \$ per hour) to supply a  $P_{Gi}$  amount of real power can be represented by a quadratic equation [85] as shown in (11):

**( 11 ) REAL POWER GENERATION COST EQUATION**

$$F_i(P_{Gi}) = a_i P_{Gi}^2 + b_i P_{Gi} + c_i$$

where  $a_i$ ,  $b_i$ , and  $c_i$  are the cost coefficients of generating unit  $i$  and  $P_{Gi}$  is the real-power generation of unit  $i$ . The objective is to minimize the total generation cost, which can be represented by the following equation:

**( 12 ) GENERATION COST EQUATION TO MINIMIZE**

$$F = \sum_{i=1}^{n_g} F_i(P_{Gi})$$



where  $n_g$  is the number of generators working in the bus system. The economic-dispatch problem is then solved subject to several formulated constraints [86]. They are listed in equations (13), (14), (15) and (16).

**( 13 ) POWER GENERATOR CONSTRAINTS**

$$P_{Gi_{\min}} \leq P_{Gi} \leq P_{Gi_{\max}} \quad \text{for } i = 1 \dots n_g$$

**( 14 ) GENERATION TO DEMAND CONSTRAINT**

$$\sum_{i=1}^{n_g} P_{Gi} = D$$

Here, constraint equation (13) implies that the power production from each generator must be within its maximum and minimum values and constraint equation (14) shows the condition that the power production from all the generators should meet the system's total demand. The power flow through the tie lines that connect the areas is an additional constraint in the MAED problem, as shown in equation (15).

**( 15 ) TIE-LINE POWER FLOW CONSTRAINT**

$$T_{mn_{\min}} \leq T_{mn} \leq T_{mn_{\max}}$$

The power flow between two areas,  $m$  and  $n$ , is subjected to a minimum and a maximum value of  $T_{mn_{\min}}$  and  $T_{mn_{\max}}$ , respectively.

**( 16 ) AREA POWER FLOW CONSTRAINT**

$$\sum_{i=1}^{m_g} P_{Gi} - \sum_{j=1}^{t_c} T_{cj} + \sum_{k=1}^{t_c} T_{kc} = D_c$$

Equation (16) ensures that the loads in each zone are satisfied by the generation within the zone and from the neighboring zones. Here,  $m_g$  indicates the number of generators

within the zone,  $t_c$  is the number of tie lines connected to the zone, and  $T_{cj}$  and  $T_{kc}$  indicate the power flowing from and coming to the zone from connected zones because the power flow is bi-directional between the clusters. Variable  $D_c$  indicates the total active load for the microgrid under consideration. In this model, the cost of the power flow through the tie lines is also considered. A generation cost of \$0.1 per MW and a 200-MW maximum tie-line flow limit were used in [87]. To compare the different zones obtained by using various clustering techniques, the generators' cost functions are assumed to be the same for all generators in the grid system. The total cost function to be minimized is the sum of the generation cost and the cost of the tie-line power flow; the modified equation is given in (17).

**( 17 ) COST FUNCTION - OBJECTIVE FUNCTION OF LINEAR PROGRAMMING FORMULATION**

$$\min F = \sum_{i=1}^{n_g} F_i(P_{Gi}) + \sum_{j=1}^t C_j T_j$$

The variable  $C_j$  denotes the cost for the tie-line power flow, which is assumed to be constant for all the tie lines;  $t$  represents the number of tie lines in the model; and  $T_j$  is the amount of tie-line power flow [88]. The ED model is developed using the concept of linear programming with the mentioned load, generator, and tie-line flow constraints. The ED model is programmed using AMPL (Algebraic Mathematical Programming Language), a popular tool that is used to solve linear-programming problems. AMPL software needs two file types: model and data files. The .mod file contains the linear-programming code and the .dat file contains the system data. The .mod file will work on the data or information in the .dat file. Separate .dat and .mod files are created for the IEEE 118 and

IEEE 300-bus systems for every decomposition structure. The ED model only considers the system's active loads and generators and doesn't consider reactive power.

4.2.6 Methodology

This chapter analyzes the utility of graph theory analytics and graph clustering for application in power systems. IEEE 118 and 300 bus systems were selected as case studies. CBI indices were formulated using information metrics from basic system analysis and graph theory indices. The effectiveness of the formulated indices in quantifying meaningful topological information about the bus systems was tested using a node removal methodology. The most informative CBIs were determined based on the results of the node removal. Betweenness centrality was identified as a key metric. For further analysis, BC based graph clustering approaches were applied to the systems to

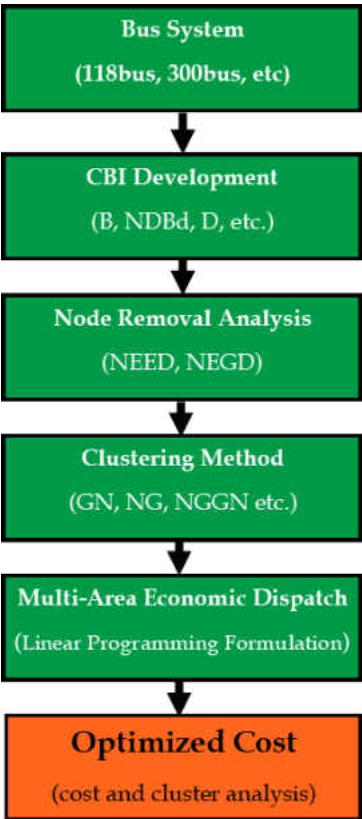


FIGURE 20. CBI AND MICROGRID DECOMPOSITION METHODOLOGY FLOW CHART

decompose the large systems into microgrids. Multiple combinations of graph clustering algorithms were tested based on their ability to improve ED for the entire system. A block diagram of the methodology is shown by figure 20.

### 4.3 Results

#### 4.3.1 CBI evaluation

The results of node removal and disruptivity analysis were tested on IEEE-118 and IEEE-300 bus test systems. The analysis NEGD and NEID node removal for the IEEE-118 bus system are shown in Figure 21, Figure 22, and Figure 23. Node removal for the IEEE-300 bus system are shown in Figure 24, Figure 25, and Figure 26.

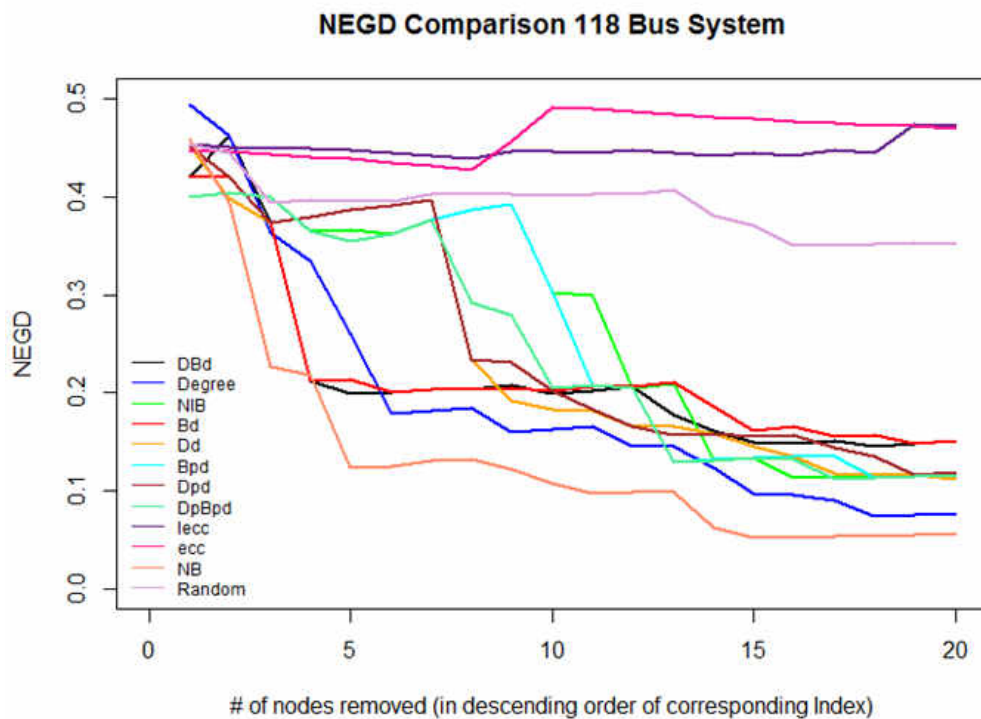
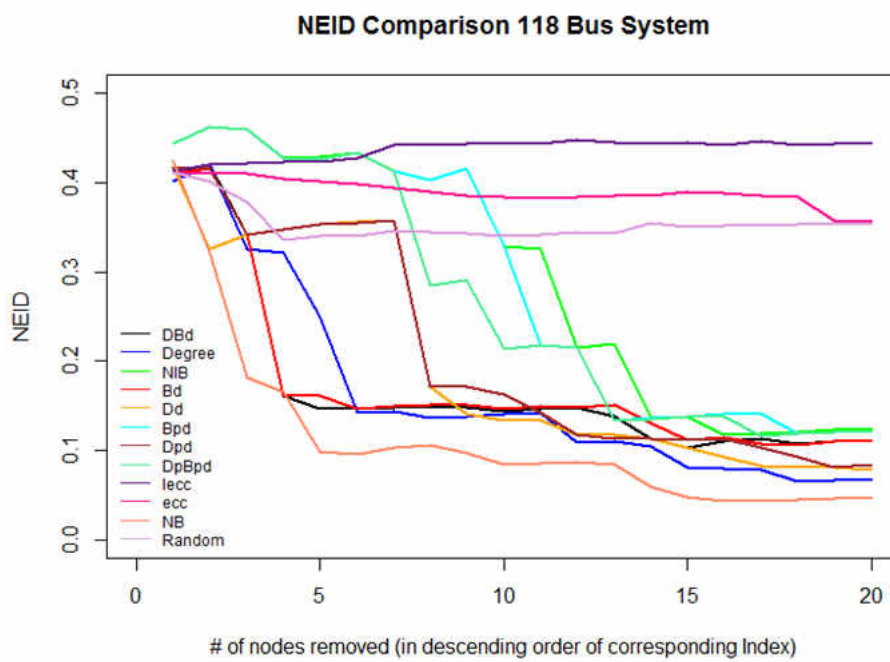


FIGURE 21. NEGD FOR ALL METRICS IN 118 BUS REMOVAL



**FIGURE 22. NEID COMPARISON FOR ALL METRICS IN 118 BUS REMOVAL**

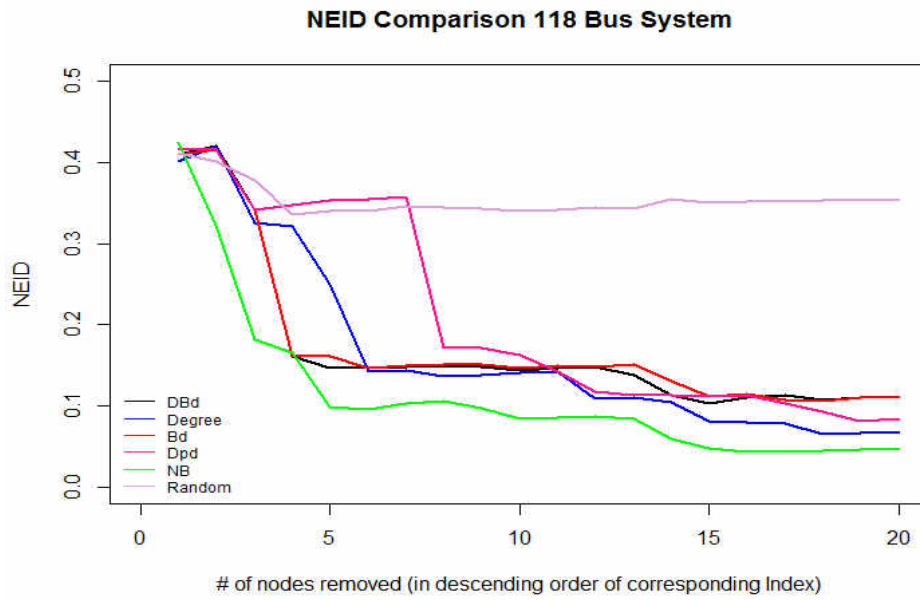


FIGURE 23. NEID COMPARISON FOR TOP 5 METRICS IN 118 BUS REMOVAL

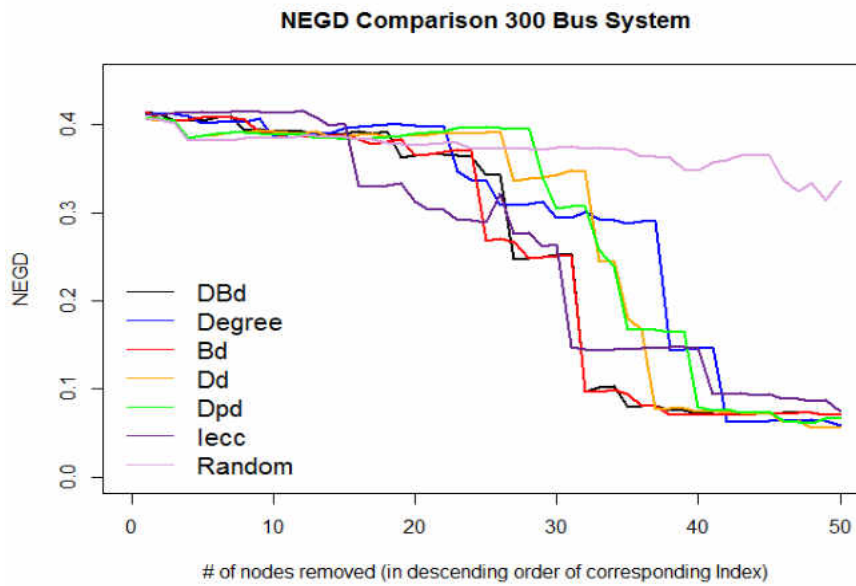


FIGURE 24. NEGD COMPARISON FOR MOST INFLUENTIAL INDICES OF 300 BUS REMOVAL

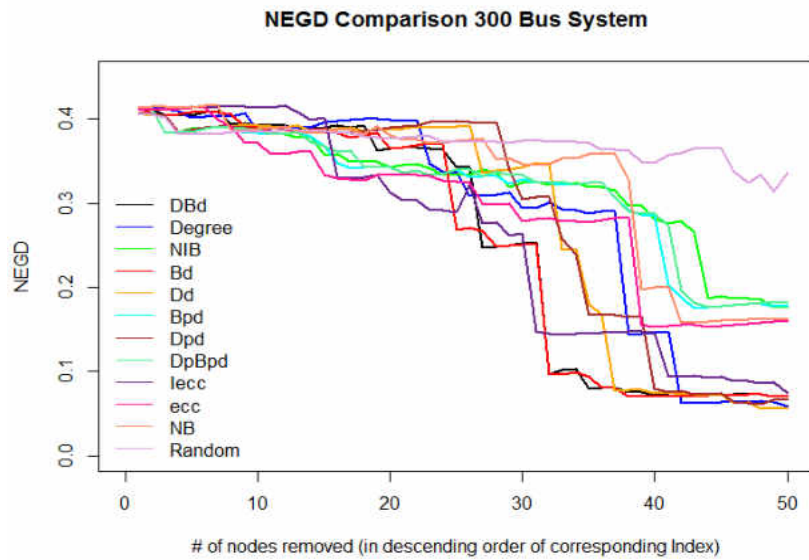


FIGURE 25. NEGD COMPARISON FOR ALL METRICS OF 300 BUS REMOVAL

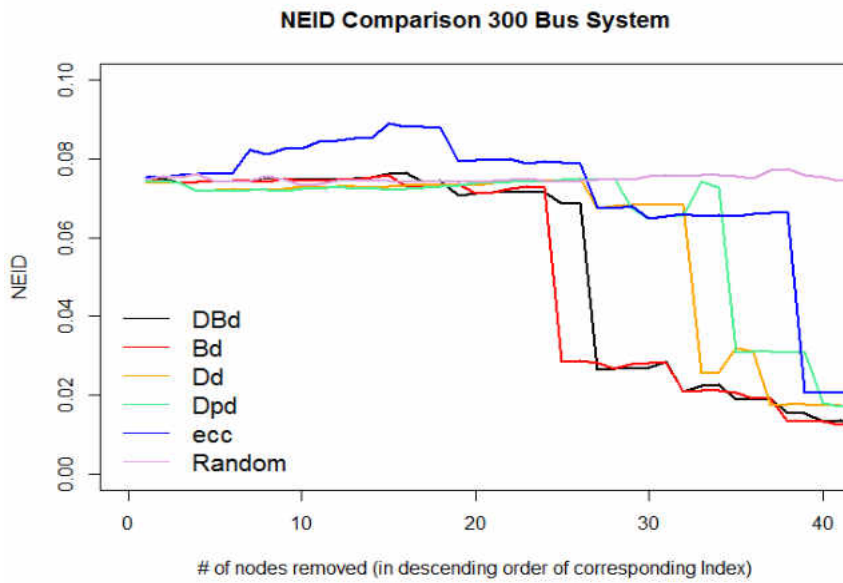


FIGURE 26. NEID COMPARISON FOR MOST INFLUENTIAL INDICES OF 300 BUS REMOVAL

#### 4.3.2 Discussion and interpretation of CBI and node removal

Figures 21-26 display the node removal process for IEEE-118 and IEEE-300 bus systems. A node removal is a bus failure, since the nodes in these systems refer to buses. When a node is removed, the edges connected to that node cannot be traveled on. This simulates a failure of a bus in a smart grid. When a bus fail, transmission of power through that bus is interrupted. Figures 21-26 compare the indices that were developed in this work. These indices are compared by removing the nodes in descending order of each index. For example, the node with the highest  $B_d$  index in the 118 system is node 81 (*e.g., bus 81*) and the node with the highest  $B$  index is node 75 (*e.g., bus 75*). The buses were removed in descending order. In the case of index  $B_d$ , the descending order starts with bus 81. In the case of index  $B$ , the descending order starts with bus 75.

Figures 21-26 display the impact of node removal is through the changes in NEGD and NEID. Specifically, when NEGD or NEID is sharply decreased through the removal of a bus, system disruption is observed. When NEGD and NEID are not decreased through the removal of nodes, this indicates that the system is not well-disrupted by the failure of the nodes. This was observed in figures 21-26 by the removal of random nodes. The removal of random nodes did not have a very large effect on NEGD or NEID compared to the indices formulated in this work.

Analysis of Figures 21-26 indicates that indices that are more heavily biased by betweenness are more disruptive to a system according to NEGD and NEID. This is observed by fact that the NEGD and NEID show a sharper decrease by a smaller number of buses that are failed through node removal. More specifically,  $B_d$  was the most effective index in determining importance in the IEEE-118 bus system. In the IEEE-300



bus system  $B_d$  was most effective when NEGD was analyzed as the disruption criteria, but betweenness ( $B$ ) was most effective in terms of NEID disruption. Because NEID is more relevant to power systems operation than NEGD, more merit is given to the results of NEID disruption.

In both test systems, a CBI denote by degree ( $D$ ) was also somewhat effective in quantifying disruption according to the expected distance metrics. However, it was not as effective as  $B_d$ . In many cases  $NB_d$  returned the same results as  $B_d$ . This is due to the heavy biasing of betweenness when multiplied by demand in this index. The influence of degree was often not sufficient to change the order of the bus indices; thus the removal order was the same. All indices in this work showed to be more effective in quantifying importance compared to a random node removal. This means that all the indices in this work quantify some level of importance of the buses to the connectivity of the systems. The removal of random nodes served as a baseline in analyzing the effectiveness of the other indices. The results show that random node removal did not significantly affect the NEGD or NEID even when large numbers of buses were removed.

#### 4.3.3 Results of graph clustering, modularity, and economic dispatch

Several different decomposition criteria were utilized in analyzing the different grid decompositions. The results section shows a sample of some effective techniques. Modularity scores for these criteria are recorded and shown in Table 4 and Table 5.

**TABLE 4. IEEE 118 BUS SYSTEM WITH MODULARITY SCORES FOR EACH CLUSTERING ALGORITHM AS WELL AS MGRs.**

<b>IEEE 118 BUS TEST SYSTEM</b>		
<b>CLUSTER SCHEME</b>	<b>MODULARITY</b>	<b>FOLLOWS MGR</b>
Topology - GN	0.6908	Yes
L-GN	0.6721	No
A-GN	0.74537	Yes
Topology-NG	0.5151	Yes
Length-NG	0.2995	Yes
Admittance-NG	0.1312	Yes
NGGN	0.6644	Yes

**TABLE 5. MODULARITY AND MGRs FOR IEEE 300 BUS SYSTEM FOR GN ALGORITHMS**

<b>IEEE 300 BUS TEST SYSTEM</b>		
<b>CLUSTER SCHEME</b>	<b>MODULARITY</b>	<b>FOLLOWS MGR</b>
Topology-GN	0.8344	No
L-GN	0.7824	No
G-GN	0.8344	No
NGGN	0.784	Yes

These tables show a modularity score, and whether the given decomposition follows rules for being considered a microgrid. Modularity index for microgrid decomposition is a useful metric in determining a grid structures ability to withstand microgrid or cascading failures. High modularity indicates dense microgrid intra-connection while simultaneously maintaining sparse interconnection with other microgrids. The physical bus system decomposition structures for the 118 and 300-bus systems that accompany these tables can be seen by the visualizations contained in Figure 27 and Figure 28. Table 6 and Table 7 list all buses in their respective zones obtained using the three clustering techniques for the IEEE 118 and 300 bus systems.

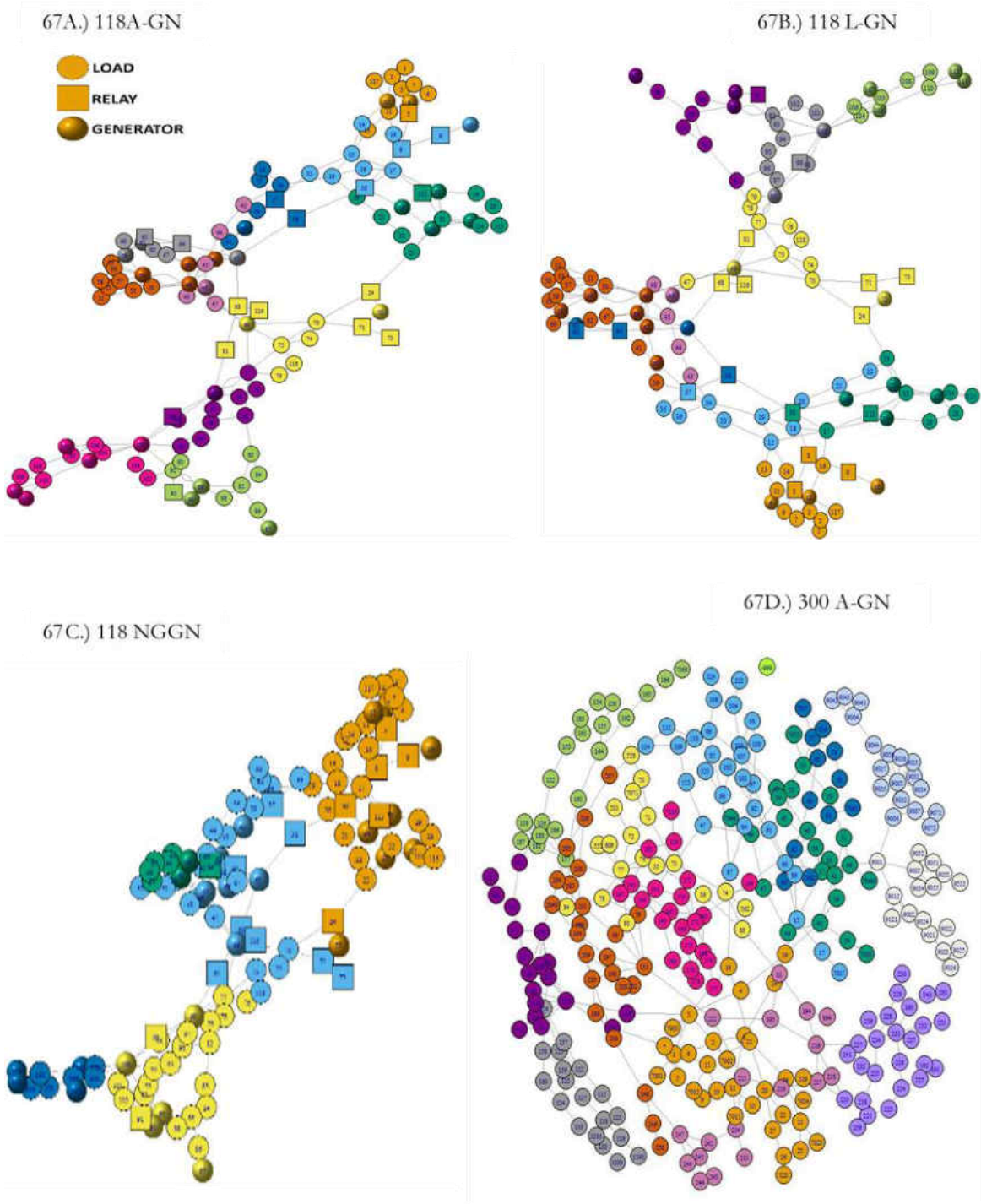
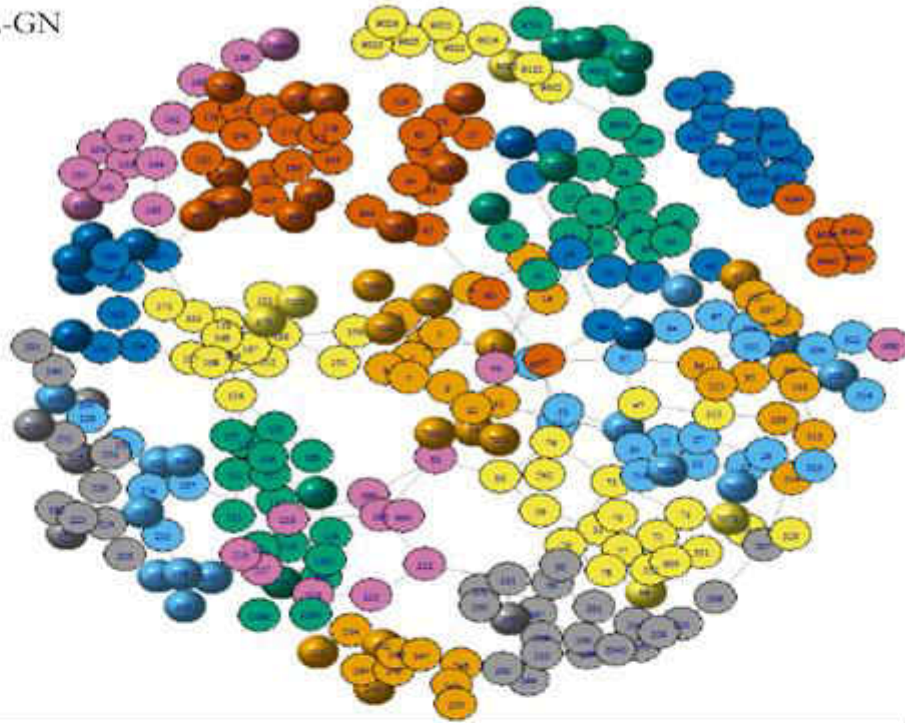


FIGURE 27. 118-BUS SYSTEM CLUSTERED WITH ADMITTANCE-WEIGHTED GN ALGORITHM. B.) 118 BUS SYSTEM CLUSTERED WITH LENGTH-WEIGHTED GN. C.) 118 BUS SYSTEM CLUSTERED WITH TWO STAGE NG+GN. D.) 300 BUS SYSTEM CLUSTERED WITH ADMITTANCE-WEIGHTED GN

A.) 300 L-GN



B.) 300 NGGN

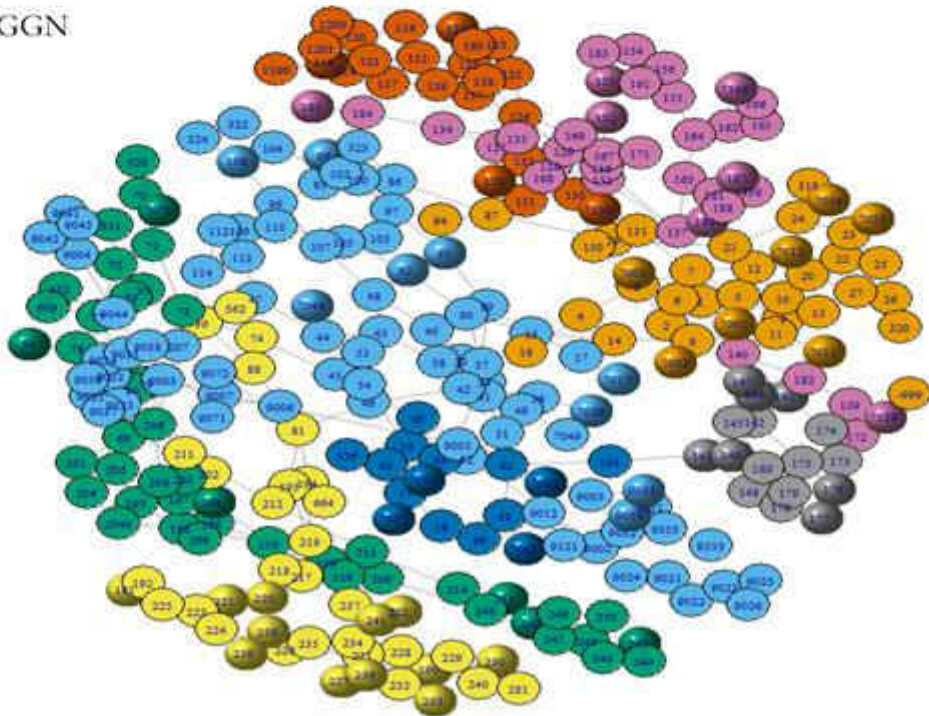


FIGURE 28. EXAMPLE DECOMPOSITIONS FOR 300 BUS SYSTEM. A.) 300 BUS SYSTEM CLUSTERED WITH LENGTH-WEIGHTED GN. B.) 300 BUS SYSTEM CLUSTERED TWO STAGE NGGN.

**TABLE 6. IEEE 118 BUS DECOMPOSITIONS BY ALGORITHM AND ZONE.**

	<b>L-GN</b>	<b>A-GN</b>	<b>LPGN</b>
<b>Zone 1</b>	1,2,3,4,5,6,7,8,9,10, 11,12,13,14,16,17,117	1,2,3,4,5,6,11,12,13, 117	1,2,3,4,5,6,7,8,9,10,11,12,13,14,1 6,117
<b>Zone 2</b>	8,9,10,14,16,17,18,19,30,33	17,23,25,26,27,28,29,30,31,32, 113,114,115	15,17,18,19,20,21,22,23,24,25,26, 27,28,29,30,31,32,33,113,114,115
<b>Zone 3</b>	15,18,19,20,21,22,33,34,35,36,37	20,21,22,23,25,26,27,28,29,31, 32,113,114,115	34,35,36,37,38,39,40,41,42,43,44, 45,46,47,48,49,50,51,52,53,54,55, 56,57,58
<b>Zone 4</b>	24,47,68,69,70,71,72,73,74,75,76,7 7,78,79 ,81,116,118	24,68,69,70,71,72,73,74,75,76, 81,116,118	59,60,61,62,63,64,65,66,67,68,69, 70,71,72,73,74,75,76,81,116,118
<b>Zone 5</b>	82,83,84,85,86,87,88,89,90,91	34,35,36,37,38,39,40,41	77,78,79,80,82,83,84,85,86,87,88, 89,90,91,92,93,94,95,96,97,98,99, 100,101,102,103
<b>Zone 6</b>	38,61,63,64,65	43,44,45,46,47,48	104,105,106,107,108,109,110,111 ,112
<b>Zone 7</b>	43,44,45,46,48	42,49,50,51,52,53,54,55,56,57, 58,66	NA
<b>Zone 8</b>	39,40,41,42,49,50,51,52,53,54,55,5 6,57,58,59,60, 62,66,67	59,60,61,62,63,64,65,67	
<b>Zone 9</b>	80,92,93,94,95,96,97,98,99,100,101 ,102	77,78,79,80,82,94,95,96,97,98, 99	
<b>Zone 10</b>	103,104,105,106,107,108,109,110,1 11,112	100,101,102,103,104,105,106,1 07,108,109,110,111,112	
<b>Zone 11</b>	NA	83,84,85,86,87,88,89,90,91,92, 93	

**TABLE 7. BUS ASSIGNMENTS FOR EACH GN ALGORITHM FOR 300 BUS SYSTEM**

	<b>L-GN</b>	<b>A-GN</b>	<b>NGGN</b>
<b>Zone 1</b>	1,2,3,4,5,6,7,8,9,10,11, 12,13,14,16,19,20,21,2 2,23,24,25,26,27,319,3 20,7001,7002,7003,701 1,7012,7023	1,2,3,4,5,6,7,8,9,10,11,12,1 3,14,16,19,20,21,22,23,24, 25,26,27,319,320,7001,700 2,7003,7011,7012,7023	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19,20,2 1,22,23,24,25,26,27,128,129,130,131,150,167,16 8,319,320,7001,7002,7003,7011,7012,7017,7023, 7024,7130

<b>Zone 2</b>	15,17,47,85,86,87,89,90,91,92,94,97,98,99,100,102,103,104,105,107,108,109,110,112,113,114,322,323,324,7017	15,17,47,85,86,87,89,90,91,92,94,97,98,99,100,102,103,104,105,107,108,109,110,112,113,114,322,323,324,7017	33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,51,52,53,54,55,69,70,71,72,73,74,76,77,78,79,80,81,84,85,86,87,88,89,90,91,92,94,97,98,99,100,102,103,104,105,107,108,109,110,112,113,114,189,193,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,322,323,324,528,531,552,562,609,2040,7039,7044,7049,7055,7071
<b>Zone 3</b>	33,34,37,38,39,40,41,42,43,44,45,46,48,49,51,52,53,54,55,7039,7044,7049,7055	33,34,37,38,39,40,41,42,43,44,45,46,48,49,51,52,53,54,55,7039,7044,7049,7055	57,58,59,60,61,62,63,64,526,7057,7061,7062
<b>Zone 4</b>	35,36,70,71,72,73,74,76,77,78,80,84,88,528,531,552,562,609,7071	35,36,70,71,72,73,74,76,77,78,80,84,88,528,531,552,562,609,7071	115,116,117,118,119,120,121,122,123,124,125,126,127,132,133,134,135,136,137,138,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,169,170,171,181,183,184,185,186,187,188,1190,1200,1201,7166
<b>Zone 5</b>	57,58,59,60,61,62,63,64,526,7057,7061,7062	57,58,59,60,61,62,63,64,526,7057,7061,7062	139,140,141,142,143,144,145,146,147,148,149,172,173,174,175,176,177,178,179,180,182,7139
<b>Zone 6</b>	69,79,189,193,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,248,249,250,2040	69,79,189,193,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,248,249,250,2040	190,191,192,194,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,281,664
<b>Zone 7</b>	81,194,195,212,213,214,215,216,217,218,219,242,243,244,245,246,247,664	81,194,195,212,213,214,215,216,217,218,219,242,243,244,245,246,247,664	213,214,215,242,243,244,245,246,247,248,249,250
<b>Zone 8</b>	115,116,117,118,119,120,121,122,123,124,125,126,157,158,159,160,1190,1200,1201	115,116,117,118,119,120,121,122,123,124,125,126,157,158,159,160,1190,1200,1201	9001,9002,9003,9004,9005,9006,9007,9012,9021,9022,9023,9024,9025,9026,9031,9032,9033,9034,9035,9036,9037,9038,9041,9042,9043,9044,9051,9052,9053,9054,9055,9071,9072,9121,9533
<b>Zone 9</b>	127,128,129,130,131,132,133,134,135,150,151,167,168,169,170,171,184,185,7130	127,128,129,130,131,132,133,134,135,150,151,167,168,169,170,171,184,185,7130	NA
<b>Zone 10</b>	136,137,138,152,153,154,155,156,161,162,163,164,165,166,181,183,186,187,188,7166	136,137,138,152,153,154,155,156,161,162,163,164,165,166,181,183,186,187,188,7166	
<b>Zone 11</b>	139,140,141,142,143,144,145,146,147,148,149,172,173,174,175,176,177,178,179,180,182,7139	139,140,141,142,143,144,145,146,147,148,149,172,173,174,175,176,177,178,179,180,182,7139	

<b>Zone 12</b>	190,191,192,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,281	190,191,192,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,281	
<b>Zone 13</b>	9001,9002,9005,9012,9021,9022,9023,9024,9025,9026,9051,9052,9053,9054,9055,9121,9533	9001,9002,9005,9012,9021,9022,9023,9024,9025,9026,9051,9052,9053,9054,9055,9121,9533	
<b>Zone 14</b>	9003,9004,9006,9007,9031,9032,9033,9034,9035,9036,9037,9038,9041,9042,9043,9044,9071,9072	9003,9004,9006,9007,9031,9032,9033,9034,9035,9036,9037,9038,9041,9042,9043,9044,9071,9072	

**TABLE 8. ECONOMIC DISPATCH COST RESULTS FOR 118 BUS SYSTEM**

	<b>L-GN</b>	<b>A-GN</b>	<b>NGGN</b>
<b>Number of clusters</b>	10	11	5
<b>Generation Cost (\$)</b>	9074.86	9137.03	9148.06
<b>Tie-line flow cost (\$)</b>	262.871	87.725	36.45
<b>Total Cost (\$)</b>	9337.73	9224.76	9184.51

**TABLE 9. ECONOMIC DISPATCH COST RESULTS FOR 300 BUS SYSTEM**

	<b>L-GN</b>	<b>A-GN</b>	<b>NGGN</b>
<b>Generation Cost (\$)</b>	141704	141704	123824
<b>Number of clusters</b>	14	14	8
<b>Tie-line flow cost (\$)</b>	233.089	233.089	163.361
<b>Total Cost (\$)</b>	141937	141937	123988



**TABLE 10. GENERATOR RATING/LOAD RATIO OF IEEE 118 CLUSTERS**

Generation/Load ratio	L-GN	A-GN	NGGN
Maximum value (%)	198.61	277.08	124.48
Minimum value (%)	49.62	47.22	50.63

**TABLE 11. GENERATOR RATING/LOAD RATIO OF IEEE 300 BUS CLUSTERS**

Generation/Load ratio	L-GN	A-GN	NGGN
Maximum value (%)	616.61	616.61	411.86
Minimum value (%)	61.90	61.90	86.96

#### 4.4. Discussion and Interpretation

Table 8 and Table 9 list the generation cost, the tie-line flow cost, and the total cost for the IEEE 118 and IEEE 300-bus systems for the Length-GN (L-GN), Admittance-GN (A-GN), and NGGN clustering technique, respectively. For the 118-bus system, there is a 66.6% reduction in tie-line flow cost for the A-GN clustered system when compared to the L-GN system and there is a significant reduction of 86.13% for the NGGN method compared to the L-GN method. For the total cost, the cost reductions are 1.21% and 1.64%, respectively, for the A-GN and NGGN method. For the 300-bus system case, the value for the tie-line flow-cost reduction is 0% for the A-GN method because the cluster was identical to the L-GN method and it's 29.91% for the NGGN method. Similarly, the reduction for the total costs are 0% for the A-GN method and 12.64% for the NGGN method. From these results, there is a significant reduction in the tie-line flow cost for the NGGN clustering technique compared to the L-GN and A-GN techniques. The usage of NGGN method also results in the reduction of total cost for the system.

Another parameter used to compare grid clusters is the generation to load (G/L) ratio. A G/L value that is more than 100% indicates a self-sufficient grid cluster with excess generation that can be given to other micro grids. A G/L value less than 100% indicates that the generation within the cluster is not sufficient to satisfy its load, thus requiring resources from neighboring micro grids to meet the demand. Table 10 and Table 11 list the maximum and minimum value of this G/L for the IEEE 118-bus and 300-bus systems, excluding the zones with no active power generation. It is evident that the NGGN clusters are more suited due to its self-sufficiency, when compared to the other two cases, because the values are closer to the ideal value of 100.

Several decomposition criteria were utilized to analyze multiple-grid structures in conjunction with economic dispatch. The modularity scores for these criteria were recorded and shown in Table 4 and Table 5 respectively. These tables show that the impact of modularity score on a given decomposition. The modularity index for micro grid decomposition is a useful metric to determine a grid structure's ability to withstand microgrid or cascading failures. A higher modularity score indicates a dense micro grid intra-connection while simultaneously maintaining a sparse interconnection with other microgrids. The physical bus system's decomposition structures for the 118- and 300-bus systems that accompany these tables can be seen with the visualizations abstracted to graphs using R software.

Preliminary results indicate that higher modularity scores for any bus system occur when using the GN algorithm with the edge weights weighted with the admittance of the transmission lines. This was evident in the 118-bus and 300-bus system. The nearest-

generator (NG) algorithm works poorly when applied by itself in the 118- and 300-bus systems.

Betweenness centrality is a good metric to determine the microgrid structures within a given grid system, if the generators are well-distributed. This distribution is likely to be the case for smart-grid networks. Further work needs to be done to observe how these algorithms scale well for larger systems. The modularity index for the microgrid decomposition is also a useful metric to determine a grid structure's ability to withstand cascading failures. High modularity indicates dense microgrid intra-connection while simultaneously maintaining sparse interconnection with other microgrids. However, a decomposition's modularity score does not appear to have a significant relationship with the economic dispatch optimization. The combination of the nearest generator algorithm in a two-stage decomposition with betweenness clustering forms a clustering scheme that logically follows the demands of microgrids while making use of the connective topology of the system quantified by betweenness.

## 4.5 Conclusions

This chapter presents a preliminary work in quantifying the importance of individual buses to the operation of a power grid. This work examines the metrics of betweenness, degree, demand, generation, and interactions of these quantities that lead to effective quantification of bus importance. Five indices were created and tested through a process of node removal. This process is a well-known technique to the fields of social-network theory and computer science, and serve as a preliminary method to examine the effectiveness of the indices that were developed. As the node removal process was

tested, the effect of removing the nodes (*e.g.*, *failing buses*) was quantified by examining the effect on NEGD and NEED.

The results indicate that indices based heavily on betweenness centrality show more disruption in a smaller number of failed nodes. Furthermore, betweenness centrality is an effective metric for quantifying the importance of a bus to the transmission of power through the system. This was quantified by large decreases in NEED, which is a relevant computational metric to understand power system disruption, in a small number of nodes removed.

When examining the economic dispatch of different micro grid decompositions, results indicate that decompositions formed using the two-stage clustering method, NNGN show a reduced cost for economic dispatch. This cost reduction is mostly due to savings that occur in the tie-line flow. The savings due to generation cost are smaller and are not significant. This reduction is due to the evenly distributed tie-line flows that due to proposed two-stage clustering approach. Although the modularity for two-stage clustering was slightly lower than it was for the one-stage schemes, the economic dispatch is very cost-effective. The two-stage clustering method using the admittance and/or impedance weighted betweenness coupled with the nearest-generator method is a novel contribution. Adding, this method do show an overall reduction in the dispatch cost compared to the single-stage clustering methods.

## Chapter 5. Investigation of Time-Series Clustering for Demand Profile Classification and Improved Load Forecasting

### 5.1 Summary

In this chapter, a forecasting framework for residential energy demand of homes with and without electric vehicles based on time-series clustering is demonstrated. Time-series energy consumption data from 200 households as well as electric vehicle charging for 200 electric vehicles associated with the households was analyzed. This work proposes and compares a novel implementation of the k-shape and partitional DTW time-series clustering algorithms to improve forecast accuracy and discover residential load profiles. As adoption of electric vehicles increases exponentially globally, there is a need for continuous relevant research on the impacts of charging infrastructure integration on the grid. The increased number of electric vehicles imposes enormous power requirements, and this leads to power imbalance which effects the stability of grid. This work uses novel analysis to provide insight into residential load forecasting in the presence of electric vehicles. With the increase in smart meter and smart grid technologies, forecasting at the residential level is becoming more prevalent. This work implements a framework to provide meaningful analysis and enhanced forecast accuracy for household energy demand data based on smart meter data.

### 5.2 Methods

#### 5.2.1. Background

Accurate models for electric power load forecasting are essential to the operation and planning of a utility company. Forecasting in power systems is an active area of research with many contributors. Because forecasting is closely related to economic success, many resources are allocated for accurate forecasting. As a result, many

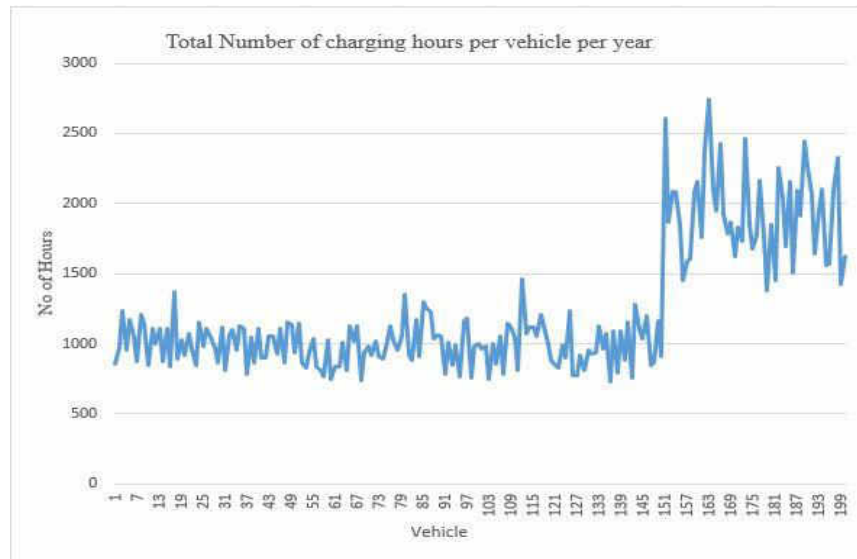
algorithms and methodologies have been developed, tested, and evaluated with varying levels of accuracy. There are a multitude of forecasting works and surveys that outline the applications related to load forecasting [26], [27], [51], [74], [89]–[93]. The success of a forecasting algorithm depends on the proper application of appropriate algorithms and any regressive variables to consider [94].

Though there are a multitude of forecasting methodologies, there is limited literature regarding using time-series clustering as a processing step for a forecasting paradigm. With the advent of smart meters, where individual household demand patterns can be known, the capability for accurate forecasting based on this new multitude of data is intriguing. However, with more data comes more challenges. If more smart meters are deployed there is more data to handle. The ability to use clustering algorithms to classify time-series data is a natural application for smart meter data. Clustering algorithms can classify which households are most similar to one another based upon their electricity demand time-series. There have been a few works that have attempted to exploit this capability, but there are no existing works that deploy time-series clustering techniques in a forecasting scheme in a large-scale case study like the one proposed by this thesis.

k-Shape has been utilized for analysis for other energy demand data [40], [95], but has not been utilized for power systems demand data on a scale proposed by this work. This work proposes an agglomerative forecasting scheme similar to the scheme proposed in [96], [97] but applies it on a larger scale to 200 smart meter datasets for an aggregate forecast. Works similar to this have been done on small scales, however, the scale of this work and the specific forecast combination is a novel contribution to literature.

### 5.2.2 Understanding the smart meter data

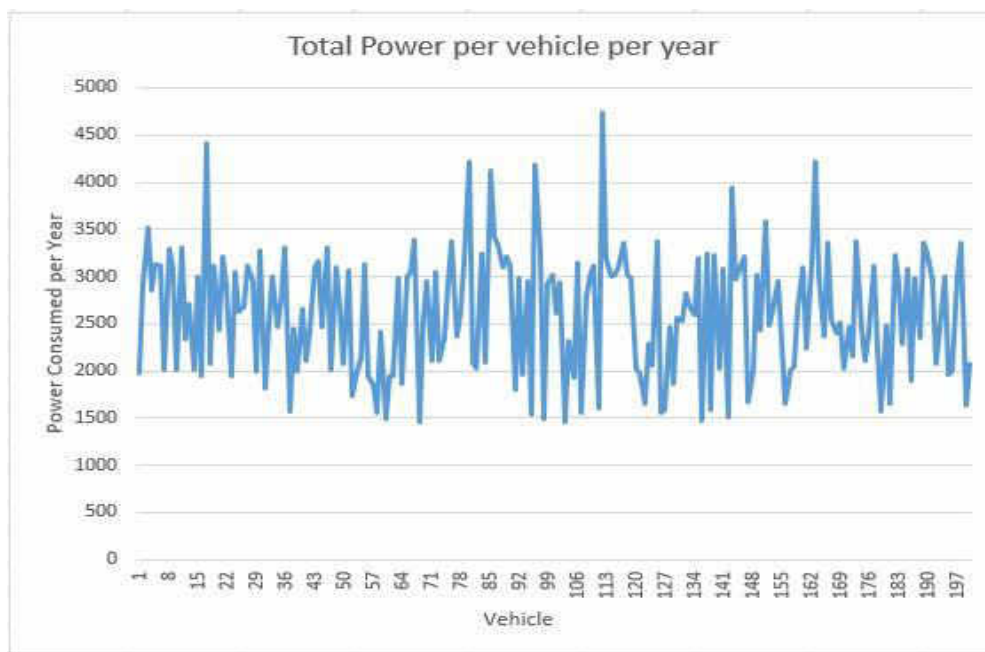
To properly understand the data, basic analysis is needed and adds value. An intriguing feature of this data is the existence of EV charging. There is vast amount of existing information regarding household energy demand profiles. However, EVs are relatively new to the industry and information regarding EV charging is still being discovered. This chapter analyzes hourly interval EV charging data in addition to the residential demand of the homes that they are associated with. Basic analysis of the charging was gathered to understand the difference between households that contain EVs and households that do not. Specifically, the number of charging hours per year and power consumed by each vehicle were analyzed and are shown in figures 29 and 30.



**FIGURE 29. YEARLY CHARGING HOURS OF THE 200 EVs IN THE SMART METER DATA**

The graph in figure 29 demonstrates the number of hours charged per vehicle per year. It is clear by looking at this data there appear to be two distinct amount of vehicle charging demands in terms of number of hours of charging per year. Based on this plot, there are about 150 vehicles that require about 1000 hours of charging per year and there

are 50 vehicles that require charging in the neighborhood of 2000 hours per year. More analysis needs to be done to uncover why there appear to be two distinct categories of charging hours per year. It is a peculiar trend to exist within the data.



**FIGURE 30. POWER CONSUMPTION FOR CHARGING OF EACH EV IN ONE YEAR.**

In addition to analysis of the charging hours per year, the amount of power consumed by EV charging is crucial from a utility perspective. Figure 30 shows the amount of power consumed by each vehicle in the dataset throughout the one year of observation. Even though figure 39 indicates that two distinct patterns of charging hours exists, the amount of power consumed by charging is more random and does not appear to exhibit any clear distinguishable trends.

### 5.2.3 Methodology

Time series clustering methods are applied (k-shape, k-means DTW, and k-means Euclidean). Because each of these schemes require input to decide the “k” number of clusters, a cluster evaluation scheme is implemented to decide how many clusters is



appropriate for each scheme. After the appropriate scheme has been identified, forecasting of each cluster of households is performed and compared to a traditional forecasting method that does not involve clustering. The traditional aggregate forecast and the forecasting of the clustered residences both invoke the two-stage loess-ARIMA method and the accuracy of the schemes is analyzed according to SSD, MSD, and MAPE. Residential load and electric vehicle charging data used in this work can be found at [98].

### 5.3 Results

#### 5.3.1 Time-series clustering results

**TABLE 12. CVIs FOR K-SHAPE CLUSTERING OF HOUSEHOLDS W/ EVs**

Scheme\CVI	Sil	SF	CH	DB	DBstar	D	COP
k2h	0.027315	0.310251	23.09686	7.448245	7.448245	0.393895	0.693639
k3h	0.051601	0.187712	17.15088	3.834803	3.919687	0.396952	0.649698
k4h	0.030665	0.138325	26.77724	6.500725	6.850897	0.402034	0.643143
k5h	0.049677	0.118644	8.518017	2.696609	2.698605	0.489629	0.64952
k6h	0.055925	0.056901	19.06344	3.438495	3.504191	0.393895	0.628746

**TABLE 13. CVIs FOR DTW CLUSTERING FOR HOUSEHOLDS W/ EVs**

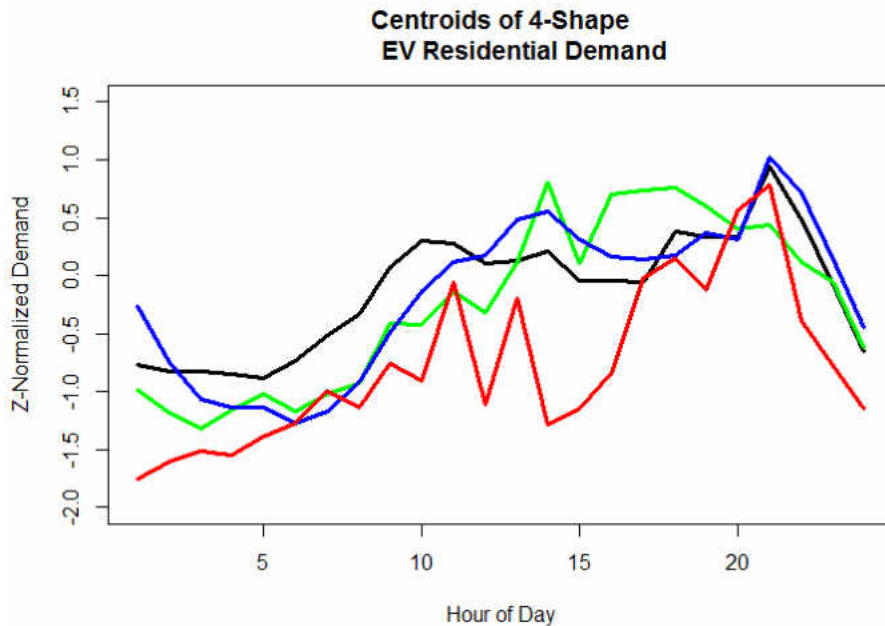
scheme\CVI	Sil	SF	CH	DB	DBstar	D	COP
d7h2	0.12855	0	132.2373	2.461714	2.461714	0.331591	0.572962
d7h3	0.154758	0	68.61463	2.078887	2.115389	0.402169	0.554176
d7h4	0.033072	0	48.71395	2.53176	3.03934	0.274114	0.509596
d7h5	0.089204	0	36.90802	1.916859	2.076822	0.350904	0.534602
d7h6	0.068136	0	31.07898	2.140913	2.472272	0.331136	0.507131

**TABLE 14. SCALED CVIs TO COMPARE ACROSS INDICES FOR DTW CLUSTERING OF HOUSEHOLDS W/ EVs**

Scheme\Index	Sil	SF	CH	DB	DBstar	D	COP
k2NE	0.101874	0.355259	38.8252	4.274693	4.274693	0.105779	0.554561
k3NE	0.109388	0.249146	25.70618	3.117905	3.380108	0.064332	0.529495
k4NE	0.135497	0.177635	16.68158	2.549536	2.651274	0.105779	0.530371
k5NE	0.111542	0.124692	15.08233	2.842039	3.152875	0.126581	0.504502
k6NE	0.126706	0.128691	11.12589	2.221103	2.373568	0.069211	0.506074

**TABLE 15. CVIs FOR K-SHAPE CLUSTERING OF HOUSEHOLDS W/OUT EVs**

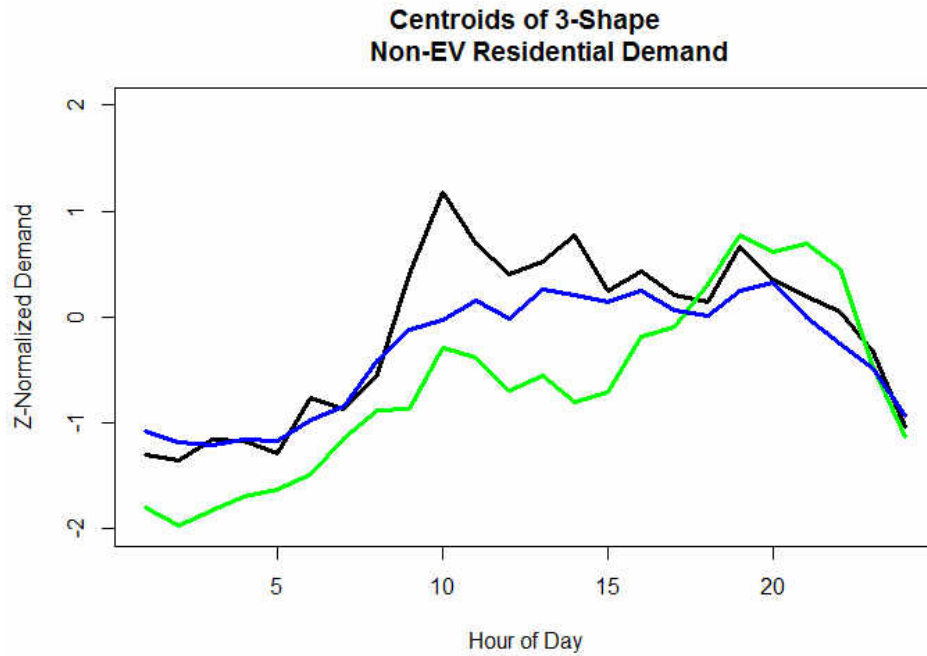
scheme\CVI	Sil	SF	CH	DB	DBstar	D	COP	SSI
d7h2	0.701615	0	1.675781	0.901339	0.074001	-0.13903	1.311798	4.525502
d7h3	1.245521	0	0.124454	-0.56271	-0.82188	1.396194	0.650548	2.032125
d7h4	-1.27994	0	-0.36079	1.169212	1.568223	-1.38929	-0.91858	-1.21117
d7h5	-0.11498	0	-0.64866	-1.18235	-0.92165	0.281068	-0.03843	-2.62499
d7h6	-0.55222	0	-0.79079	-0.3255	0.101313	-0.14894	-1.00534	-2.72147



**FIGURE 31. CENTROID DAILY DEMAND PROFILES FROM 4-SHAPE CLUSTERING FOR HOUSEHOLDS W/ EVs**

The compactness and separation of the clustering schemes were quantified by seven CVIs. Tables 12 and 15 show the CVI analysis of the k-shape clustering schemes. The CVIs were used to evaluate the effect number of clusters that are appropriate for this data. For example, table 12 displays the CVIs for the k-shape clustering of households

with EVs. The most appropriate k-shape clustering method was selected by analyzing which number of clusters, k, is consistently resulting in the highest CVIs according to the seven CVIs that were analyzed.



**FIGURE 32. CENTROIDS OF DAILY HOUSEHOLD DEMAND FOUND BY 3-SHAPE CLUSTERING FOR HOUSEHOLDS W/OUT EVS**

### 5.3.2 Forecasting Results

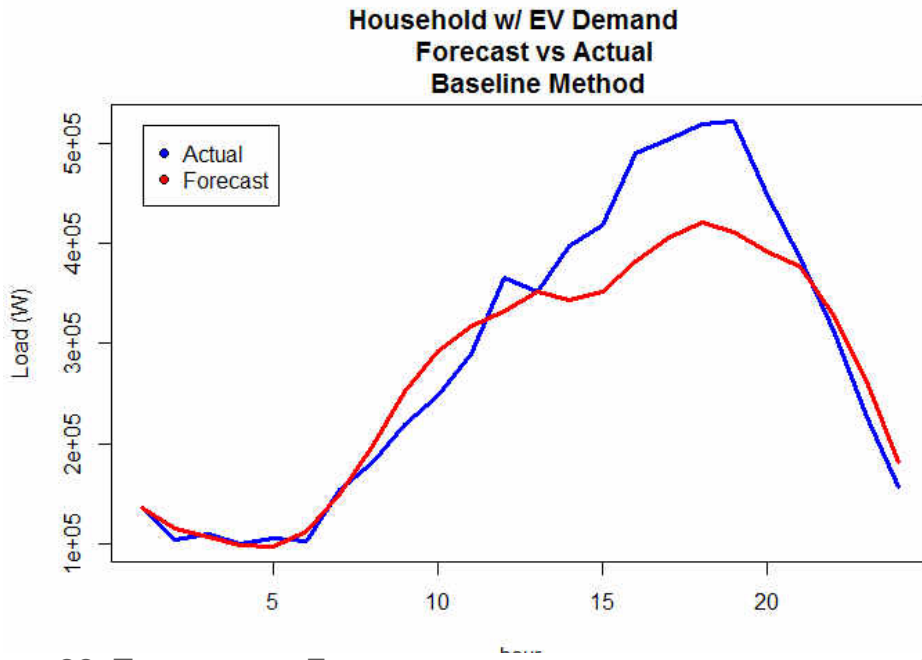


FIGURE 33. TRADITIONAL FORECASTING SCHEME FOR HOUSEHOLDS W/ EVs

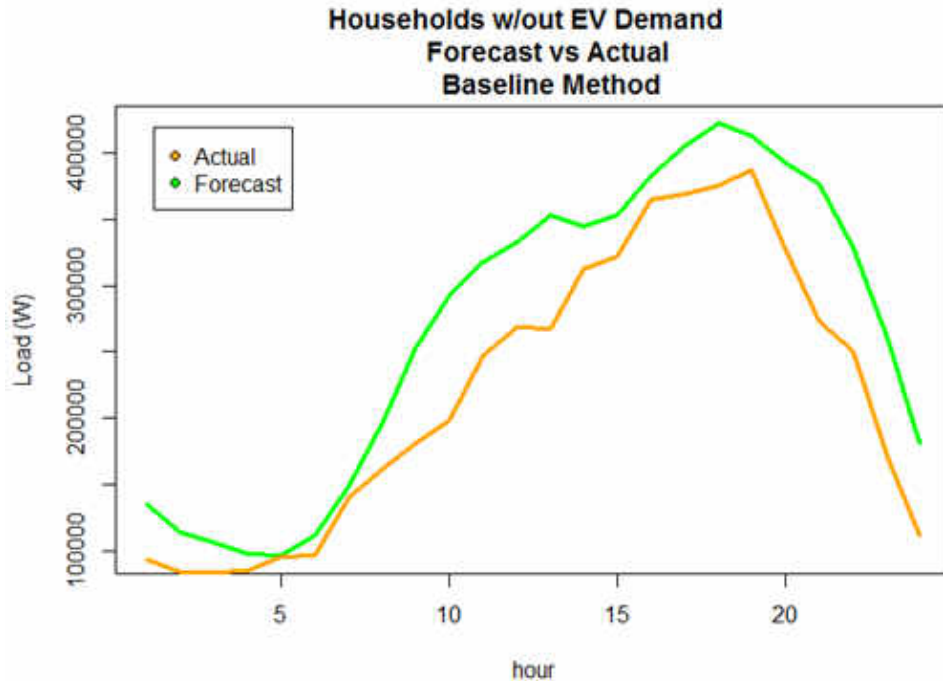


FIGURE 34. TRADITIONAL FORECASTING METHOD APPLIED TO HOUSEHOLDS W/OUT EVs

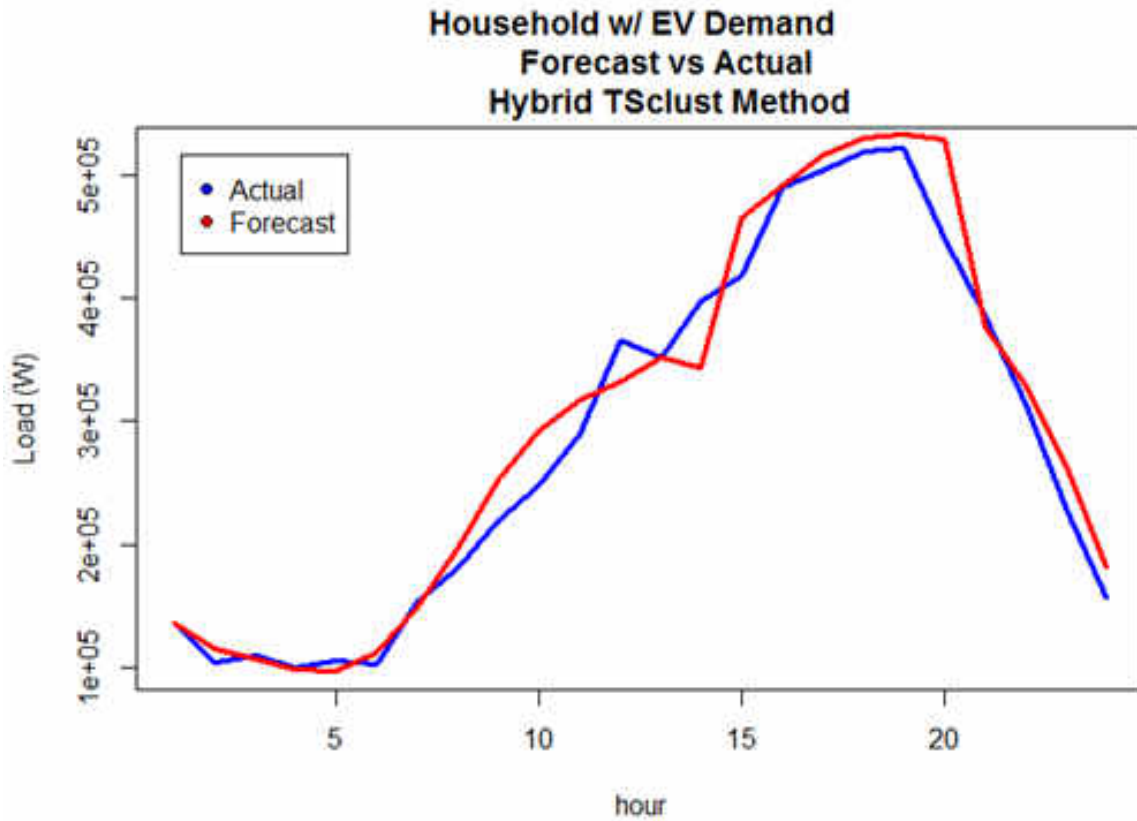


FIGURE 35. PROPOSED FORECASTING METHOD THAT USES TIME-SERIES CLUSTERING APPLIED TO HOUSEHOLDS W/ EVs

Figures 33 and 34 show the baseline forecasting methods. These forecasting methods show the traditional STL-ARIMA forecasting results for a day-ahead forecasting using 30 days previous as training data. These methods do not consider any time-series clustering in the forecasting scheme. These forecasts were used to compare with the results from forecasting that used time-series clustering. The MAPE for forecasting of households that contained EVs was MAPE=10.6%. For non-EV households the MAPE was over 24% using this baseline method.

## 5.4 Discussion and Interpretation

### 5.4.1 Clustering Evaluation

Based on the CVI in table 12, k-shape clustering with 4 clusters was consistently giving higher values of CVI across the 7 indices so it was selected as the method to use for forecasting. Similarly, in table 13, k-shape with 3 clusters consistently performed well according to the CVIs. Therefore, this scheme was used for forecasting for the non-EV dataset. It is also noted that in both cases, k=2 clusters performed well according to the CVIs.

Figures 12 and 13 display the centroids of the selected k-shape clustering schemes. For the residential data that contained EVs, the k-shape scheme with 4 clusters was selected. For the non-EV data, k-shape with 3 clusters was selected. The centroids represent decomposed time series from within the data. These centroids represent common daily load profiles from within the data and they are the centroids of the clusters found by the k-shape algorithm. The centroids represent a characteristic time series upon which each time-series are quantified in relation to these clusters using the distance metrics of the k-shape algorithm. Examining figure 12 for example, the results are

interpreted as the 3 types of daily demand profiles that exist from within the dataset. The k-shape clustering scheme can generally be stated as separating the different types of demand profiles that exist from within the dataset.

#### 5.4.2 Forecasting evaluation

The method of forecasting used for the plot in figure 35 was a newly proposed hybrid method. This forecasting scheme uses the time-series clustering framework to forecast for peak hours, but uses traditional forecasting schemes to forecast for off peak hours. The accuracy of this method is shown to be an increase over both the baseline method and the time-series clustering methods alone. When combined, the strengths of both forecasts can be leveraged. The time-series cluster was shown to be accurate for forecasting during peak times, but inaccurate during off-peak hours. Conversely, the baseline method was more accurate during off-peak hours and inaccurate during peak consumption periods. The MAPE of the proposed hybrid scheme is  $MAPE=7.6\%$ . This is a 3% reduction as compared to the next best forecasting scheme for this data.

#### 5.5 Conclusions

A hybrid forecasting framework utilizing time-series clustering is a promising approach to achieving accurate forecast values for high volume datasets that will be encountered by the increased use of smart meters. Traditional forecasting methods are accurate during non-peak periods. These methods are accurate because off-peak periods are more regular and thus easier to forecast using traditional methods. The traditional method used in this work was less accurate during peak periods. The use of time-series clustering to classify the data into different categories helped in increasing forecast accuracy during peak periods. The increase in accuracy of the clustering methods occurs

because most of the variation in time series occurs during peak periods. Thus, when a clustering algorithm is applied to the time-series data in this work, most of the differences between individual time-series occur during peak hours. These differences are captured by the clustering algorithms, and the peak periods that are most like one another are clustered together. This clustering of like time-series allows for more precise forecasting of different types of demand profiles that exist from within a utility's jurisdiction.



## Chapter 6. Results and Future Directions

### 6.1 Summary

This thesis demonstrates the effectiveness of clustering algorithms for 3 scenarios of smart grid data analysis, while comparing and analyzing specific methods of clustering that are most appropriate for each application. The utility of hclust for anomaly detection in phasor measurement unit (PMU) datasets was demonstrated. Hclust was effective in identifying anomalies according to Dunn Index (DI) criteria. A method previously demonstrated in literature, Density Based Spatial Clustering of Applications with Noise (DBSCAN) performed less effectively according to DI and was computationally inefficient in comparison to hclust.

The efficacy of betweenness-centrality (BC) for topological analysis was shown in two phases. BC was compared against other indices and was the most efficient index according to node removal. To further analyze its utility, betweenness centrality-based graph clustering (BCGC) was used in a novel clustering scheme for the determination of microgrids from large scale bus systems. BCGC was demonstrated and compared against other graph clustering techniques. The BC based clustering showed an overall decrease in economic dispatch cost when compared to other methods of graph clustering. Additionally, the utility of BC for identification of critical buses was showcased.

Finally, this work demonstrates the utility of partitional dynamic time warping (DTW) and k-shape clustering methods for classifying power demand profiles of households with and without electric vehicles (EVs). The utility of DTW time-series clustering was compared against other methods of time-series clustering and tested based upon its ability to improve demand forecasting using traditional forecasting

techniques as a baseline. Additionally, a process for selecting an optimal time-series clustering scheme based upon a scaled sum of cluster validity indices (CVIs) was developed. Forecasting schemes based on DTW and k-shape demand profiles showed an overall increase in forecast accuracy.

In summary, the use of clustering methods for three distinct types of smart grid datasets is demonstrated. The use of clustering algorithms as a means of processing data can lead to overall methods that improve forecasting, economic dispatch, event detection, and overall system operation. These three specific areas of application are critically important for optimal power systems operation as well as the economic success of utilities or software that may employ these techniques. The use of data clustering algorithms allows power systems operators to gain actionable insights from otherwise ambiguous power systems data. Ultimately, the techniques demonstrated in this thesis give analytical insights and foster data-driven management and automation for smart grid power systems of the future.

## 6.2 Future directions

This thesis has demonstrated 3 situations where clustering algorithms can improve operational efficiency or situational awareness in power systems. Though the results of this work are conclusive, there are areas where further work would provide even greater meaning.

For the application of clustering algorithms to streaming PMU data, a future work could analyze the use of hclust in combination with machine learning for autonomous detection of fault and give further insight as to the type of fault. The current method of hclust shows a good method to detect faults, but further research involving the use of

machine learning techniques could diagnose specific fault types based upon data distributions and pattern the data presents in the fault.

In the application of time-series clustering for load forecasting, there are many variations of the proposed method of cluster-based forecasting that could be analyzed. One proposed method would be to use cluster identity as a regressive variable in the forecasting scheme. Additionally, the use of non-traditional forecasting schemes such as deep learning or neural networks may be an appropriate selection for smart meter data. Since the volume of data from smart meters is large, a forecasting scheme that uses time-series clustering in combination with deep learning approaches could further reduce forecast error.

## BIBLIOGRAPHY

- [1] Texas-Tech-University, "Introduction to Smart Grid," 2012. [Online]. Available: [http://www.ee.ucr.edu/~hamed/Smart\\_Grid\\_Topic\\_2\\_Smart\\_Grid.pdf](http://www.ee.ucr.edu/~hamed/Smart_Grid_Topic_2_Smart_Grid.pdf).
- [2] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery in Databases," *AI Mag.*, vol. 17, no. 3, p. 37, 1996.
- [3] P. Berkhin, "Grouping Multidimensional Data: Recent Advances in Clustering," J. Kogan, C. Nicholas, and M. Teboulle, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 25–71.
- [4] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York City: Springer New York Heidelberg Dordrecht London, 2013.
- [5] B. C. Becker and E. G. Ortiz, "Evaluation of face recognition techniques for application to facebook," *2008 8th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2008.
- [6] S. Balaban, "Deep Learning and Face Recognition: The State of the Art," *Biometrick Surveill. Technol. Hum. Act. Identif.*, vol. 12, 2015.
- [7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of Recommendation Algorithms for E-Commerce ABSTRACT," 2000.
- [8] M. Khanna, "Data Mining in Smart Grids-A Review," vol. 5, no. 3, pp. 709–712, 2015.
- [9] K. Le Zhou, S. L. Yang, and C. Shen, "A review of electric load classification in smart grid environment," *Renew. Sustain. Energy Rev.*, vol. 24, pp. 103–110, 2013.
- [10] J. Guckenheimer, T. J. Overbye, D. Bienstock, A. Bose, T. Boston, J. Dagle, M. D. Ilic, C. K. Jones, F. P. Kelly, Y. G. Kevrekidis, R. D. Masiello, J. C. Meza, C. Rudin, R. J. Thomas, M. H. Wright, and Committee on Analytical Research Foundations for the Next-Generation Electric Grid, *Analytic Research Foundations for the Next-Generation Electric Grid*. 2016.

- [11] A. Mukherjee, S. Member, R. Vallakati, and S. Member, "Situational Awareness Framework for openPDC Datasets," pp. 1–8.
- [12] A. Pal, J. S. Thorp, T. Khan, and S. S. Young, "Classification Trees for Complex Synchronphasor Data," *Electr. Power Components Syst.*, vol. 41, no. 14, pp. 1381–1396, 2013.
- [13] K. Le Zhou, S. L. Yang, and C. Shen, "A review of electric load classification in smart grid environment," *Renew. Sustain. Energy Rev.*, vol. 24, pp. 103–110, 2013.
- [14] G. Chicco, "Overview and performance assessment of the clustering methods for electrical load pattern grouping," *Energy*, vol. 42, no. 1, pp. 68–80, 2012.
- [15] R. Sánchez-García, M. Fennelly, S. Norris, N. Wright, G. Niblo, J. Brodzki, and J. Bialek, "Hierarchical Spectral Clustering of Power Grids," *IEEE Trans. Power Syst.*, vol. 29, no. 5, pp. 2229–2237, 2014.
- [16] C. G. Wang, B. H. Zhang, Z. G. Hao, J. Shu, P. Li, and Z. Q. Bo, "A novel real-time searching method for power system splitting boundary," *IEEE Trans. Power Syst.*, vol. 25, no. 4, pp. 1902–1909, 2010.
- [17] S. Blumsack, P. Hines, M. Patel, C. Barrows, and E. C. Sanchez, "Defining power network zones from measures of electrical distance," *2009 IEEE Power Energy Soc. Gen. Meet. PES '09*, pp. 1–8, 2009.
- [18] D. T. Nguyen, Y. Shen, and M. T. Thai, "Detecting critical nodes in interdependent power networks for vulnerability assessment," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 151–159, 2013.
- [19] Y.-S. Li, D.-Z. Ma, H.-G. Zhang, and Q.-Y. Sun, "Critical Nodes Identification of Power Systems Based on Controllability of Complex Networks," *Appl. Sci.*, vol. 5, no. 3, pp. 622–636, 2015.

- [20] E. Bompard, E. Pons, L. Luo, and M. Rosas Casals, "A Perspective overview of topological approaches for vulnerability analysis of power transmission grids," *Int. J. Crit. Infrastructures*, vol. 11, no. JANUARY, 2015.
- [21] P. Panigrahi, "Topological Analysis of Power Grid to Identify Vulnerable Transmission Lines and Nodes Topological Analysis of Power Grid to Identify Vulnerable Transmission Lines and Nodes Master of Technology Control & Automation Prof . Somnath Maity," no. May, 2013.
- [22] M. Bairey and S. Stowell, "US Power Grid Network Analysis," 2014.
- [23] M. Parvizmosaed, F. Farmani, H. Monsef, and A. Rahimi-Kian, "A multi-stage Smart Energy Management System under multiple uncertainties: A data mining approach," *Renew. Energy*, vol. 102, pp. 178–189, 2017.
- [24] X. Pan, X. Niu, X. Yang, B. Jacquet, and D. Zheng, "Microgrid energy management optimization using model predictive control: A case study in China," *IFAC-PapersOnLine*, vol. 48, no. 30, pp. 306–311, 2015.
- [25] D. Wang, H. Luo, O. Grunder, Y. Lin, and H. Guo, "Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm," *Appl. Energy*, vol. 190, pp. 390–407, 2017.
- [26] B. Yildiz, J. I. Bilbao, and A. B. Sproul, "A review and analysis of regression and machine learning models on commercial building electricity load forecasting," *Renew. Sustain. Energy Rev.*, vol. 73, no. March 2016, pp. 1104–1122, 2017.
- [27] A. I. Saleh, A. H. Rabie, and K. M. Abo-Al-Ez, "A data mining based load forecasting strategy for smart electrical grids," *Adv. Eng. Informatics*, vol. 30, no. 3, pp. 422–448, 2016.
- [28] X. Pan, X. Niu, X. Yang, B. Jacquet, and D. Zheng, "Microgrid energy management optimization using model predictive control: A case study in China," *IFAC-PapersOnLine*,

- vol. 48, no. 30, pp. 306–311, 2015.
- [29] R. A. Chinnathambi, “Investigation of forecasting methods for the hourly spot price of the Day-Ahead Electric Power Markets,” in *2016 IEEE International Conference on Big Data*, 2016, pp. 3079–3086.
- [30] X. Zhang, J. Wang, and K. Zhang, “Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by Cuckoo search algorithm,” *Electr. Power Syst. Res.*, vol. 146, pp. 270–285, 2017.
- [31] A. . Jain, M. . Murty, and P. J. Flynn, “Data Clustering: A Review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.
- [32] J. Paparrizos and L. Gravano, “k-Shape: Efficient and Accurate Clustering of Time Series,” *Acm Sigmod*, pp. 1855–1870, 2015.
- [33] D. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series,” *Work. Knowl. Knowl. Discov. Databases*, vol. 398, pp. 359–370, 1994.
- [34] J. C. Dunn, “Well-Separated Clusters and Optimal Fuzzy Partitions,” *J. Cybern.*, vol. 4, no. 1, 1974.
- [35] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J. Comput. Appl. Math.*, vol. 20, no. C, pp. 53–65, 1987.
- [36] D. L. Davies and D. W. Bouldin, “A Cluster Separation Measure,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, 1979.
- [37] S. Saitta, B. Raphael, and I. Smith, *A Bounded Index for Cluster Validity*. Springer, 2007.
- [38] T. Calinski and J. Harabasz, “A Dendrite Method for Cluster Analysis,” *Commun. Stat.*, vol. 3, no. 1, 1974.
- [39] I. Gurrutxaga, I. Albisua, O. Arbelaitz, J. I. Martín, J. Muguerza, J. M. Pérez, and I. Perona,

- “SEP/COP: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index,” *Pattern Recognit.*, vol. 43, no. 10, pp. 3364–3373, 2010.
- [40] J. Yang, C. Ning, C. Deb, F. Zhang, D. Cheong, S. E. Lee, C. Sekhar, and K. W. Tham, “k-Shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement,” *Energy Build.*, vol. 146, pp. 27–37, 2017.
- [41] U. Maulik and S. Bandyopadhyay, “Performance evaluation of some clustering algorithms and validity indices,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1650–1654, 2002.
- [42] S. P. Lloyd, “Least Squares Quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [43] W. S. Sarle, A. K. Jain, and R. C. Dubes, “Algorithms for Clustering Data,” *Technometrics*, vol. 32, no. 2, p. 227, 1990.
- [44] W. S. Sarle, A. K. Jain, and R. C. Dubes, “Algorithms for Clustering Data,” *Technometrics*, vol. 32, no. 2, p. 227, 1990.
- [45] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” *Proc. 2nd Int. Conf. Knowl. Discov. Data Min.*, pp. 226–231, 1996.
- [46] J. Gao, “Clustering Lecture 2: Partitional Methods,” *SUNY Buffalo*. [Online]. Available: [https://www.cse.buffalo.edu/~jing/cse601/fa12/materials/clustering\\_partitional.pdf](https://www.cse.buffalo.edu/~jing/cse601/fa12/materials/clustering_partitional.pdf).
- [47] B. Guoqiang Zhang and M. Y. H. Eddy Patuwo, “Forecasting with Artificial Neural Networks: The State of the Art,” *Int. J. Forecast.*, vol. 14, pp. 35–62, 1998.
- [48] R. D. Tobias, “An introduction to partial least squares regression,” *Proc. Ann. SAS Users Gr. Int. Conf., 20th, Orlando, FL*, pp. 2–5, 1995.



- [49] R. Wehrens, “The pls Package: Principal Component and Partial Least Squares Regression in R,” *J. Stat. Softw.*, vol. 18, no. 2, 2007.
- [50] R. Nau, “Statistical Forecasting: Notes on Regression and Time Series Analysis,” 2017. [Online]. Available: <http://people.duke.edu/~rnau/411home.htm>.
- [51] K. Metaxiotis, A. Kagiannas, D. Askounis, and J. Psarras, “Artificial intelligence in short term electric load forecasting: a state-of-the-art survey for the researcher,” *Energy Convers. Manag.*, vol. 44, no. 9, pp. 1525–1534, 2003.
- [52] R. J. Hyndman, “auto.arima,” *RDocumentation*, 2017. [Online]. Available: <https://www.rdocumentation.org/packages/forecast/versions/7.3/topics/auto.arima>.
- [53] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, “STL: A Seasonal-Trend Decomposition Procedure Based on Loess,” *J. Off. Stat.*, vol. 6, pp. 3–73, 1990.
- [54] R. B. Cleveland, W. S. Cleveland, and I. Terpenning, “STL: A Seasonal-Trend Decomposition Procedure Based on Loess,” *J. Off. Stat.*, vol. 6, no. 1, 1990.
- [55] K. Ruohonen, “Graph theory,” p. 108, 2013.
- [56] L. L. Grigsby, D. R. Tobergte, and S. Curtis, *Power Systems*, 2nd ed., vol. 53, no. 9. New York City: CRC Press, 2007.
- [57] R. Diestel, *Graph Theory (Graduate Texts in Mathematics)*. 2000.
- [58] L. C. Freeman, “A Set of Measures of Centrality Based on Betweenness,” *Am. Sociol. Assoc.*, vol. 40, no. 1, pp. 35–41, 1977.
- [59] G. B. Dantzig, “On the Shortest Path Route Through a Network,” *Manage. Sci.*, no. 6, pp. 187–190, 1960.
- [60] M. Girvan and M. E. J. Newman, “Community Structure in Social and Biological Networks,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, no. 12, pp. 7821–6, 2002.

- [61] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 69, no. 6 2, pp. 1–5, 2004.
- [62] S. E. Schaeffer, "Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, 2007.
- [63] M. Girvan and M. E. J. Newman, "Finding and evaluating community structure in networks," *Cond-Mat/0308217*, pp. 1–16, 2003.
- [64] M. E. J. Newman, "Modularity and Community Structure in Networks.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, no. 23, pp. 8577–82, 2006.
- [65] H. Shiokawa, Y. Fujiwara, and M. Onizuka, "Fast Algorithm for Modularity-Based Graph Clustering," *Proceeding Twenty-Seventh Conf. Artif. Intell.*, pp. 1170–1176, 2013.
- [66] R. Gentleman and R. Ihaka, "R: The R Project for Statistical Computing," *R-project.org*, 2016. [Online]. Available: <https://www.r-project.org/>.
- [67] H. Xiao, Y. Huimei, W. Chen, and L. Hongjun, "A survey of influence of electric vehicle charging on power grid," *Proc. 2014 9th IEEE Conf. Ind. Electron. Appl. ICIEA 2014*, pp. 121–126, 2014.
- [68] "Global Plug-In Deliveries for Q3-2017 and YTD," *EVVolumes.com*, 2017. [Online]. Available: <http://www.ev-volumes.com/>.
- [69] M. Panteli and D. S. Kirschen, "Situation awareness in power systems: Theory, challenges and applications," *Electr. Power Syst. Res.*, vol. 122, pp. 140–151, 2015.
- [70] A. G. Phadke and J. S. Thorp, "Synchronized Phasor Measurements and their Applications," *Springer*, p. 246, 2008.
- [71] A. Abur and F. Galvan, "Synchro-phasor assisted state estimation (SPASE)," in *2012 IEEE PES Innovative Smart Grid Technologies, ISGT 2012*, 2012.
- [72] L. Zhao and a. Abur, "Multi area state estimation using synchronized phasor

- measurements,” *Power Syst. IEEE Trans.*, vol. 20, no. 2, pp. 611–617, 2005.
- [73] S. Chakrabarti and E. Kyriakides, “Optimal placement of phasor measurement units for state estimation,” *Proc. IASTED Int. Conf. Energy Power Syst.*, pp. 73–78, 2007.
- [74] H. P. Oak, “A Survey on Short Term Load Forecasting,” 2015.
- [75] M. Al Karim, M. Chenine, K. Zhu, and L. Nordstrom, “Synchrophasor-based data mining for power system fault analysis,” in *IEEE PES Innovative Smart Grid Technologies Conference Europe*, 2012.
- [76] A. H. Al-Mohammed and M. A. Abido, “An adaptive fault location algorithm for power system networks based on synchrophasor measurements,” *Electr. Power Syst. Res.*, vol. 108, pp. 153–163, 2014.
- [77] B. Singh, N. Sharma, A. Tiwari, K. Verma, and S. Singh, “Applications of phasor measurement units (PMUs) in electric power system networks incorporated with FACTS controllers,” *Int. J. Eng. Sci. Technol.*, vol. 3, no. 3, 2011.
- [78] S. Dasgupta, M. Paramasivam, U. Vaidya, and V. Ajarapu, “Real-time monitoring of short-term voltage stability using PMU data,” *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 3702–3711, 2013.
- [79] C. S. Yu, C. W. Liu, S. L. Yu, and J. A. Jiang, “A new PMU-based fault location algorithm for series compensated lines,” *IEEE Trans. Power Deliv.*, vol. 17, no. 1, pp. 33–46, 2002.
- [80] G. Kron, “Diakoptics: The Piecewise Solution of Large-Scale Systems.,” 1963, vol. 2.
- [81] M. Rosas-Casals and B. Corominas-Murtra, “Assessing European power grid reliability by means of topological measures,” *WIT Trans. Ecol. Environ.*, vol. 121, pp. 515–525, 2009.
- [82] X. Chen, “Critical nodes identification in complex systems,” *Complex Intell. Syst.*, vol. 1, no. 1–4, pp. 37–56, 2015.

- [83] P. Hage and F. Harary, "Eccentricity and centrality in networks," *Soc. Networks*, vol. 17, no. 1, pp. 57–63, 1995.
- [84] Y. Levron, J. M. Guerrero, and Y. Beck, "Optimal power flow in microgrids with energy storage," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 3226–3234, 2013.
- [85] M. Y. Hassan, "Application of Particle Swarm Optimization for Solving Optimal Generation Plant Location Problem," *Int. J. Electr. Electron. Syst. Res.*, vol. 5, 2012.
- [86] Z. M. Roozegar, A. Kazemzadeh, and R. Kauffmann, "Two Area Power Systems Economic Dispatch Problem Solving Considering Transmission Capacity," *Power*, vol. 12, 2007.
- [87] A. Sudhakar, "Multi Area Economic Dispatch with Tie Line Loss Using Secant Method and Tie Line Matrix," *Int. J. Appl. Power Eng.*, vol. 2, no. 3, pp. 115–124, 2013.
- [88] D. Streiffert, "Multi-area Economic Dispatch with Tie Line Constraints," *IEEE Trans. Power Syst.*, vol. 10, no. 4, pp. 1946–1951, 1995.
- [89] H. Alfares and M. Nazeeruddin, "Electric Load Forecasting: Literature Survey and Classification," *Int. J. Syst. Sci.*, pp. 23–24, 2010.
- [90] K. Metaxiotix, A. Kagiannas, D. Askounis, and J. Psarras, "Artificial intelligence in short term electric load forecasting: a state-of-the art survey for the researcher," *Energy Convers. Manag.*, vol. 44, no. 9, pp. 1525–1534, 2003.
- [91] T. Hong, T. Laing, and P. Wang, "Four Best Practices of Load Forecasting for Electric Cooperatives," in *2014 IEEE Rural Electric Power Conference*, 2014.
- [92] E. Almeshaiei and H. Soltan, "A methodology for Electric Power Load Forecasting," *Alexandria Eng. J.*, vol. 50, no. 2, pp. 137–144, 2011.
- [93] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: the forecast package for R Automatic time series forecasting: the forecast package for R," *J. Stat.*

*Softw.*, vol. 27, no. 3, pp. 1–22, 2008.

- [94] H. Hahn, S. Meyer-Nieberg, and S. Pickl, “Electric load forecasting methods: Tools for decision making,” *Eur. J. Oper. Res.*, vol. 199, no. 3, pp. 902–907, 2009.
- [95] J. D. Rhodes, W. J. Cole, C. R. Upshaw, T. F. Edgar, and M. E. Webber, “Clustering analysis of residential electricity demand profiles,” *Appl. Energy*, vol. 135, pp. 461–471, 2014.
- [96] T. K. Wijaya, S. Humeau, M. Vasirani, and K. Aberer, “Individual, Aggregate, and Cluster-based Aggregate Forecasting of Residential Demand,” *Lausanne, Switzerland, Tech. Rep*, 2014.
- [97] T. K. Wijaya, M. Vasirani, S. Humeau, and K. Aberer, “Cluster-based aggregate forecasting for residential electricity demand using smart meter data,” *2015 IEEE Int. Conf. Big Data (Big Data)*, pp. 879–887, 2015.
- [98] National Renewable Energy Laboratory, “Impact of uncoordinated plug-in electric vehicle charging on residential power demand - supplementary data,” 2018. [Online]. Available: <https://data.nrel.gov/submissions/69>.

## APPENDIX I – R CODE FOR CLUSTERING IN PMU DATASETS

```
library(shiny)

library(cluster)

library(fpc)

library(DMwR)

library(c1Valid)

start.time <- Sys.time()

data$X<-NULL

colnames(data) <- c("MeanFrequency", "Time")

data <- na.omit(data)

Logger <- data

resultsK <-kmeans(Logger$MeanFrequency,3)

cluster1<-data[resultsK$cluster==1,]

cluster2<-data[resultsK$cluster==2,]

cluster3<-data[resultsK$cluster==3,]

#Cluster1

max1<-max(cluster1$MeanFrequency)

min1 <-min(cluster1$MeanFrequency)

d1<- max1-min1

mean1 <-mean(cluster1$MeanFrequency)

#Cluster2

max2<-max(cluster2$MeanFrequency)
```

```

min2<-min(cluster2$MeanFrequency)

d2<- max2-min2

mean2<-mean(cluster2$MeanFrequency)

c21<-mean2-mean1

#Cluster3

max3<-max(cluster3$MeanFrequency)

min3<-min(cluster3$MeanFrequency)

d3<- max3-min3

mean3<-mean(cluster3$MeanFrequency)

c31<-mean3-mean1

c32<-mean3-mean2

#Combine

ds <- c(d1,d2,d3)

#View(ds)

Dunnss<-c(1000)

centersD<- c(c21,c31,c32)

#View(centersD)

for(i in 1:3){

  for (j in 1:3){

    c<- centersD[i]/ds[j]

    Dunnss<- list(Dunnss,c)

  }
}

```

```

}

l<-unlist(Dunnss)

Dunnss<-min(abs(l))

View(Dunnss)

valuK <- resultsK$cluster

#View(valuK)

valu1K <- as.data.frame(valuK)

#View(valu1K$valuK)

#View(Logger)

#View(Logger[valuK, 1])

dfK = data.frame(Logger$MeanFrequency, valu1K$valuK)

colnames(dfK) <- c("Voltage", "Cluster")

par(xpd=NA,oma=c(0,0,0,10))

plot(dfK$Cluster, dfK$Voltage, col = resultsK$cluster, main = "K-Means Clustering Current Magnitude", xlab =
"Cluster", ylab = "Current Magnitude", cex.axis = 1.5)

points(resultsK$centers, pch = "x")

legend(par("usr")[2],par("usr")[4],title="Elements of Clusters",col=c("black","red","green"),
c(toString(resultsK$size[1]),toString(resultsK$size[2]),toString(resultsK$size[3])),pch = 1,lty=0,xjust=0, yjust=1.0)

legend(par("usr")[2],par("usr")[4],title="Centroids of Clusters",col=c("black","red","green")
,c(toString(signif(resultsK$centers[1],digits=5)),toString(signif(resultsK$centers[2],digits=5)),toString(signif(results
K$centers[3], digits=5))),pch = 1,lty=0,xjust=0, yjust=2.0)

legend(par("usr")[2],par("usr")[4],title="Max of Clusters",col=c("black","red","green")
,c(toString(signif(max1,digits=5)),toString(signif(max2,digits=5)),toString(signif(max3,
digits=5))),pch
=1,lty=0,xjust=0, yjust=3.0)

```



```

legend(par("usr")[2],par("usr")[4],title="Min          of          Clusters",col=c("black","red","green")
,c(toString(signif(min1,digits=5)),toString(signif(min2,digits=5)),toString(signif(min3,          digits=5))),pch
=1,lty=0,xjust=-1, yjust=3.0)

legend(par("usr")[2],par("usr")[4],title="Dunns Index",c(toString(signif(Dunnss,digits=5))),lty=0,xjust=0, yjust=6.0)

end.time <- Sys.time()

time.taken <- end.time - start.time

View(time.taken)

###dbscan

library(shiny)

library(cluster)

library(fpc)

library(DMwR)

#library(cIValid)

start.time <- Sys.time()

#Get Max,Min, and Distances

index<-1

while(index<numClus){

  cluster<-data[db$cluster==index,]

  maxVal<-max(cluster$MeanFrequency)

  minVal<-min(cluster$MeanFrequency)

  meanVal<-mean(cluster$MeanFrequency)

  disInner<-maxVal-minVal

```

```

dis<-list(dis,disInner)

max<-list(max,maxVal)

min<-list(min,minVal)

mean<-list(mean, meanVal)

if(index>1){

  clustT<-cbind(clustT,cluster$MeanFrequency)

}

#View(index)

index<-index+1

}

clustT<-as.data.frame(clustT)

#View(clustT)

min<-as.data.frame(min)

max<-as.data.frame(max)

mean<-as.data.frame(mean)

dis<-as.data.frame(dis)

min[1]<-NULL

max[1]<-NULL

mean[1]<-NULL

dis[1]<-NULL

totmin<-t(min)

totmax<-t(max)

```

```

totmea<-t(mean)

#total<-as.data.frame(totmea,totmin,totmax)

#View(total)

rownames(totmin)<-NULL

rownames(totmax)<-NULL

rownames(totmea)<-NULL

colnames(totmin) <- c("Min")

colnames(totmax) <- c("Max")

colnames(totmea) <- c("Mean")

total<-cbind(totmea,totmax,totmin)

#Get distances between all center means

d<-dist(t(mean))

#Take out values

cenD<-unique(d)

cenD<-as.data.frame(cenD)

#View(cenD)

num<-(numClus-1)

if(num > 3) {

  cenD<-cenD[-c(1:num),]

}

Dunns<-min(cenD)

par(xpd=NA,oma=c(0,0,0,10))

```

```
plot(data$MeanFrequency, col=db$cluster+1L, main="DBSCAN Current eps=5, MinPts=50", ylab="Current  
Magnitude")  
  
legend(par("usr")[2],par("usr")[4],title="Dunns Index",c(toString(signif(Dunns,digits=4))),lty=0,xjust=0, yjust=1.0)  
  
end.time <- Sys.time()  
  
time.taken <- end.time - start.time  
  
View(time.taken)
```

## APPENDIX II – R CODE FOR CLUSTERING BASED MICROGRID DECOMPOSITION

```
library(igraph)

#ieee300common is bus dataset IEEE 300 bus

ic3=ieee300common[-301,]

i300=graph.data.frame(ieee300linecommon,directed=F,vertices=ieee300common)

ic=graph.data.frame(ieee300linecommon,directed=F,vertices=ic3)

plot(i300,vertex.size=7,vertex.label=NA)

layout <- layout.reingold.tilford(i300, circular=F)

plot(i300, vertex.size=7, vertex.label.cex=.5)

## admittance

E(i300)$Admittance=1/(((ieee300linecommon$BranchResistance)^2+(ieee300common$BranchReactanceX)^2)^(1/2))

E(i300)$Length=abs(ieee300linecommon$BranchReactanceX)*260.36

#E(ic)$Admittance=1/(((ic3$BranchResistance)^2+(ic3$BranchReactanceX)^2)^(1/2))

gens300=which(ieee300common$GenerationMW!=0)

V(i300)$shape="circle"

V(i300)$shape[gens300]="sphere"

## clustering

lengthclust300=cluster_edge_betweenness(i300,weights=E(i300)$Length)

plot_dendrogram(lengthclust300)
```

```

modularity(lengthclust300)

V(i300)$color=membership(lengthclust300)

plot(i300, vertex.size=7, vertex.label.cex=.5,main="Length Betweenness")

clust300=cluster_edge_betweenness(i300)

plot_dendrogram(clust300)

modularity(clust300)

#betweenness admit

admitclust300=cluster_edge_betweenness(i300,weights=E(i300)$Admittance)

plot_dendrogram(admitclust300)

modularity(admitclust300)

##### 300 BUS SYSTEM CRITICAL NODE ANALYSIS

Critical300$BDd=Critical300$NormalizedImpedBet*Critical300$Degree*Critical300$LoadMW

Critical300$BDg=Critical300$NormalizedImpedBet*Critical300$Degree*abs(Critical300$GenerationMW)

View(Critical300)

Critical300a=Critical300[-301,]

### 300 bus indices are columns 24,28,29

ind=c(24,28,29)

thing=data.frame(NB=double(),NDBd=double(),NDBdg=double())

thing[1:300,1:3]=0

num=1

for (i in ind){

  thing[,num]=(Critical300a[,i] - min(Critical300a[,i], na.rm=TRUE)) /

```

```

(max(Critical300a[,i],na.rm=TRUE) - min(Critical300a[,i], na.rm=TRUE))

num=num+1

}

###

#### # # # # 300 bus system

##

Critical300=ieee300common

bus300=graph.data.frame(ieee300linecommon,directed=F,vertices=ieee300common)

E(bus300)$Admittance=1/(((ieee300linecommon$BranchResistance)^2+(ieee300linecommon$BranchReactanceX)^
2)^(1/2))

E(bus300)$Length=abs(ieee300linecommon$BranchReactanceX)*260.36

E(bus300)$AdmitInv=1/E(bus300)$Admittance

Critical300$Degree=degree(bus300)

NormalizedAdmitBet=betweenness(bus300,normalized=TRUE,weights=E(bus300)$Admittance)

NormalizedImpedBet=betweenness(bus300,normalized=TRUE,weights=E(bus300)$AdmitInv)

Critical300$NormalizedAdmitBet=NormalizedAdmitBet

Critical300$NormalizedImpedBet=NormalizedImpedBet

### DISTANCE TO NEAREST GENERATOR

distMatrix <- shortest.paths(bus300, v=V(bus300), to=V(bus300),weights=E(bus300)$AdmitInv)

gens3=which(ieee300common$GenerationMW>0)

gens300=ieee300common$BusNumber[which(ieee300common$GenerationMW>0)]

shortgen=rep(0,301)

genID=rep(0,301)

```

```

for (i in 1:300) {

  shortgen[i]=min(distMatrix[i, gens3])

  shortgen[i]

  ID=which(distMatrix[i, gens3]==shortgen[i])

  genID[i]=gens300[ID]

  genID[i]

}

Critical300$NearestGenImpedDistance=shortgen

Critical300$NearestGen=genID

#View(Critical300)

#### plotting attributes and plotting

V(bus300)$color=Critical300$NearestGen

V(bus300)$shape="circle"

V(bus300)[gens3]$shape="sphere"

V(bus300)[store300]$shape="sphere"

V(bus300)[relay300]$shape="square"

plot(bus300, vertex.size=10, vertex.label.cex=.6)

#View(Critical300)

#### converge and do second stage clustering

##admitclust$membership=Critical300$NearestGen

##membership(admitclust)

g=unique(Critical300$NearestGen)

```



```

o=order(g)

for (i in 1:nrow(Critical300)) {

  thing=which(g==Critical300$NearestGen[i])

  Critical300$NGID[i]=thing

  Critical300$NGID[i]

}

#View(Critical300)

converged300=contract.vertices(bus300,Critical300$NGID)

converged300= simplify(converged300, remove.loops=FALSE)

plot(converged300, vertex.label.cex=.65, main="Nearest Generator Converged")

twostage300=cluster_edge_betweenness(converged300)

convergedtwostage=contract.vertices(converged300,membership(twostage300))

plot(convergedtwostage, vertex.label.cex=.65, main="Two-Stage NearestGen + GN")

plot(convergedtwostage, vertex.label=NA, main="Two-Stage NearestGen + GN")

#V(bus300)$color=twostage300$membership

#plot(bus300, vertex.size=10, vertex.label.cex=.6, main="Two-Stage Nearest+GN")

##### final assignment tracking

finalassign=rep(0,301)

for (i in 1:301){

  lp=Critical300$NGID[i]

  lp

  finalassign[i]=twostage300$membership[lp]

```

```
finalassign[i]
}
finalassignlist=list()
for (i in 1:max(finalassign)){
  finalassignlist[[i]]=ieee300common$BusNumber[which(finalassign==i)]
}
V(bus300)$color=finalassign
finalassign300=finalassign
plot(bus300, vertex.size=10, vertex.label.cex=.6, main="Two-Stage Nearest+GN")
```

## APPENDIX III – R CODE FOR CLUSTERING SMART METER DATA AND LOAD FORECASTING

```
library(dtwclust)
```

```
library(TSclust)
```

```
library(ggplot2)
```

```
library(stats)
```

```
library(forecast)
```

```
library(caret)
```

```
library(zoo)
```

```
library(dtw)
```

```
library(cluster)
```

```
library(reshape)
```

```
library(reshape2)
```

```
library(tidyr)
```

```
library(kml)
```

```
June1h=summerh[which(summerh$Time<"2010-06-02" & summerh$Time>="2010-06-01"),]
```

```
June1NEh=summerNEh[which(summerNEh$Time<"2010-06-02" & summerNEh$Time>="2010-06-01"),]
```

```
fdate=which(summerh$Time=="2010-06-01")
```

```
past7h=c((fdate-(7*24)):fdate)
```

```
sumpast7h=summerh[past7h,]
```

```
fdateNE=which(summerNEh$Time=="2010-06-01")
```

```
past7NEh=c((fdateNE-(7*24)):fdateNE)
```

```
sumpast7NEh=summerNEh[past7NEh,]
```

```

#days7h=Mayh[which(Mayh$Time="2010-06-02"),]

#days7NEh=MayNEh[which(MayNEh$Time<"2010-06-02" & MayNEh$Time>="2010-06-01"),]

sample=sumpast7NEh[,3:202]

sample=t(sample)

#hclust5=tsclust(series=sample,type="hierarchical",k=5, distance="dtw")

d72NE=tsclust(series=sample,type="partitional",k=2,distance="dtw")

d73NE=tsclust(series=sample,type="partitional",k=3,distance="dtw")

d74NE=tsclust(series=sample,type="partitional",k=4,distance="dtw")

d75NE=tsclust(series=sample,type="partitional",k=5,distance="dtw")

d76NE=tsclust(series=sample,type="partitional",k=6,distance="dtw")

k72NE=tsclust(series=sample,type="partitional",preproc=zscore,distance="sbd",centroid="shape") ## this is the k-
shape algorithm

k73NE=tsclust(series=sample,type='partitional',k=3,preproc=zscore,distance='sbd',centroid='shape')

k74NE=tsclust(series=sample,type="partitional",k=4,preproc=zscore,distance="sbd",centroid="shape") ## this is the
k-shape algorithm

k75NE=tsclust(series=sample,type="partitional",k=5,preproc=zscore,distance="sbd",centroid="shape") ## this is the
k-shape algorithm

k76NE=tsclust(series=sample,type="partitional",k=6,preproc=zscore,distance="sbd",centroid="shape") ## this is the
k-shape algorithm

##### with electric vehicles

sample=sumpast7h[,3:202]

sample=t(sample)

```

```

#hclust5=tsclust(series=sample,type="hierarchical",k=5, distance="dtw")

d72h=tsclust(series=sample,type="partitional",k=2,distance="dtw")

d73h=tsclust(series=sample,type="partitional",k=3,distance="dtw")

d74h=tsclust(series=sample,type="partitional",k=4,distance="dtw")

d75h=tsclust(series=sample,type="partitional",k=5,distance="dtw")

d76h=tsclust(series=sample,type="partitional",k=6,distance="dtw")

k72h=tsclust(series=sample,type="partitional",preproc=zscore,distance="sbd",centroid="shape") ## this is the k-
shape algorithm

k73h=tsclust(series=sample,type='partitional',k=3,preproc=zscore,distance='sbd',centroid='shape')

k74h=tsclust(series=sample,type="partitional",k=4,preproc=zscore,distance="sbd",centroid="shape") ## this is the k-
shape algorithm

k75h=tsclust(series=sample,type="partitional",k=5,preproc=zscore,distance="sbd",centroid="shape")

k76h=tsclust(series=sample,type="partitional",k=6,preproc=zscore,distance="sbd",centroid="shape") ## this is the k-
shape algorithm

##### clustering analysis and plotting #####

D7hCVI=rbind(cvi(d72h),cvi(d73h),cvi(d74h),cvi(d75h),cvi(d76h))

rownames(D7hCVI)=c('d7h2','d7h3','d7h4','d7h5','d7h6')

D7NECVI=rbind(cvi(d72NE),cvi(d73NE),cvi(d74NE),cvi(d75NE),cvi(d76NE))

rownames(D7NECVI)=c('d7NE2','d7NE3','d7NE4','d7NE5','d7NE6')

K7NECVI=rbind(cvi(k72NE),cvi(k73NE),cvi(k74NE),cvi(k75NE),cvi(k76NE))

rownames(K7NECVI)=c('K7NE2','K7NE3','K7NE4','K7NE5','K7NE6')

```

```

K7HCVI=rbind(cvi(k72h),cvi(k73h),cvi(k74h),cvi(k75h),cvi(k76h))

rownames(K7HCVI)=c('k7h2','k7h3','k7h4','k7h5','k7h6')

##### sumdev

sD7hCVI=as.data.frame(scale(D7hCVI))

sD7hCVI[is.na(sD7hCVI)]=0

sD7hCVI$Sumdev=rowSums(sD7hCVI)

sD7NECVI=as.data.frame(scale(D7NECVI))

sD7NECVI[is.na(sD7NECVI)]=0

sD7NECVI$Sumdev=rowSums(sD7NECVI)

sK7NECVI=as.data.frame(scale(K7NECVI))

sK7NECVI[is.na(sK7NECVI)]=0

sK7NECVI$Sumdev=rowSums(sK7NECVI)

sK7hCVI=as.data.frame(scale(K7HCVI))

sK7hCVI[is.na(sK7hCVI)]=0

sK7hCVI$Sumdev=rowSums(sK7hCVI)

#####3 based on cluster validity sum of scaled indices

D2hmem=as.numeric(unlist(d72h@cluster))

D3hmem=as.numeric(unlist(d73h@cluster))

DNE2mem=as.numeric(unlist(d72NE@cluster))

DNE5mem=as.numeric(unlist(d75NE@cluster))

d21=which(D2hmem==1)

```

d22=which(D2hmem==2)

d31=which(D3hmem==1)

d32=which(D3hmem==2)

d33=which(D3hmem==3)

dne21=which(DNE4mem==1)

dne22=which(DNE4mem==2)

dne51=which(DNE5mem==1)

dne52=which(DNE5mem==2)

dne53=which(DNE5mem==3)

dne54=which(DNE5mem==4)

dne55=which(DNE5mem==5)

#### can add the columns to put in the weekday and other things here too later

dh21=Mayh[,2+d21]

dh22=Mayh[,2+d22]

dh31=Mayh[,2+d31]

dh32=Mayh[,2+d32]

dh33=Mayh[,2+d33]

dNE21=MayNEh[,2+dne21]

dNE22=MayNEh[,2+dne22]

dNE51=MayNEh[,2+dne51]

```
dNE52=MayNEh[,2+dne52]
```

```
dNE53=MayNEh[,2+dne53]
```

```
dNE54=MayNEh[,2+dne54]
```

```
dNE55=MayNEh[,2+dne55]
```

```
holder1=list()
```

```
data=list(dh21,dh22,dNE21,dNE22,dNE51,dNE52,dNE53,dNE54,dNE55,dh31,dh32,dh33)
```

```
x=1
```

```
for (j in data) {
```

```
  for (i in 1:length(j[,1])) {
```

```
    j$cumulative[i]=sum(j[i,])
```

```
  } ### end sum and weekday loop
```

```
  holder1[[x]]=j
```

```
  x=x+1
```

```
} ### end data for loop
```

```
May7hd21=as.data.frame(holder1[[1]])
```

```
May7hd22=as.data.frame(holder1[[2]])
```

```
May7NEd21=as.data.frame(holder1[[3]])
```

```
May7NEd22=as.data.frame(holder1[[4]])
```

```
May7NEd51=as.data.frame(holder1[[5]])
```

```
May7NEd52=as.data.frame(holder1[[6]])
```



```
May7NEd53=as.data.frame(holder1[[7]])
```

```
May7NEd54=as.data.frame(holder1[[8]])
```

```
May7NEd55=as.data.frame(holder1[[9]])
```

```
May7hd31=as.data.frame(holder1[[10]])
```

```
May7hd32=as.data.frame(holder1[[11]])
```

```
May7hd33=as.data.frame(holder1[[12]])
```

```
##### forecasting
```

```
library(forecast)
```

```
tsholder=list()
```

```
stlholder=list()
```

```
fholder=list()
```

```
data=list(May7hd21,May7hd22,May7NEd21,May7NEd22,May7NEd51,May7NEd52,May7NEd53,May7NEd54,May7NEd55,May7hd31,May7hd32,May7hd33)
```

```
x=1
```

```
for (j in data) {
```

```
  for (i in 1:length(j[,1])) {
```

```
    ts=ts(j$cumulative,frequency=24)
```

```
    stl=stl(ts,s.window='periodic',t.window=480)
```

```
    f=forecast(stl,h=24,method='arima')
```

```
  } ### end sum and weekday loop
```

```
tsholder[[x]]=ts
```

```
stlholder[[x]]=stl
```

```

fholder[[x]]=f

x=x+1

}## end second for

fh21=as.numeric(fholder[[1]]$mean)

fh22=as.numeric(fholder[[2]]$mean)

fh2=fh21+fh22

fne21=as.numeric(fholder[[3]]$mean)

fne22=as.numeric(fholder[[4]]$mean)

fne2=fne21+fne22

fn51=as.numeric(fholder[[5]]$mean)

fn52=as.numeric(fholder[[6]]$mean)

fn53=as.numeric(fholder[[7]]$mean)

fn54=as.numeric(fholder[[8]]$mean)

fn55=as.numeric(fholder[[9]]$mean)

fn5=fn51+fn52+fn53+fn54+fn55

fh31=as.numeric(fholder[[10]]$mean)

fh32=as.numeric(fholder[[11]]$mean)

fh33=as.numeric(fholder[[12]]$mean)

```

```

fh3=fh31+fh32+fh33

hfh=c(jf[1:14],fh[15:20],jf[21:24])

plot(June1h$sum200,type='l',col="blue",lwd=3,xlab='hour',ylab='Load (W)',main='Household w/ EV Demand

Forecast vs Actual

Hybrid TSclust Method', xlim(min(c(fh21,fh22,June1h$sum200))), ylim(max(c(fh21,fh22,June1h$sum200))))

lines(fh22,col='red',lwd=3)

lines(fh21,col='green',lwd=3)

legend(locator(1),c("Actual", "Forecast22", "Forecast21"),pch=c(21,21),pt.bg=c("blue", "red", "green"))

### MAPE

APE=100*abs(hfh-June1h$sum200)/June1h$sum200

MAPE=mean(APE)

MAPE

plot(density(APE),main="Distribution of APE TSclust hybrid Method

Households w/ EV",col='green',lwd=3)

hist(APE)

### SSD

res=abs(jf-J1h$sum200)

S=res^2

SSD=sum(S)

###MSD

MSD=mean(S)

TSHh4metrics=cbind(MAPE,SSD,MSD)

```

```

rownames(TSHh4metrics)=c("EVhouseholdsk4")

##### plotting centroids and other stuff

#####

plot(k74h@centroids[[4]],type="l",col='orange',main="Centroids of 4-shape Clustering

Households w/ Electric Vehicles",

ylab='Centroid Z', xlab="Hourly sample, 1 week")

lines(k74h@centroids[[1]],col='green')

lines(k74h@centroids[[3]],col='blue')

lines(k74h@centroids[[2]],col='red')

plot(d74h@centroids[[3]],type="l",col='orange',main="Centroids of DTW4 Clustering

Households with Electric Vehicles",

ylab='Centroid Z', xlab="Hourly sample, 1 week")

lines(d74h@centroids[[1]],col='green')

lines(d74h@centroids[[4]],col='blue')

lines(d74h@centroids[[2]],col='red')

```