National College of Ireland

Real-Time Reduction of Micro Phasor Measurement Units and Noise Detection: California

MSc Research Project
Data Analytics

Barry Fitzgerald
x17161371

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Barry Fitzgerald
Student ID:	x17161371
Programme:	Data Analytics **Year:** 2019
Module:	MSc Research Project
Supervisor:	Dr. Catherine Mulwa
Submission Due Date:	……
Project Title:	Real-Time Reduction of Micro Phasor Measurement Units and Noise Detection: California
Word Count:	…… **Page Count**……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……

Date: ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Real-Time Reduction of Micro Phasor Measurement Units and Event Detection: California

Barry Fitzgerald
x17161371

Abstract

The electrical grid at present is going through one of its most fundamental changes where there is a high concentration of distributed renewable energy generation such as Wind Farms, Tidal Waves and Solar Power creating enormous intermittent flows of energy that can be fed back onto the power grid. Digital technology is being used to monitor the actual flow and consumption of electricity in real-time. To actively monitor the distribution network Micro-Phasor Measurement Units (μPMUs) are starting to be rolled out that create time-stamped Global Positioning System (GPS) synchrophasor data that actively monitors the state of the network in real-time. To actively monitor, and store this information this report provides a solution that addresses the dimensionality of synchrophasor data using Cassandra and Incremental Principal Components Analysis (IPCA) on a distributed system using Spark processing that reduces the dimensionality of the synchrophasor data from 15 dimensions to 1 principal component capturing over 98.9% of the variance for current and 2 components capturing over 96.5% for voltage. This approach captures over 96.5% of the energy without too much loss of information using five real-time μPMUs from the Lawrence Berkeley National Laboratory – Berkeley Lab[1] in the US. The resultant principal components are then clustered using DBSCAN to detect noise which can have a detrimental effect on dimensionality reduction. The resultant information can then be used to actively create a Wide Area Monitoring (WAM) system for the smart grid.

1 Introduction

This introduction section comprises the background history of the project, why it was chosen and why it is important for today's Smart Grid. This research report looks at the roll out of Phasor Measurement Units (PMUs) (Benmouyal *et al.*, 2014), in today's electrical grids and why they are an important part of the Smart Grid. Their function and application are analysed from the point of view of a Big Data Pipe Line. The rollout of Phasor Measurement Units (PMUs) across the globe in recent years is unprecedented where it is now possible to identify and isolate phenomenon that occurs on the transmission line in real-time. With the deregulation of the power industry, heterogeneous sources of energy are now available to supply energy into the electrical grid where once only the natural monopoly could operate. This occurrence brought forth its own problems due to the different intermittent sources as well

[1] https://www.lbl.gov/

as the distribution of these sources (Zhang, Huang and Bompard, 2018). The volume of data being collected and catalogued is now unprecedented and there are issues on how the data can be stored and what it's used for. So, a solution needs to be identified that can make use of this data, data that can be used in real-time and can provide actionable insights that can provide system operators with the information they need while limiting the storage costs. This report presents a research project to develop a real-time dimensional reduction big data pipeline that can be used to identify and isolate problems that occur on the distribution grid. It is estimated that power outages in the US cost the economy $119 billion a year and 92% of them have their origins in the distribution system which will be the focus of this project (Lacommare, Eto and Lawrence, 2004).

1.1 Project Background and Motivation

Phasor Measurement Units (PMUs) are global positioning system (GPS) satellite transmission synchronised devices used to measure the real-time transient nature of the dynamic taking place on the electrical grid (PHADKE and BI, 2018). They arose out of the need to improve the power system's performance in the face of catastrophic failures and the need to monitor the transient state of the grid with the introduction of heterogeneous supplies of power such as wind, solar and wave connected to the grid which was once the preserve of the national utility.

Real-world data such as speech, photographs and MRI scans need to be reduced to be used effectively and dimensional reduction is one option where the transformation of high-dimensional data into a meaningful representation of reduced dimensionality is ideal for real-time analysis. One traditional technique that is widely used is Principal Component Analysis (PCA), PCA techniques are essential due to the unprecedented volume of data being generated. Each PMU is capable of measuring 16 phasors with a 32-bit magnitude and 32-bit phase angle, recording 60 samples per second, 5,184,000 samples per day, this produces close to ~712 MB of data per PMU per day (Buyya *et al.*, 2016). For a μPMU operating on the distribution side, it will be double this figure producing close to 1.5GB of data per device.

In Figure 1 a representational image of a single sinusoidal with its phase is presented by (PHADKE and BI, 2018), where:

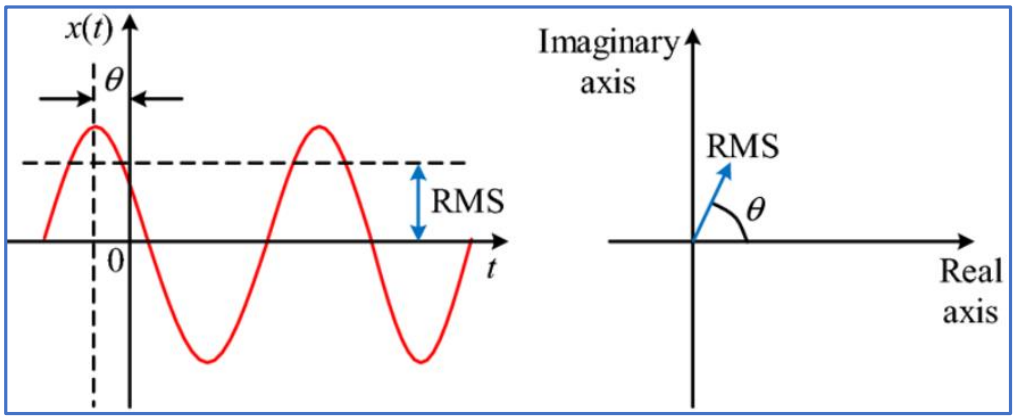


Figure 1: Sinusoidal representation and Phasor

t is the time, *RMS* is the root mean squared, which is just a measure of the magnitude of a time-varying signal and θ is the phase angle. In Equation 1, the phasor is estimated over one time period using the Discrete Fourier Transform where X_r and X_i are real and imaginary parts of the actual synchrophasor, x_n is the data sample, and N is the number of samples in one period.

$$\bar{X} = X_r + jX_i = \frac{\sqrt{2}}{N}\sum_{n=1}^{N}\left(x_n \cos\frac{2n\pi}{N} + jx_n \sin\frac{2n\pi}{N}\right)$$

Equation 1: Phasor representation using DFT

One major motivation is to apply reduction techniques to the μPMU dataset so that the high dimensional data can be transformed into a reduced subspace where the reduced representation has a dimensionality corresponding to the intrinsic dimensionality of the data so that noise can be reduced and the data can be visually inspected in real time. One other motivating factor that is applicable to big data is that by reducing the dimensionality, you also reduce the amount of memory and time available for data mining techniques. This problem has yet to be addressed and is only going to increase with the drive towards alternative sources of energy and the real-time observations of the electrical grid which from a business motivation perspective provide for unprecedented commercial opportunities such as improving customer service, asset performance, reduction in operational costs and improvements in network reliability (Bhuiyan, Khan and Murphy, 2017).

1.2 Project Requirement Specification

1.2.1 Research Question

The Research question addresses the need to observe, understand and manage the grid from the point of view of the distribution side, where advances in technology such as charge points for electrical vehicles are being rolled out. The problem of online dimensional reduction is addressed where the synchrophasors are measured at different geographical locations where each *n* measurement of *p* variables is of the same type and correspond to the same GPS time.
Why is there a need for reduction? First of all, μPMUs are high frequency devices that sample the voltage and current phasors from the analogue to the digital domain every 13-15.6 μs, each sample contains its own GPS signal for synchronisation of each device and subsequent metadata. Because it has its own GPS, its location is also identified and any problems from the norm can be identified and located (Shand *et al.*, 2015).

RQ: *To what extent can dimensional reduction be applied at the distributional level using the following real time Micro Phasor Measurement Units (P3001199, P3001065, P3001352, P3001095, P3001289) so that maximum variance can be maintained and noise reduced?*

The following research objectives shown in Table 1 will be examined and implemented in detail so that the research question can be answered.

1.3 Research Objectives and Contributions

The project is implemented on a distributed computing platform. The use of Cassandra and Python with Spark using a real-world dataset will be used to determine principal components that can be used for further study. The resultant PCA can then be presented visually to stakeholders. One of the tasks that needs to be addressed is in the area of visualisation of PMU information, this is due to the fact that the data generated is of such a large size that it is becoming a problem for stakeholders such as industrial operators to understand (Yang *et al.*, 2009).

Table 1 Research Objectives

Objectives	Description	Evaluation Metrics
Objective 1	A critical review of the present literature on PMUs and dimensional reduction techniques is to be carried out.	
Objective 2	Creation and Implementation of μPMU real-world dataset.	
Objective 2 (i)	Develop a real-time database that can handle large amounts of time-series variables providing high availability with no single point of fail.	Implement a timeseries range query
Objective 2 (ii)	One task is to identify a solution that addresses the low latency requirement due to the fact that 120 samples/sec from each μPMU are recorded.	
Objective 2 (iii)	Perform Data extraction and analysis on dataset before pre-processing.	Bartlett's test of sphericity, Kaiser-Meyer-Olkin (KMO), Kaiser's criterion, Dickey Fuller test, Correlation Analysis
Objective 3	Implement, evaluate and analyse the results of PCA.	Scree Plot, Kaiser's criterion, Explained Variance Plot
Objective 4	Implement, evaluate and analyse the results of IPCA using Singular Value Decomposition (SVD).	Scree Plot, Kaiser's criterion, Explained Variance Plot
Objective 5	Implement, evaluate and analyse the results of IPCA using K-means Clustering.	Silhouette Score, Elbow method, Cost Function, Cluster distribution
Objective 6	Implement, evaluate and analyse the results of IPCA using DBSCAN.	Number of Clusters, Noise points, Mahalanobis, Euclidean distance
Objective 7	Compare and Contrast the results from the transmission line and distribution line. Reconstruct Signal from PCs.	Comparison of Components

Contributions: The major contribution of this research will ensure that the future of the smart grid is made more efficient, reliable and lessen the financial burden to utilities and their stakeholders which consists of regulators, customers, environmental partners, local authorities and investors so that the utilities make full use of the massive datasets at their disposal. This goal will ensure that a disturbance in one section of the smart grid does not propagate to another section due to the electrically coupled nature of the transmission system. The minor contribution will ensure that the burden of storage costs is not placed on consumers.

The remainder of the report is structured as follows. Section 2 is a review of the current literature surrounding PMUs and dimensional reduction, Section 3 is the methodology and design process that was followed, this is followed by Section 4 the implementation, evaluation and results section followed in Section 5 by the discussion chapter and finally section 6 the conclusion and future work summary.

2 Literature Review on Dimensional Reduction for Phasor Measurement Units (2009-2018)

2.1 Introduction

In this section, the literature is reviewed around several key areas divided into many subsections with (2.2) literature review on PMUs using big data technologies (2.3) there is a discussion of Event Detection using PMUs and their use in analytical problems and (2.4) dimensional reduction techniques are examined such as PCA next, and finally (2.5) methods of presenting visualisations to end users are looked at.

2.2 Literature Review on PMUs using Big Data Technologies

In this subsection, some of the literature around whether PMUs are a part of the big data science are looked at and the various applications that make use of this data using big data technologies are examined.

In (Yang *et al.*, 2015) they note that the traditional power industry is not ready to handle the information explosion that is taking place with the introduction of PMUs and so they can't take advantage of the technological solutions that are available today such as machine learning, data mining and cloud based computation. As outlined by (Akhavan-Hejazi and Mohsenian-Rad, 2018) it has all the hall marks of big data namely "high-volume, high-velocity and high-variety", but at the same time most of the data is not stored for long term use. From its inception there was never any intention of transferring it to an enterprise data warehouse, which is the typical approach in the IT sector,[2] so there is no data logged and hence a paradigm shift is needed. One other problem that the authors have identified is that because the sampling rate of PMUs is so high and the time window so tight that the typical database systems such as SQL or HDFS are not capable of capturing this data.

[2]https://www.crcpress.com/Big-Data-Analytics-Strategies-for-the-Smart-Grid/Stimmel/p/book/9781482218282

(Khan *et al.*, 2014) proposes a parallel detrended fluctuation analysis (PDFA) approach for fast detection of transient events using a computer cluster based on the MapReduce model. The results outperform the detrended fluctuation analysis (DFA) in computation significantly using 8 VMs, while the execution time is almost constant as opposed to DFA where it increases with an increasing number of data samples. Looking at accuracy the results are similar. Because the MapReduce Hadoop framework has over 180 configuration parameters and the performance are affected by the choice of these parameters the authors are researching the Starfish system to self-tune so that the performance of Hadoop is optimized for big data analytics. Because MapReduce relies on periodic batch processing it is not suitable for real-time data streaming of enormous dynamic datasets which is a major drawback of Apache Hadoop. Correcting for this (Ganesh *et al.*, 2015) utilize the Apache Spark framework with its real-time in-memory approach and spark streaming component that calls dependent logic on each new data instance instead of waiting for the next batch of data to do the processing all at once. The advantage being that repetition of reprocessing of the data is avoided and more accurate and timely results are attained relative to batch processing.

In this section, it has been shown that PMU data although it is part of the big data environment, utility companies are not taking advantage of this data where at present it is siloed and not integrated. The current Hadoop MapReduce batch processing model is not sufficient to capture the real-time nature of the data and there are problems with the current SQL and HDFS database systems being unable to capture the data in the time window available. Apache Spark with its in-memory processing capability was put forward as an alternative to MapReduce.

In the next subsection, a number of detection-based models are examined which can be useful in the task of detecting known events.

2.3 A Critique of Methods and Techniques for Phasor Measurement Units and Identified Gaps

In the previous section it was shown that current approaches using MapReduce method to detect anomalies are not suitable for detecting known events in real time, the following section will advance on this topic by examining various models put forward to solve these problems.

Anomaly detection is a frequently required task. (Yang *et al.*, 2018) develop a novel PMU fog inspired by edge-fog-cloud[3] computing using two anomaly detection approaches Singular Spectrum Analysis (SSA) and k-Nearest Neighbours (k-NN) resulting in the data flow end-to-end delay being condensed without losing data completeness due to the real-time nature of PMU having strict latency requirements. The hard part is figuring out which data is of vital importance to the operator and which data can be ignored. This requires being able to access PMUs on-site so that marked data will be transmitted with a high priority which is not feasible for my situation. One interesting finding from the paper is that transient stability analysis requires the latency to be less than 100ms, this further advances the requirement that Apache Spark is the right choice with its new structured streaming continuous processing providing

[3] https://www.information-age.com/cloud-edge-fog-computing-123476326/

latencies as low as 1ms.[4] If low computing cost is the main factor then k-NN is the appropriate choice but the detection method has a delay about one cycle which is the window length. In (Vittal, 2012), decision trees (DTs) are used to develop a transient and voltage stability application by identifying critical attributes that can characterize phenomena associated with system dynamic performance using PMU data. By comparing real-time measurements to thresholds stored in the DT related paths and terminal nodes can be determined. This method can give more reliable assessment results against changes in operating conditions because for each path an insecurity score is calculated and a classification result based on this score is used to identify the security classification "secure" or "insecure". The classification results of the DTs can be easily demonstrated to stakeholders and quick decisions can be acted upon using this information by operators. While this approach is sufficient to address the problem locally it can't be scaled due to the fact that the number of operating conditions is infinite and many DTs would need to be trained for each eventuality which is not feasible. Looking at the problem of detecting abnormal events, (Zhou *et al.*, 2016) suggest applying ensemble bundle classifiers (EBC) to micro-PMU data where multiple classifiers are trained each with a generated μPMU positive event for the training sample and all stable data as a negative sample and then all the classifiers are pooled together in an ensemble method to create a new sample with the final decision being made by the most confident classifier. This is advantageous since present machine learning algorithms such as Support Vector Machines (SVM), Decision Trees (DT) and Logistic Regression build a single classifier by pooling events of interest into positive training samples and stable instances into negative instances but the problem arises when the positive sample is made up of many smaller subgroups each of which is allocated a weight for example in the case of SVM where the algorithm has to discriminate by finding a common weight for features of all subgroups. The resulting performance is illustrated in Table 2.

Table 2 Comparison of Detection Performance

Method	EBC	Ada. Boost	SVM	Logistic Regression	Author
Accuracy %	95.23	83.44	85.69	80.66	
False Positive Rate %	9.54	17.10	6.73	9.72	(Zhou *et al.*, 2016)
MDR %	0.0	16.02	21.89	28.96	

This choice of ensemble model has some worth but to detect the known events for the μPMU positive event, domain experts will have to have access to the data to detect these positive events which to non-subject matter experts may suffer from overdetermination where there are more causes present than necessary to cause the event of interest and it also can't be scaled as the results are germane to this location as discussed previously.

In this section it has been shown that there are various event detection models put forward each of which has some merit. The PMU fog method requires being able to access the PMU onsite, EBC has been shown to be far superior to that of other models but requires a domain level expert to identify the known event. Nevertheless, being able to access the known events

[4] http://spark.apache.org/docs/2.3.0/structured-streaming-programming-guide.html

from another said device it may be possible to train the classifiers locally. It has been shown that classification techniques do not scale due to the local operating conditions.

The next subsection will examine dimensional reduction techniques and how they can be applied to reduce the computational requirement of analysing μPMU data.

2.4 A critique of Dimensional Reduction Techniques and Identified Gaps

In the previous section certain models were looked at with regards to event detection, in this section dimensionality reduction is examined with the aim of reducing the noise in the dataset so that machine learning models are capable of detecting such events. Dimensionality reduction is important because it alleviates the curse of dimensionality.

In this subsection, dimensional reduction techniques are examined and their applications for real-time applications are looked at. (Wang and Yang, 2009) look at three data mining models namely frequent patterns, clusters and classifiers that have proven successful in analysing very large data sets. One of the challenges that needs to be resolved is the "curse of dimensionality" where the algorithm becomes computationally expensive and not suitable for many real-time applications and specificity of data points in high dimensional space i.e. the distance between points becomes indistinguishable from neighbouring points. One widely used algorithm is PCA (Wold, Esbensen and Geladi, 1987) which is an unsupervised frequent linear technique i.e. we assume the data lie on, or near a linear subspace in the high dimensional space used for dimensional reduction. In (Chen, Xie and Kumar, 2013), they use actual PMU data from Electric Reliability Council of Texas to propose an Early Event Detection algorithm using PCA. One of the main advantages of such an approach is that by reducing the dimensionality of the data, operators can improve their situational awareness of the electrical grid. One major advantage of the proposed algorithm is that it requires no previous knowledge of the system. For global variables such as bus frequency the reconstruction accuracy will be high but this is not the case for local variables such as voltage magnitude or reactive power so a new algorithm will need to be created. For online detection, the model could not detect the frequency deviation that occurred in the window frame of 0.0005pu because it was too small to capture which was also confirmed by (Yang *et al.*, 2018).

PCA is used to extract the dominant features of the system which are then fed as the input to a multi-class SVM in (Niazazari and Livani, 2018) by simulating two disruptive events and comparing against the normal load changing event. While the results are positive it suffers from the lack of a real-world dataset due to the data being simulated. Looking at the various options available today for dimensionality reduction, (Box *et al.*, 2009) investigate nonlinear techniques on artificial and natural tasks revealing that while nonlinear techniques outperform PCA on artificial tasks when it comes to real world tasks PCA outperforms all other techniques. This is due to the fact that local learners suffer from the curse of dimensionality and hence learning techniques that use Gaussian kernels such as SVM and Gaussian processes do to. The downside with PCA is that the size of the covariance matrix is proportional to the dimensionality of the datapoints i.e. the direction of the linear relationship between variables is proportional to the dimensionality of the datapoints.

This subsection has examined dimensionality reduction techniques and found that PCA is still the number one choice for real-time data. In the final subsection, presentation of high dimensional data is looked at.

2.5 An Investigation of Visualisation Techniques for High Dimensional Datasets and Identified Gaps

Dimensionality reduction was examined in the previous section, this subsection addresses the issue of presenting high-dimensional PMU data to stakeholders so that effective decisions can be made.

Viewing multivariate data is notoriously difficult for humans to interpret beyond three dimensions, resulting in complex visualisation plots with less interpretability. Visualising dimensionality reduction techniques have their use in images but they are not applicable in time-series data where each data source could have different properties. One research gap identified is that there are not many research papers published on the presentation of power system data especially visualising high-dimensional multivariate time series. The following paper by (Bhuiyan, Khan and Murphy, 2017) presents a real representation of the grid using the R programming environment using a Naïve Bayes classification method based on MapReduce. One drawback on using R for PMU data is the need to ensure there is enough memory so that the 'R' session does not freeze. By using this approach, the PMU data can be reduced to a few thousand samples that will be used as the focus for further analysis. The results can detect all disturbances either in the long term or the very short term by plotting a density estimation on the frequency variable of all the samples in a "file". The frequency variable being chosen because disturbances manifest themselves in the frequency signal. Looking at this problem of visualising high-dimensional data, (Nguyen *et al.*, 2017), propose a new framework "m-TSNE", by projecting them onto a low-dimensional space while preserving the underlying data properties, which looks for similarity in high dimensional space between the data points and then preserves this similarity by applying gradient descent to produce a low dimensional structure. The advantage being that it is easy to use and understandable so that interpretable insight can be gained by its use. (Chambers *et al.*, 1983) proposed using a star diagram for high dimensional data, but this was superseded by the work of (Kandogan and Kandogan, 2000) who created a new technique which builds layouts using dimensional reduction called Star-Coordinates (SC) whose aim is "not numerical analysis but to gain insight", but with the added drawback that users have to be willing to tolerate loss of information but gain insight in the process so that they know where to look when performing further numerical analysis. This work was further built on by (Garcia Zanabria, Nonato and Gomez-Nieto, 2016) who created a new technique called iStar (interactive Star Coordinates) which mitigates against the deficiency of SC which is that as the number of dimensions increase in the present era of big data the visualisations become cluttered. So, they expand on the SC approach which clusters datapoints around the dimensions by allowing user interaction through automatic and interactive data exploration so that most patterns can be revealed. This has the added benefits that it reduces visual clutter while preserving the essence of the information to be conveyed in the layout. From an aesthetic point of view the iStar method is appealing.

In this subsection a few methods were looked at that could be useful to an end user such as a network operator to reduce clutter of high dimensional data while retaining insight.

2.6 Conclusion

In this section various research papers were examined with regards to the introduction of PMUs and how their potential is not fully realised. Real-time event detection with dimensionality reduction to reduce noise for the application of machine learning models was examined and finally an aesthetic visualising method was put forward that may be used to build data insights was explored. The next chapter presents the scientific methodology that was followed and the architecture design chosen. This fulfilled Chapter 1, Table 1, Research Objective 1.

3 Micro-Phasor Methodology Approach and Design Used

3.1 Introduction

In this section the Knowledge Discovery Database (KDD) research methodology approach as proposed by (M. Fayyad, 1994) is followed. The motivation for the research is to bring insight to business so that effective decisions can be made by end users and stakeholders. A three-tier architecture approach is presented showing the flow of steps that were followed and technologies used to implement this project.

3.2 Micro Phasor Methodology Approach Used

The research methodology steps are shown in Figure 2. It consists of the following stages:

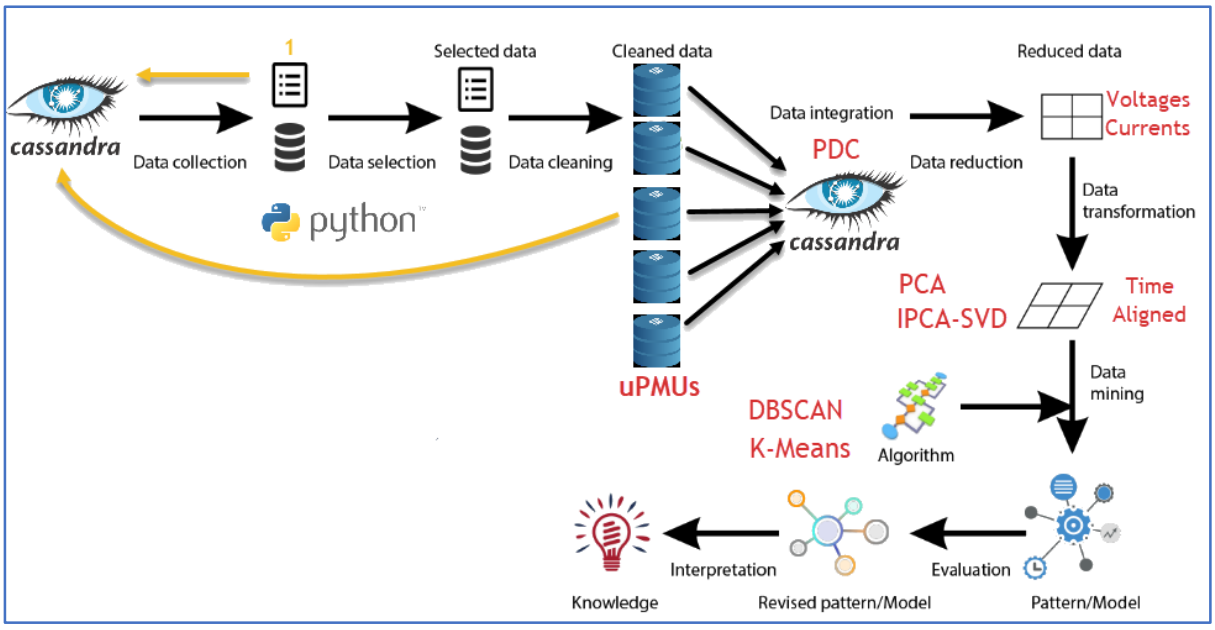


Figure 2: Micro Phasor Methodology Approach Used

3.2.1 Data Selection

The dataset was made available to qualified researchers through a Python API obtained from the Lawrence Berkeley National Laboratory (Peisert *et al.*, 2018). The µPMUs datasets made available are similar to PMUs at the transmission level, but provide a higher sampling rate and precision at the distribution level. Each µPMU device produces 12 GPS time-stamped

measurement to provide time-synchronized observability (Zhou *et al.*, 2016) of the three-phase voltage and current magnitudes and angles at 120 Hz i.e. 120 samples a sec stored in a Cassandra NoSQL database. The streamed data is cleaned automatically to facilitate researchers rapid deployment of human-centric analysis (Andersen *et al.*, 2015).

This data was stored in a Cassandra database. The Cassandra database was queried using the device ids for each of the μPMUs with a start and end time in epoch time with the provision that only one thread could be used at a time. Each call to the Berkeley Cassandra database returned 13 features at 120 samples a second from each of the μPMU devices and these were then stored in SSTables in a Cassandra database where they could be queried.

3.2.2 Data Pre-processing

Each stream of data from the Python API produces the following features: three-phase voltage and current magnitude and angle measurements from five locations as shown in Table 3 for one μPMU device.

Table 3 μPMU data from LBNL campus (Von Meier et al., 2017)

	Variables/Features	Feature Description	Measurement Units	Datatype
1	Timestamp	GPS synchronised timestamp	100ns	Timestamp
2	C1mag	C1 Current Magnitude	10^{-4} *per unit*	Float
3	C1angle	C1 Current Phase	0.01^{o}	Float
4	C2mag	C2 Current Magnitude	10^{-4} *per unit*	Float
5	C2angle	C2 Current Phase	0.01^{o}	Float
6	C3mag	C3 Current Magnitude	10^{-4} *per unit*	Float
7	C3angle	C3 Current Phase	0.01^{o}	Float
8	L1mag	L1 Voltage Magnitude	10^{-4} *per unit*	Float
9	L1angle	L1 Voltage Phase	0.01^{o}	Float
10	L2mag	L2 Voltage Magnitude	10^{-4} *per unit*	Float
11	L2angle	L2 Voltage Phase	0.01^{o}	Float
12	L3mag	L3 Voltage Magnitude	10^{-4} *per unit*	Float
13	L3angle	L3 Voltage Phase	0.01^{o}	Float

Each dataset was grouped in Python into separate dataframes according to device id before being aggregated into a timestamped column in Cassandra, from here column selection on each feature was carried out and an exploratory analysis on one of the devices was carried out in PySpark because it was more efficient. The process to format the data as described is found in the configuration manual.

3.2.3 Data Transformation

In this step the Cassandra database was queried and the results from the Cassandra query were stored in a pyspark dataframe from each of the μPMU devices. The data is transformed and integrated into a format for dimensional reduction where the timestamps were aligned from each of the devices and subsequent columns were filtered.

3.2.4 Data Mining

The following algorithms were run on the data, Principal Component Analysis (PCA), Incremental Component Analysis using SVD, K-Means a clustering algorithm was applied to the principal components, followed by Density-Based Spatial Clustering of Application with Noise (DBSCAN) to detect noise for dimensional reduction to be effective.

3.2.5 Data Interpretation and Evaluation

Augmented Dickey Fuller tests were carried out prior to the application of PCA to ensure stationarity, Bartlett's test of sphericity was carried out to check that the observed variables intercorrelate. Kaiser-Meyer-Olkin (KMO) test was carried out to ensure whether the data was suitable for factor analysis if applied to the dataset. Kaiser's criterion was evaluated to see how many components to extract, followed finally by a Scree Plot and Explained Variance Plot were examined for the number of Principal Components to extract. A hexbin plot was plotted due to the large number of datapoints to gain insight followed by Silhouette score for K-Means and DBSCAN and cluster distribution. This was followed by an examination of the number of noise points and a comparison of the two clustering algorithms.

3.2.6 Architectural Design Process Flow

The following three tier architecture diagram with the different technologies that were used to implement this project are shown in Figure 3. It consists of the Data Persistence layer, Business Logic Layer and finally the Presentation layer.

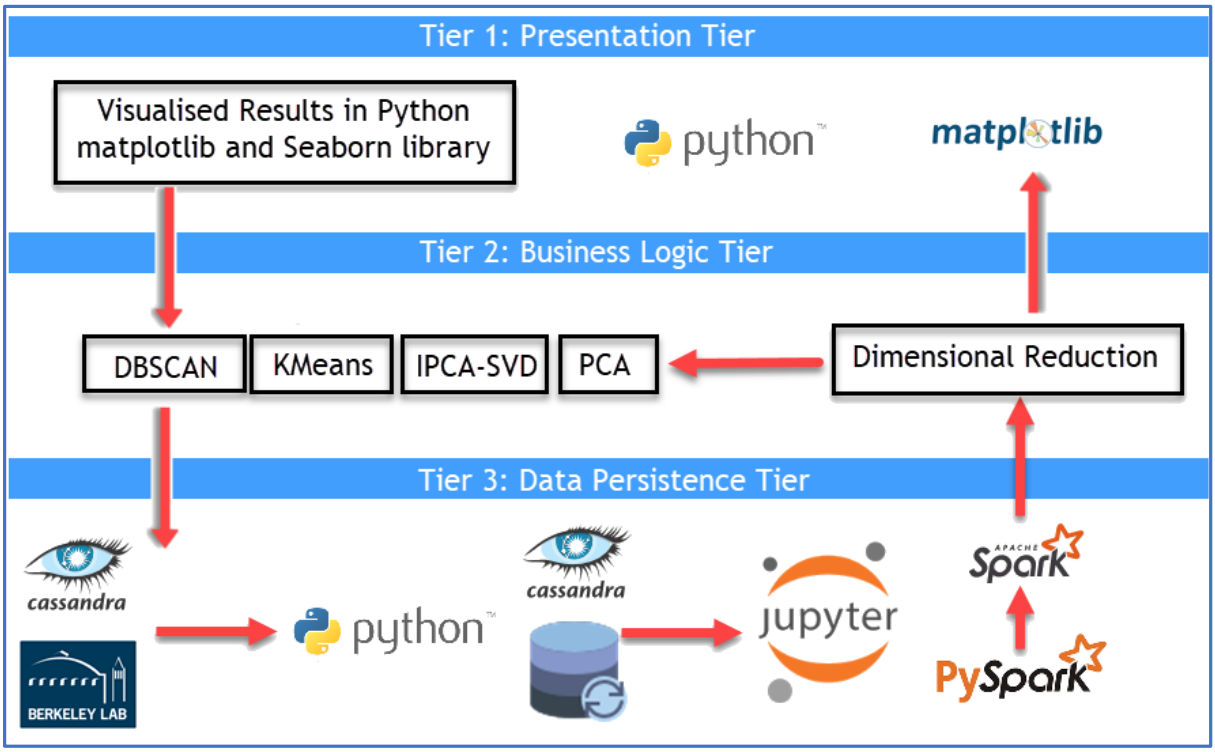


Figure 3 Micro Phasor Architecture Design

3.3 Conclusion

The KDD methodology was chosen as the best methodology for this project and amended for my requirements as sought. The dataset was obtained from the Lawrence Berkeley National Laboratory Cassandra database, and dimensional reduction using PCA/IPCA/SVD was applied as this was the best performing reduction techniques identified previously in the literature using real-time data by (Box *et al.*, 2009). Now principal components could be identified as a pre-processing step before the following clustering data mining models such as K-Means and DBSCAN could be implemented.

In the following chapter the implementation, evaluation and subsequent results to identify known components that will be fed into the data mining models will be implemented.

4 Implementation, Evaluation and Results of Real Time Reduction Models

4.1 Introduction

This chapter discusses the design decisions taken and how the project was implemented. The choice of technologies that were chosen and the various feature extraction and data explorations that were carried out with subsequent visualisations.

4.2 Creation of Dataset from Berkeley National Laboratory

In this subsection I am solving Objective 2 (i). The first step to answer the research question was to obtain a real-time dataset from the distribution network. The Berkeley National Laboratory[5] provided me with access to the dataset on condition that it was for research purposes only and that I could only use one thread at a time. The code provided was Python code that connected to their Cassandra database and providing the device id and the start and end time in epoch time I was able to obtain the μPMU data for one device at a time. The rest of the device id's I was able to obtain from their website. Each device had an ID, so I created a separate id feature to be used later to identify the device when upserting into Cassandra.

4.2.1 Cassandra Database Creation and Design

I created a Cassandra database using CQL to store the time-stamped datasets. I chose Cassandra because it provides a wide-column store that emphasizes scalability and high availability considering the nature of its use case. Cassandra provides no single point of failure and so one important advantage being that it helps you avoid outages. You can replace any failed nodes in the cluster without any downtime. Cassandra's data model is a partitioned row-store which stores data in "sparse" multidimensional hashtables. As was seen earlier in Table 3 each device produces 13 features and the device id producing 14 features. Each partition can support up to 2 billion *cells*, which is just a key/value pair which will hold our 13 features, timestamp being the 13th and target/device added bringing the total to 14. With 120 samples a sec producing 14 features for 1 minute would produce 6,048,000 million cells for just one

[5] https://www.lbl.gov/about/

device in one hour, multiply that by 5 produces 30,240,000 cells in just 1 hour and 735,726,000 in one day. Since Cassandra doesn't support joins, I had to create a table that fulfils all the following queries since you build your Cassandra tables around the queries since it is a query-based data model.

1. Select the latest timestamp and device id and features.
2. Select all known device ids and features.
3. Select all devices and features within a time range.

One problem that had to be overcome was the problem of unbounded row growth. Each row was partitioned with a unique key called the target id. The partitioned key means that its data can be accessible, and the keys are used to distribute the data across multiple data stores. When the design of the Cassandra database was completed, I created a Cassandra keyspace called upmu, which is like a relational database. I chose SimpleStrategy with a replication factor of 1 because I was working on a single local node. Next I created a Cassandra table called micro_pmu_data with the features that contain the values having a double datatype. I created a timestamp column in Cassandra for the timestamp coming from Python. Unix timestamps are in seconds since epoch but the timestamps in Cassandra are stored in milliseconds since epoch so I multiplied the timestamps by 1000 before upserting to Cassandra. Cassandra understands how to store your data i.e. in which token range in the cluster to use with the use of partition keys. When the data is written to a data partition it is sorted by your primary key. So, I created a separate variable called daybucket which I then used so that there would not be unbounded row growth as discussed. So, my final table contained a Primary Key which consisted of a clustering key which consisted of ((target, daybucket), timestamp).

Once, the design of the Cassandra database was completed I went back to the Python code and iterated over the list calling a function called getDataCassandra() with the device id and start and end time in epoch. Once the data was retrieved I created a separate function called processdataCassandra() to process the data, which consisted of connecting to my Cassandra cluster on my local machine, using a session to connect to the keyspace I then created a prepared statement to upload the datasets into my Cassandra table. For my analysis I retrieved ~5,000,000 rows of data, ~500MB, ~2hr 30 mins but this is not restricted. You can retrieve as much as your storage can hold on your hard drive or unbounded in the cloud for future research. This fulfilled my Chapter 1, Table 1, Research Objective 2 (i).

4.2.2 Cassandra Spark Connector

One of the Research objectives was to have low latency requirement to process the dataset. I chose Spark in conjunction with Cassandra due to Sparks in-memory processing and Cassandras ability to handle real-time timestamped data. Once chosen, I then had to integrate them. So, to process this large number of records I installed Spark on Windows, and installed a Cassandra-Spark connector from DataStax. I then created a SparkSession that connected to my localhost on port 9042. With the SparkSession I was then able to create a SQLContext which I then used to create a DataFrame that connected to my Cassandra keyspace and table using the SQLContext.read.format which accepts a connector class and once the load() function is called with the keyspace and table the data was read into a Spark DataFrame which is a

special type of RDD using lazy functions, thereby fulfilling my Chapter 1, Table 1, Research Objective 2 (ii).

4.3 Data Exploration

Once the datasets were inserted into a Spark DataFrame, the following exploratory analysis was carried out. I took a look at the first few rows to get a feel for the type of data I was dealing with using Pyspark. I split the Spark DataFrame up into separate dataframes by filtering on their target ids, to create a separate dataframe for each device so that I could perform exploratory analysis on one of the devices. I examined the number of records and examined each of the features. I dropped the daybucket and target column from this dataframe since they were of no use for this analysis. I examined 2400 rows which was only 20 seconds worth of data i.e. 120 samples a second. I imported pandas and numpy Python libraries and scikit-learn, and created a timestamp index to perform an exploratory analysis. The dataset had no missing values and was already cleaned and so no further pre-processing was needed. Next a correlation analysis was run showing how highly correlated the variables of interest are, see Figure 4.

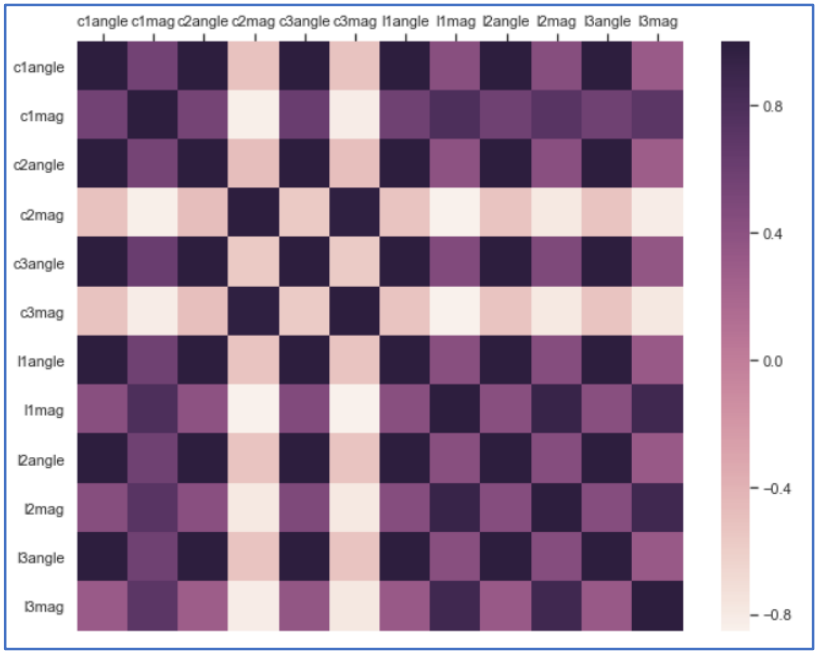


Figure 4 Correlation Plot for device P3001065

4.3.1 Evaluation and Results of Tests necessary for PCA

Before applying PCA to the datasets the Augmented Dickey Fuller (ADF) test was ran on device id P3001289, of which the voltage magnitude (l1mag) is shown in Figure 5 to check for stationarity, which is a statistical property where the mean, variance and the correlation coefficients do not change over time i.e. are constant (Kiyoung Yang *et al.*, 2005). The test was run over 56000 samples / 7 secs.

Note: For further exploratory analysis done see the configuration manual.

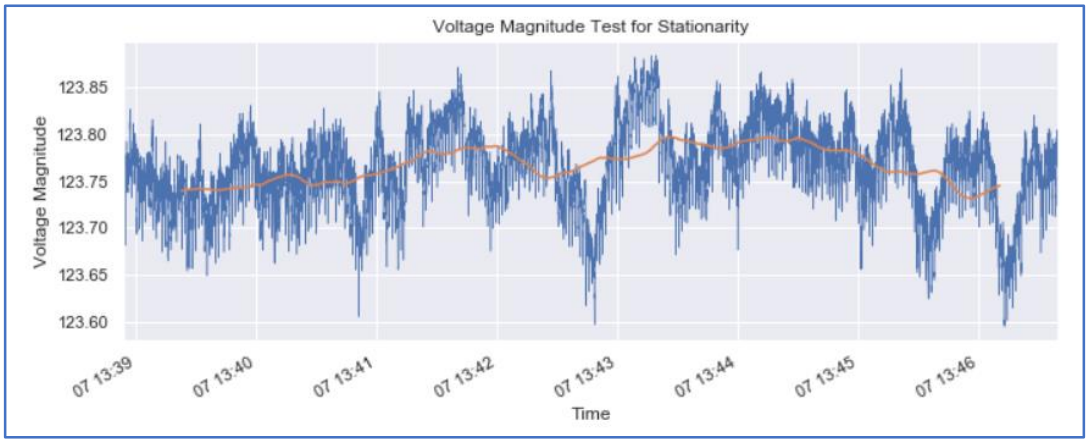


Figure 5 Dickey Fuller Test for Stationarity

The results in Table 4 show that as the Test Statistic is less than the Critical Value for all Multivariate Time Series the variables are stationary which would imply that a correlation-based representation of the original MTS items is more stable and PCA can be applied.

Table 4: Dickey-Fuller Test Results

Variables	Test Statistic	Critical Value (1%)
c1angle	-2.324085	-3.430467
c2mag	-37.582425	-3.430467
c2angle	-5.007482	-3.430467
c2mag	-37.574909	-3.430467
c3angle	-5.056427	-3.430467
c3mag	-1.063858e+01	-3.430467
l1angle	-5.419823	-3.430467
l1mag	-1.143068e+01	-3.430467
l2angle	-4.995829	-3.430467
l2mag	-1.081254e+01	-3.430467
l3angle	-5.063520	-3.430467
l3mag	-1.081254e+01	-3.430467

Bartlett's test of sphericity resulted in a p-value of 0, which was statistically significant indicating that the observed correlation matrix is not an identity matrix and Kaiser-Meyer-Olkin (KMO) test result was 0.873 which is useful if factor analysis was applied.

4.4 Implementation, Evaluation and Results of Principal Component Analysis Model for Dimensional Reduction

PCA was chosen as per the literature as outlined by (Box *et al.*, 2009) when it comes to real-world datasets PCA outperforms all other techniques. PCA is an unsupervised dimensionality reduction technique. In this process new components are created that will transform the dataset into a lower dimensional space while retaining most of the energy in the components. PCA can be used to reduce the curse of dimensionality and thereby improving the predictive performance by reducing the noise content in the electrical signals as discussed in the literature.

For the following implementations each of the currents and voltages were grouped separately into 15 dimensions each and PCA applied to maximise variance. The first principal component has the largest possible variance, and all subsequent principal components have the largest variance giving that they are orthogonal to the other principal components. Correlation analysis was already reviewed early in this chapter showing the strong correlation among the features.

4.4.1 Implementation

PCA was implemented twice first using scikit-learn library in Python and then in Spark using SPARKML where a dataframe was created with a Vector called features which contained the dimensions row wise of our features. The data was then rescaled to have zero mean and unit variance using fit and transform, the results of which can be seen in the evaluation section.

After the Cassandra datasets were read into a Spark DataFrame, a temporary view was created using createOrReplaceTempView called upmu_data and from this table separate dataframes were created with the query results of each of the voltages and currents renamed alongside their timestamps. Then all the dataframes were joined together using an inner join on their timestamps. All the timestamps bar one was dropped and the remaining one was cast as a timestamp index. The code for these steps was produced in a Jupyter notebook using PySpark and Python and can be found in the configuration manual. For the first implementation the following steps were implemented using Python, the *d*-dataset was standardized using StandardScaler function from sklearn.preprocessing and StandardScaler fit_transform function, a covariance matrix was constructed using numpy on the transposed dataset which indicates the level to which two variables vary together. The eigenvectors and eigenvalues were found by decomposing the covariance matrix using np.linalg.eig from the numpy library. The eigenvalues were sorted in descending order to rank the corresponding eigenvectors. Then *k* eigenvectors were selected which corresponded to the *k* largest eigenvalues. A projection matrix W was constructed from the "top" *k* eigenvectors. And the original *d*-dimensional dataset was transformed using the projection matrix to obtain the new *k*-dimensional feature subspace.

4.4.2 Evaluation and Results

To evaluate PCA, the proportion of variance explained by each component was calculated using the following formula, in Equation 2.

$$Proportion\ of\ Variance\ Explained = \frac{PCA\ Variance}{Total\ PCA\ Variance}$$

Equation 2 Proportion of Variance Explained

The results of PCA applied to the current magnitudes is shown in Figure 6 (a) it indicates that the first principal component (PC) alone accounts for 98.9% of the variance with the second principal component only accounting for less than 1%.

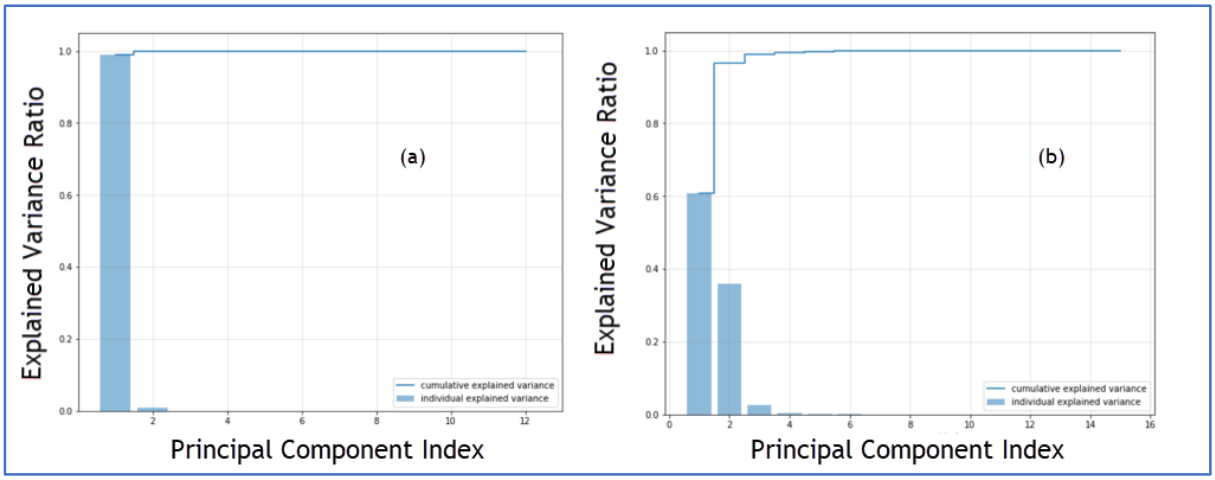


Figure 6 Current and Voltage Eigenvalues

The results for the voltage magnitudes are shown in Figure 6 (b) where the first 3 PCs account for over 99% of the variance. The results indicate that we can reduce the dimensions for the voltages from 15 dimensions to 2 components and retain over 96.5% of the energy. For the currents retaining 1 component from 15 is enough to capture over 98.9% of the variance. This fulfils my Chapter 1, Table 1, Objective 3.

4.5 Implementation, Evaluation and Results of Incremental Principal Component Analysis using Singular-Value Decomposition

For the second implementation, IPCA with SVD was used for dimensional reduction using SVD because it is best suited to large datasets which do not fit in main memory. Whereas PCA is used for batch processing, IPCA creates a low-rank approximation for the input data independent of the number of samples. This algorithm computes the principal components incrementally without estimating the covariance matrix.

4.5.1 Implementation

It was implemented in pandas using sklearn decomposition library using the following IncrementalPCA function[6] which stores estimates of component and noise variances and so memory usage depends on the number of samples per batch. The voltage dataframe was converted to a numpy array and was iterated over in chunks of 1000, and each chunk was fetched sequentially using the partial_fit method and then transformed.

4.5.2 Evaluation and Results

The results for the SVD IPCA approach are shown for the voltages which displays a similar explained variance to that of the PCA one previously but the implementation time was far quicker. Examining the plot in Figure 7 only components 1 and 2 would be extracted because their eigenvalues are above 1, with the proportion of variance explained by these two components accounting for over 96.5% of the variance.

[6] https://scikit-learn.org/stable/auto_examples/decomposition/plot_incremental_pca.html

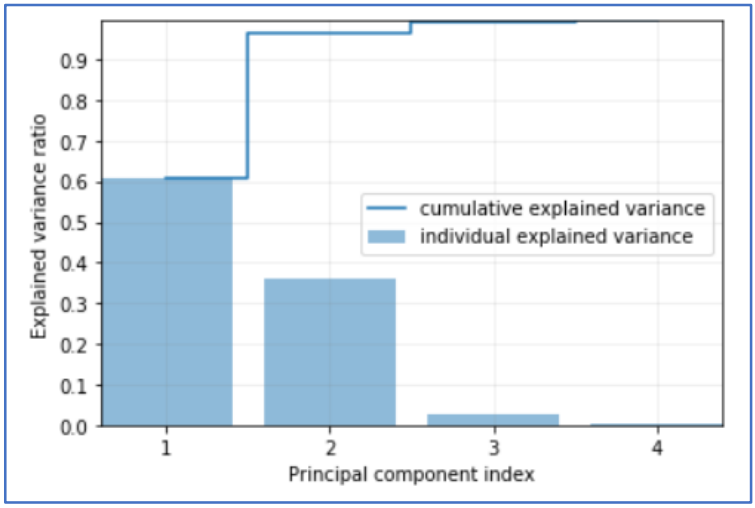


Figure 7 SVD Explained Variance

Because there were too many points to plot on the scatterplot for any meaningful insight in Figure 8 (a), a hexbin plot was used in Figure 8 (b) to bin the two variables using a log scale. The result show that within the data there are 4 visible clusters. From this insight, KMeans was used to cluster the principal components next. This fulfils Chapter 1, Table 1, Objective 4.

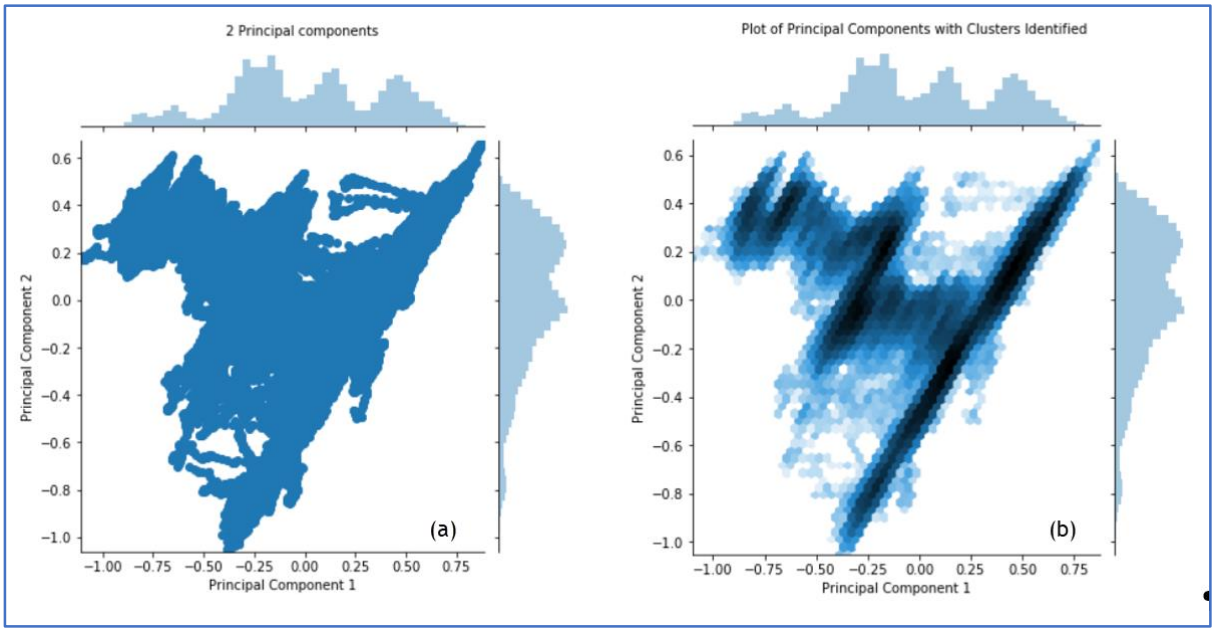


Figure 8 Plot of 2 Principal Components

4.6 Implementation of K-Means using SparkML

K-Means is an unsupervised clustering model which clusters the datapoints into *k* distinct clusters by randomly assigning each of the observations to one of the *k* clusters. The downside is that the choice of *k* has to be chosen a priori, once *k* is chosen the algorithm will assign observations to that cluster and only that cluster, therefore an observation can't belong to two clusters. Good clustering is evident when the variation within the cluster is as small as possible. The cluster centroid is calculated and the observations are assigned to the nearest cluster whose

centroid is closest based on Euclidean distance. This process will iterate until there is convergence and no more assignments (Ding, 2004). The within-cluster variation measures how different the observations are to each other within a cluster using the following formula in Equation 3:

$$W(C_{\mathrm{k}}) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2$$

Equation 3 Within-Cluster variation for *k* cluster

Where $|C_k|$ denotes the number of observations in a cluster *k*, *p* is the number of observations. It calculates the Euclidean distances of all the observations within a cluster and then divides this by the total number of observations in the k^{th} cluster.

4.6.1 Implementation

The results of IPCA were then applied to the K-Means algorithm by first creating a Spark dataframe from a Pandas dataframe. To use the SparkML machine learning library a features vector was created by passing in the spark dataframe to the VectorAssembler function in the pyspark.ml.features library which creates a single column called features which contains all the observations. From pyspark.ml.clustering library KMeans and KMeansModel were imported. For evaluating the choice of *k*, I trained a kmeans model by iterating over a choice of *k* and computed the Within Set Sum of Squared Errors cost function using the model.computeCost function on the Spark dataframe. It optimizes *k* by clustering a fraction of the data for different choices of *k* and then you look at the graph for an elbow in the cost function. The results are showing in the evaluation section. Once the number of clusters was calculated I then calculated the Silhouette score for the cluster (Thinsungnoen *et al.*, 2015).

4.6.2 Evaluation and Results

The silhouette score[7] ranges from -1 to +1, with a high value indicating that the observation is well matched to its cluster, the silhouette score is calculated using Equation 4 where:

$$s(i) = \frac{b_i - a_i}{Max(a_i, b_i)}$$

Equation 4 Silhouette Coefficient

i, is a single datapoint, a_i is the average difference between this point and all the other points within the same cluster. b_i is the lowest average difference between point *i* to all the points in the other clusters. Subtract one from the other and divide by the maximum number between a_i and b_i. The Silhouette score was: 0.736, indicating that the clustering configuration is adequate and the clusters are well compacted. The choice of *k* is shown in Figure 9 indicating that there is little gain after 4 clusters.

[7] https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient

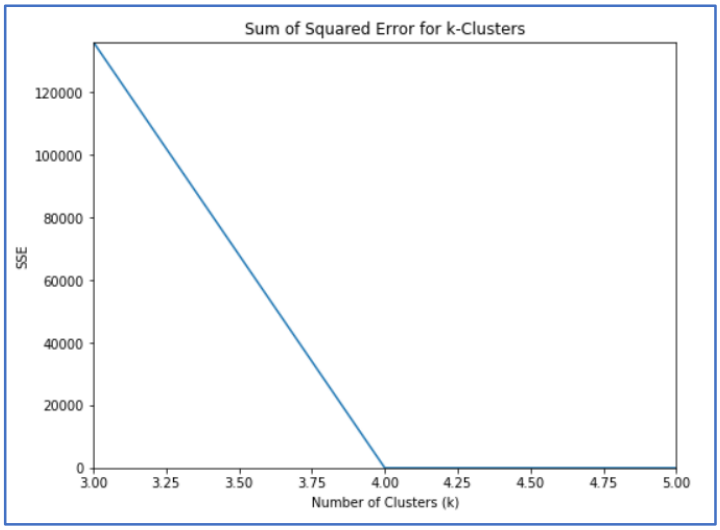
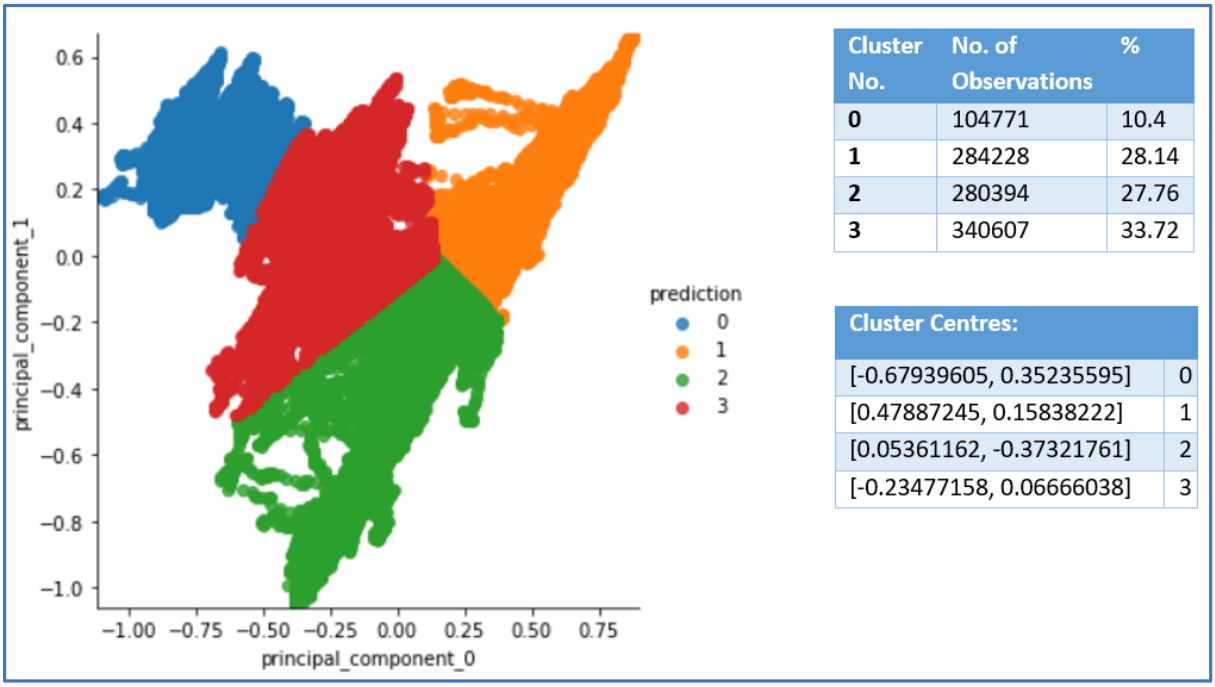


Figure 9 WSSSE Cost Function v *k*

The results from the K-Means model are shown in Figure 10 with the number of observations belonging to each cluster showing a fairly uniform distribution with Cluster 0 being different. With the clusters now assigned colours, the result from K-Means doesn't produce nice clusters because it assumes spherical clusters. From the hexbin plot in Figure 8 you can see that our clusters are very dense, the next algorithm implemented will try to identify these clusters. This fulfils Chapter 1, Table 1, Objective 5.



Cluster No.	No. of Observations	%
0	104771	10.4
1	284228	28.14
2	280394	27.76
3	340607	33.72

Cluster Centres:	
[-0.67939605, 0.35235595]	0
[0.47887245, 0.15838222]	1
[0.05361162, -0.37321761]	2
[-0.23477158, 0.06666038]	3

Figure 10 Plot of the K-Means Clusters Identified

4.7 Implementation of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) using Sklearn

DBSCAN[8] calculates dense regions of points by assigning cluster labels to these points. The density of each region is determined by the number of points within a specified radius ε. A point can be a core point if there lies a *min_samples* within its radius ε. It can be a border point if it has fewer neighbours than the *min_samples* within ε, but is itself contained within the radius of a core point. All other points are considered noise points, being neither core nor border points. If two core points are less than ε away from each other they are joined to the same cluster, otherwise a new cluster is created and then you assign each border points to the cluster that's within the radius of its core point. One advantage DBSCAN has is that it can handle large datasets but the choice of ε needs to be chosen carefully using domain knowledge.

4.7.1 Implementation

DBSCAN was implemented in sklearn, from sklearn.cluster I imported DBSCAN passing it the values for *eps* (ε), and the *min_samples*. But first I had to choose the values for eps and min_samples, these two parameters are essential to get right, otherwise most of your points will be noise and you will have memory issues. I chose ε as 0.01 and *min_samples* as 3 after reading the following papers by (Lee *et al.*, 2019) and (Wang *et al.*, 2019). (Lee *et al.*, 2019) uses DBSCAN on PMU data and chose ε as 0.07 for voltage, as seen in Figure 11 (a), but this wouldn't work for my situation using μPMUs so I chose 0.01 as indicated in the nearest neighbour graph in Figure 11 (b), which was created by calculating the distance from each point to its closest neighbour using the μPMUs and the results are displayed next.

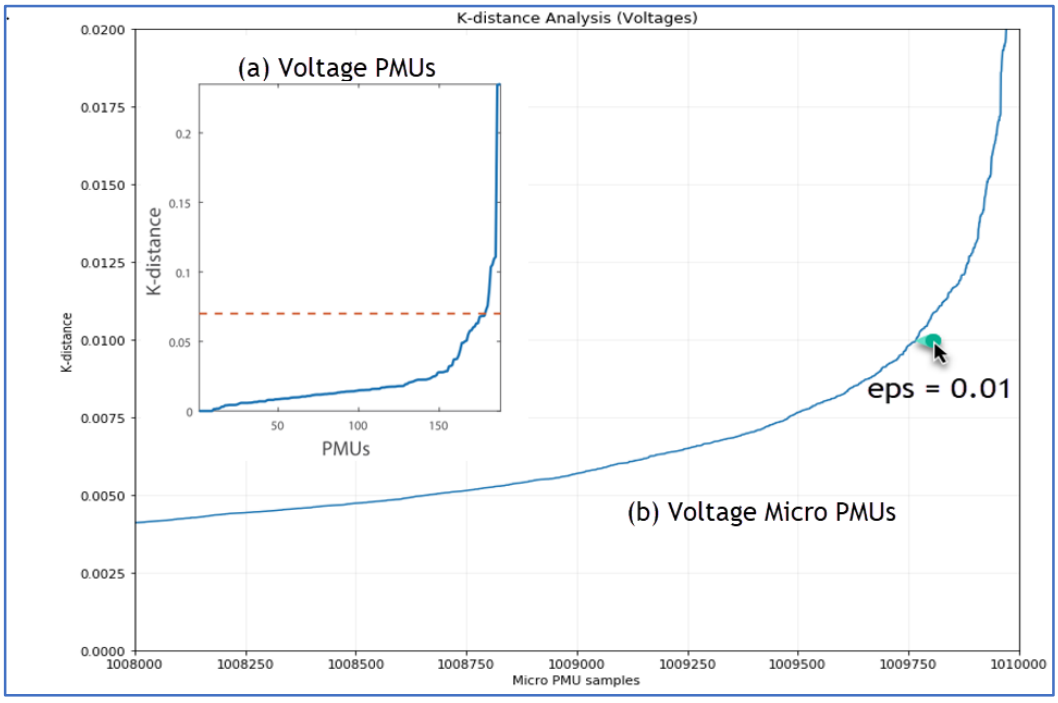


Figure 11 K-distance analysis. (a) Voltage magnitude PMUs, (b) Voltage magnitude μPMUs

[8] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

4.7.2 Evaluation and Results

The result of the Silhouette Coefficient took over 5 hours to run and produced the following figure -0.736, but on further study it appears it's the same figure as K-Means but has a negative sign in front of it, after researching this further it appears that Silhouette doesn't understand the concept of noise.[9]

The results estimated that the number of clusters was: 115, and that the number of noise points being 327. Out of 1010000 points only 327 didn't belong to any cluster. These points being the noise and can be removed or investigated further to ensure an accurate representation of the µPMU signal. These points are shown in black in Figure 12 in contrast to the other clusters identified in colour. What's apparent from looking at the main cluster is that the noise points lie along the edge of the main cluster and not every point needs to be part of a cluster as is the case with K-Means.

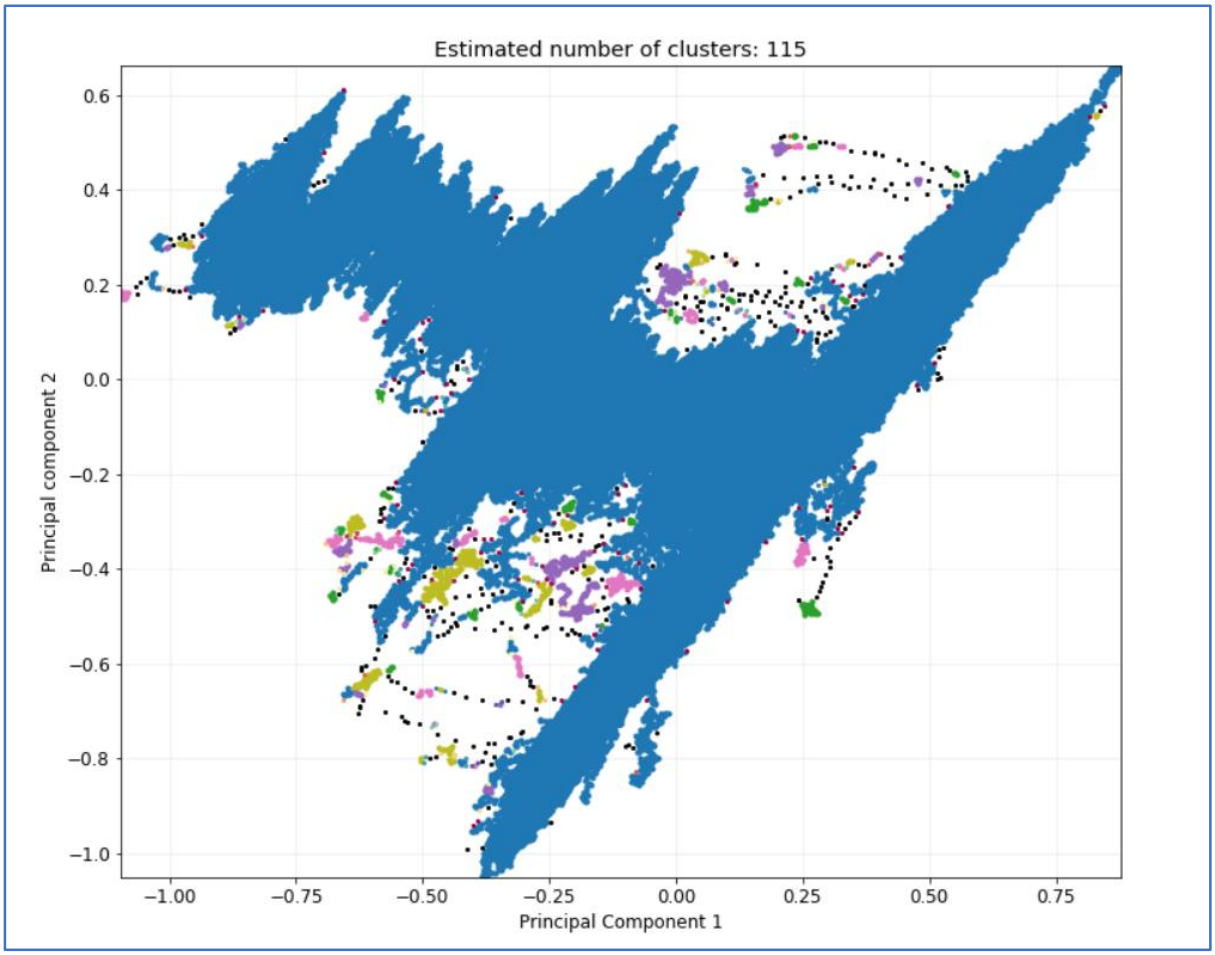


Figure 12 Estimated Number of Clusters with Noise

Next, the distance metric used by DBSCAN was changed to Mahalanobis distance which resulted in the number of clusters increasing by a multiple of 7 and the number of noise points by a multiple of 14 as shown in Table 5. The plot of which is available in the configuration manual.

[9] https://stats.stackexchange.com/questions/406587/determining-epsilon-for-dbscan

Table 5 DBSCAN distance metrics results

Metric	Estimated No. Of Clusters	Estimated No. Of Noise Points
Euclidean Distance	115	327
Mahalanobis Distance	848	4591

This fulfils Chapter 1, Table 1, Objective 6.

4.8 Comparison of Developed Dimensional Reduction Models with Existing Models

Table 6 provides results from existing research from the transmission side of the network where real and synthetic datasets were used, in comparison with my results in the last row for the distribution side of the network, where PC stands for Principal Component.

Table 6 Dimensional Reduction Transmission Line

Line	Dataset	Voltage Magnitudes	Current Magnitudes	Dimensional Reduction	Author
Transmission	N60, SEL421 PMUs	95% -98%	95%-98%	8 Voltages, 1 PC 8 Currents, 2 PC	(Dahal, King and Madani, 2012)
Transmission	PSS/E Data	97%-98%	80%	Voltages, 6 PC	(Xie, Chen and Kumar, 2014)
Transmission	BPA Data PJM	95%-96% 96%-97%		Voltages, 10 PC Voltages, 5 PC	(Chen, Xie and Kumar, 2015)
Distribution	µPMUs	96.5%	98.9%	15 Voltages, 2 PC 15 Currents, 1 PC	

4.8.1 Reconstruction of Original Signal from Principal Components

To reconstruct the original signal from the principal components each principal component must be placed on the same vector that was used for projection, as shown in Figure 13.

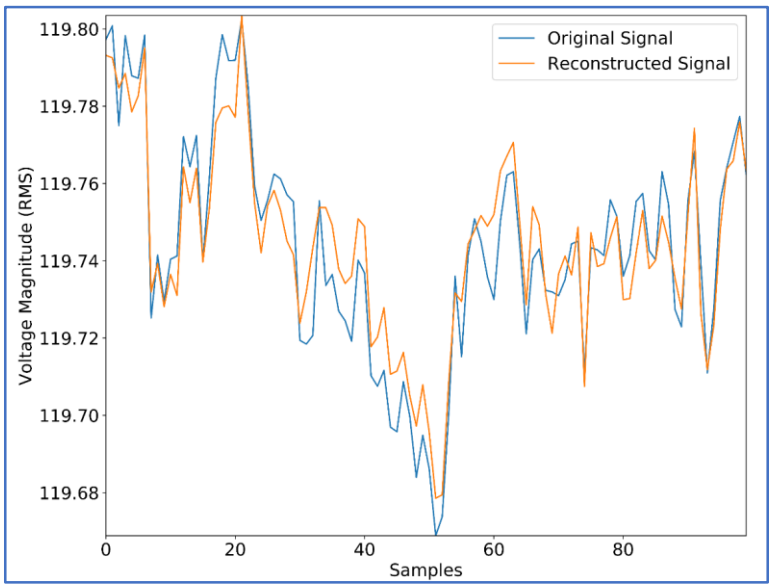


Figure 13 Reconstruction of original signal from principal components

The results indicate that dimensional reduction applied to the distribution line is effective, though it is not a lossless method, it may be used as a pre-processing method in data analysis and data storage where absolute accuracy is not required although looking at the following results in Table 7, the results indicate that it is highly correlated with minimum error.

Table 7 Correlation Coefficient of reconstructed signal Transmission v Distribution Line

Voltages	Distribution Line	Transmission Line
0	0.998082	0.9998
1	0.995423	0.9986
2	0.991999	0.9998
3	0.985804	0.9999
4	0.988464	0.9993
5	0.983415	0.9999
6	0.776728	0.9979
7	0.114446	0.9972
8	0.015131	
9	0.990419	
10	0.989815	
11	0.992117	
12	0.990523	
13	0.990177	
14	0.992117	

With a Root Mean Squared Error of just 0.02367

4.9 Conclusion

The results presented in this chapter and the evaluation results enable us to answer the research question. Additionally, the research objectives set out in subsection 1.3 have been achieved. IPCA with SVD in conjunction with DBSCAN will enable us to reduce the dimensionality of the data so the "curse of dimensionality" is alleviated and also able to detect and remove noise in the dataset but a careful choice of ε is warranted.

5 Discussion

In this research report, an online dimensional reduction technique using IPCA was used to reduce the dimensionality of synchrophasor data and DBSCAN was applied to the resultant components to remove the bad noisy data due to the highly dense nature of the data. In a steady state, there exists only a few clusters needed to represent the power system but in a dynamic system such as the power grid which is ever changing, the number of clusters (k) can't be known beforehand. DBSCAN has shown that it can detect these uncorrelated signals and adjust the number of clusters to the ever-changing state of the dataset. The results of which proved that this method could be used in conjunction with Spark to alleviate the "cure of

dimensionality". Looking at previous work in this area from the transmission side of the network in the work of (Dahal, King and Madani, 2012) where the authors used 8 voltage measurements that reduced to 1 principal component capturing 95%-98% of the energy, compared to 2 principal components capturing 96.5% of the energy from 15 measurements in this research report from the distribution side using a real-world dataset. More components are necessary to capture the same amount of information than would be required from the transmission side of the network. This could be because the distribution network in the US is not all one network like the transmission side and most of the power outages occur at the distribution side of the network. The higher resolutions and precise time synchronisations of synchrophasors could be revealing subtle changes in the dynamics of the distribution network where voltages are orders of magnitude lower than that of the transmission network (Arghandeh -Florida *et al.*, 2018). But, looking at the current magnitudes where 2 components were needed to represent 98% of the contained variance, only 1 component was needed to capture 98.9% of the variance from the distribution side in this report. This fulfils Chapter 1, Table 1, Objective 7.

Each of the methods used solved the problem of the "curse of dimensionality" of synchrophasor data and can be used for dimensional reduction without too much loss of energy but IPCA using SVD was a lot faster and is more suitable to big data than using PCA in batch mode and so was a good choice from the point of view of memory usage. The voltages could be represented using two principal components that capture over 96.5% of the variance and one principal component can capture over 98.9% of the variance for currents. This reduction from 30 dimensions to 3 would enable the storage of data without too much loss of information and so there would be no need to delete the data which is the case of EirGrid[10] which deletes PMU data every two years. The reduced synchrophasor data with DBSCAN can be used to detect and isolate noise or other events on the line which can cause significant distortions when applying PCA (Lee *et al.*, 2019). PCA can be used to exploit the similarity between the μPMU signals that originated from the electrically coupled structure of the power system due to spatial sparsity. The desired performance criteria of efficiency and robustness i.e. averagely low reconstruction error are met. The reasons I didn't apply dimensional reduction to the phases was because of the nature of the data where phases rotate in cycles and change sign.

6 Conclusion and Future Work

One of my research objectives at the start of this project was to apply dimensional reduction techniques to real-time micro-phasor measurement units in real-time. So, from the point of view of this final research project the following research question had to be answered: *To what extent can dimensional reduction be applied at the distributional level using the following real time Micro Phasor Measurement Units (P3001199, P3001065, P3001352, P3001095, P3001289) so that maximum variance can be maintained and noise reduced?* From the point of view of the main research question I have implemented and being successful in answering the research question whereby I have reduced the size of the datasets that can capture over 99%

[10] http://www.eirgridgroup.com/about/eirgrid-group/

of the variance using only three principal components from an initial 30 dimensions and noise can be detected and removed using the DBSCAN algorithm. A real-time scalable NoSQL database was implemented and a dimensional reduction approach using DBSCAN can remove noise from the principal components so that dimensional reduction could be successful. More devices could be added to the program with very little code changes if they became available.

I created a Kafka cluster that would use my Cassandra database as a source and would monitor the Cassandra database for any new events and publish them to Spark for further continuous real-time streaming but didn't implement it into my final implementation due to integration problems. It would be advantageous to do so and so the creation of a real-time Wide Area Monitoring System[11] would become attainable if the research project was advanced by another student. This then could be commercialised and would become a part of the Smart Grid where the utilization of big datasets would make the grid more reliable, environmentally friendly and with minimal financial outlays for utilities and their stakeholders. This project can then be used and further models added to and studies done for example to detect cyber-attacks because at this stage it is a new area of study and so little work has been done in this area from the distribution side.

The results from the DBSCAN clustering analysis can now be investigated and classification labels can be added to them to create a supervised machine learning model. A further analysis can now take place to identify the other lesser clusters that are clustered together but are not a part of the main cluster, are they high frequency interference signals from another source being picked up on the distribution line? One area where the principal components can be used is in the area of event detection because as outlined by (Chen, Xie and Kumar, 2015), these events can now be caught at an earlier stage than would be the case in using the raw data.

Note: P300* are the target ids of the micro phasor measurement units.

Acknowledgements

Many thanks to my supervisor, Dr. Catherine Mulwa, for her constant support and guidance during this research project. I would also like to thank Dr. Sean P. Peisert and Dr. Reinhard Gentz at the Berkeley National Laboratory for providing me with access to the micro-phasor measurement units.

References

Akhavan-Hejazi, H. and Mohsenian-Rad, H. (2018) 'Power systems big data analytics: An assessment of paradigm shift barriers and prospects', *Energy Reports*. Elsevier Ltd, 4, pp. 91–100. doi: 10.1016/j.egyr.2017.11.002.

Andersen, M. P. *et al.* (2015) 'DISTIL: Design and implementation of a scalable synchrophasor data processing system', in *2015 IEEE International Conference on Smart Grid*

[11] https://www.gegridsolutions.com/Software_Solutions/catalog/wams.htm

Communications (SmartGridComm). IEEE, pp. 271–277. doi: 10.1109/SmartGridComm.2015.7436312.

Arghandeh -Florida, R. *et al.* (2018) 'Synchrophasor Monitoring for Distribution Systems: Technical Foundations and Applications A White Paper by the NASPI Distribution Task Team', (January), pp. 1–62. Available at: https://www.naspi.org/sites/default/files/reference_documents/naspi_distt_synchrophasor_monitoring_distribution_20180109.pdf (Accessed: 30 July 2019).

Benmouyal, G. *et al.* (2014) 'An Overview of the IEEE Standard C37.118.2-Synchrophasor Data Transfer for Power Systems', *IEEE TRANSACTIONS ON SMART GRID*, 5(4). doi: 10.1109/TSG.2014.2302016.

Bhuiyan, S. M. A. A., Khan, J. F. and Murphy, G. V. (2017) 'Big data analysis of the electric power PMU data from smart grid', *Conference Proceedings - IEEE SOUTHEASTCON*. IEEE, pp. 3–7. doi: 10.1109/SECON.2017.7925277.

Box, P. O. *et al.* (2009) *Tilburg centre for Creative Computing Dimensionality Reduction: A Comparative Review Dimensionality Reduction: A Comparative Review*. Available at: http://www.uvt.nl/ticc (Accessed: 8 October 2018).

Buyya, R. *et al.* (2016) 'Big Data Analytics on a Smart Grid: Mining PMU Data for Event and Anomaly Detection', *Big Data*. Morgan Kaufmann, pp. 417–429. doi: 10.1016/B978-0-12-805394-2.00017-9.

Chambers, M. J. *et al.* (1983) 'Graphical Methods for Data Analysis (Statistics/Probability Series)'. Available at: http://www.amazon.com/Graphical-Analysis-Wadsworth-Statistics-Probability/dp/053498052X.

Chen, Y., Xie, L. and Kumar, P. R. (2013) 'Dimensionality reduction and early event detection using online synchrophasor data', *IEEE Power and Energy Society General Meeting*. IEEE, pp. 1–5. doi: 10.1109/PESMG.2013.6672974.

Chen, Y., Xie, L. and Kumar, P. R. (2015) *Synchrophasor data---driven early anomaly detec4on via dimensionality reduc4on*. Available at: https://www.naspi.org/documents (Accessed: 21 November 2018).

Dahal, N., King, R. L. and Madani, V. (2012) 'Online dimension reduction of synchrophasor data', in *PES T&D 2012*. IEEE, pp. 1–7. doi: 10.1109/TDC.2012.6281467.

Ding, C. (2004) *K-means Clustering via Principal Component Analysis*. Banff. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.162&rep=rep1&type=pdf (Accessed: 1 August 2019).

Ganesh, B. H. *et al.* (2015) 'ScienceDirect Apache Spark a Big Data Analytics Platform for Smart Grid', *Procedia Technology*, 21, pp. 171–178. doi: 10.1016/j.protcy.2015.10.085.

Garcia Zanabria, G., Nonato, L. G. and Gomez-Nieto, E. (2016) 'iStar (i*): An interactive star coordinates approach for high-dimensional data exploration', *Computers & Graphics*. Pergamon, 60, pp. 107–118. doi: 10.1016/J.CAG.2016.08.007.

Kandogan, E. and Kandogan, E. (2000) *Star Coordinates: A Multi-dimensional Visualization Technique with Uniform Treatment of Dimensions*, *IN PROCEEDINGS OF THE IEEE INFORMATION VISUALIZATION SYMPOSIUM, LATE BREAKING HOT TOPICS*. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.8909 (Accessed: 6 June 2019).

Khan, M. *et al.* (2014) 'Big data analytics on PMU measurements', *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2014*. IEEE, pp. 715–719. doi: 10.1109/FSKD.2014.6980923.

Kiyoung Yang *et al.* (2005) 'On the Stationarity of Multivariate Time Series for Correlation-Based Data Analysis', *Proceedings - IEEE International Conference on Data Mining, ICDM*. IEEE, pp. 805–808. doi: 10.1109/ICDM.2005.109.

Lacommare, K. H., Eto, J. H. and Lawrence, E. O. (2004) *Understanding the Cost of Power Interruptions to U.S. Electricity Consumers Environmental Energy Technologies Division*. Berkeley. Available at: http://eetd.lbl.gov/ea/EMP/EMP-pubs.html (Accessed: 14 June 2019).

Lee, G. *et al.* (2019) 'Multiscale PMU Data Compression via Density-Based WAMS Clustering Analysis', *Energies*. Multidisciplinary Digital Publishing Institute, 12(4), p. 617. doi: 10.3390/en12040617.

M. Fayyad, U. (1994) 'KDD-94: AAAI-94 Workshop on Knowledge Discovery in Databases', *KDD-94 Workshop and Ramasamy Uthurusamy*, p. 2. Available at: https://www.aaai.org/Papers/Workshops/1994/WS-94-03/WS94-03-000.pdf.

Nguyen, M. *et al.* (2017) *m-TSNE: A Framework for Visualizing High-Dimensional Multivariate Time Series*. Available at: http://arxiv.org/abs/1708.07942 (Accessed: 30 May 2019).

Niazazari, I. and Livani, H. (2018) 'Disruptive event classification using PMU data in distribution networks', *IEEE Power and Energy Society General Meeting*, 2018-Janua, pp. 1–5. doi: 10.1109/PESGM.2017.8274154.

Peisert, S. *et al.* (2018) *LBNL Open Power Data*. Available at: https://powerdata-explore.lbl. (Accessed: 30 November 2018).

PHADKE, A. G. and BI, T. (2018) 'Phasor measurement units, WAMS, and their applications in protection and control of power systems', *Journal of Modern Power Systems and Clean Energy*. Springer Berlin Heidelberg, 6(4), pp. 619–629. doi: 10.1007/s40565-018-0423-3.

Shand, C. *et al.* (2015) 'Exploiting massive PMU data analysis for LV distribution network model validation', in *2015 50th International Universities Power Engineering Conference (UPEC)*. IEEE, pp. 1–4. doi: 10.1109/UPEC.2015.7339798.

Thinsungnoen, T. *et al.* (2015) 'The Clustering Validity with Silhouette and Sum of Squared Errors'. doi: 10.12792/iciae2015.012.

Vittal, V. (2012) 'Application of phasor measurements for dynamic security assessment using

decision trees', *IEEE Power and Energy Society General Meeting*. IEEE, pp. 1–3. doi: 10.1109/PESGM.2012.6344580.

Wang, W. and Yang, J. (2009) 'Mining High-Dimensional Data', in *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US, pp. 803–808. doi: 10.1007/978-0-387-09823-4_41.

Wang, X. *et al.* (2019) 'Online Identification and Data Recovery for PMU Data Manipulation Attack', *IEEE Transactions on Smart Grid*. IEEE, PP(c), pp. 1–1. doi: 10.1109/tsg.2019.2892423.

Wold, S., Esbensen, K. and Geladi, P. (1987) *Principal Component Analysis*. Available at: https://imedea.uib-csic.es/master/cambioglobal/Modulo_V_cod101615/Theory/lit_support/pca_wold.pdf (Accessed: 30 March 2019).

Xie, L., Chen, Y. and Kumar, P. R. (2014) 'Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis', *IEEE Transactions on Power Systems*, 29(6), pp. 2784–2794. doi: 10.1109/TPWRS.2014.2316476.

Yang, B. *et al.* (2015) 'Big data analytic empowered grid applications - Is PMU a big data issue?', *International Conference on the European Energy Market, EEM*. IEEE, 2015-Augus(May 2015), pp. 1–4. doi: 10.1109/EEM.2015.7216718.

Yang, G. *et al.* (2009) 'PMU Applications–From Situation Awareness to Blackout Prevention', *Siemens-Future Energy …*. Available at: http://orbit.dtu.dk/ws/files/119170111/PMU_Applications_From_Situation_Awareness_to_Blackout_Prevention.pdf.

Yang, Z. *et al.* (2018) 'A Novel PMU Fog Based Early Anomaly Detection for an Efficient Wide Area PMU Network', *2018 IEEE 2nd International Conference on Fog and Edge Computing, ICFEC 2018 - In conjunction with 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE/ACM CCGrid 2018*, pp. 1–10. doi: 10.1109/CFEC.2018.8358730.

Zhang, Y., Huang, T. and Bompard, E. F. (2018) 'Big data analytics in smart grids: a review', *Energy Informatics*. Springer International Publishing, 1(1), p. 8. doi: 10.1186/s42162-018-0007-5.

Zhou, Y. *et al.* (2016) *UC Berkeley Sustainable Infrastructures Title Abnormal event detection with high resolution micro-PMU data Publication Date Distribution Network Event Detection with Ensembles of Bundle Classifiers*. Available at: https://escholarship.org/uc/item/53w685k5 (Accessed: 7 May 2019).