POLITECNICO DI MILANO

School of Industrial and Information Engineering

Master Thesis in Electrical Engineering

Artificial Intelligence Applications for Residential LV Distribution Networks

Author:

Angela Simonovska, 10597084

Supervisor: Prof. Marco Merlo

Co-Supervisor: Prof. Luis F.Ochoa

Milan, April 2020

(Page intentionally left blank)

Contents

Table of figures

Table of tables

Declaration

This is to certify that

1. the thesis comprises only of my original work towards the degree of Master of Science,

2. due acknowledgement has been made in the text to all other material used.

Copyright statement

Dedication

This Thesis is dedicated to my lovely parents

Dragan and Snezana.

They have raised me to become the person I am today

and have always supported me through all these years.

Without them, none of this would have been possible.

Acknowledgement

Foremost, I want to express my deepest appreciation to my supervisor Prof. Luis F. Ochoa, whose support and continuous supervision made the completion of this thesis possible. His expertise and guidance helped me in all the time of research and writing of this thesis.

Also, I would like to thank my supervisor Prof. Marco Merlo, for his support during the last six months, as well as his valuable and constructive suggestions for completion of this thesis.

Special thanks to my friends in Milan: Hristina, Aleksandar, Ivana, Dejan, Jelena, Vinicius and Shahrukh. Your support and friendship mean a lot to me and I could not imagine my master studies without you being around me. I will always appreciate the time spent together in Milan.

I would also like to thank all the members of the team in Melbourne and my Space Lab neighbors Bastian and Carmen, as well as my friends Goran and Gleice for their friendship and support.

Last but by no means least, I would like to express my sincere and uttermost appreciation to my partner Zoran, whose support and care were so much helpful in the last year. I owe my motivation and dedication for completing this thesis to him.

Executive Summary

The amount of residential solar photovoltaic (PV) installations in Australia and around the world has significantly increased in the last few years. The resulting reverse power flows are creating significant challenges for distribution companies to manage voltages as customers are now experiencing voltage rise issues. Therefore, distribution companies are recognizing the need for adequate three-phase low voltage (LV) feeder models so detailed studies that identify potential impacts and solutions can be carried out.

The growing adoption of smart meters (or similar) across Australia and around the world gives an opportunity to exploit the corresponding data using advanced analytics to identify the phase connectivity. Using only voltage time-series data extracted from smart meters, -this work firstly proposes an accurate algorithm based on clustering techniques that can determine the phase group to which each customer is connected to in a given LV feeder.

The large deployment of the smart meters in residential LV distribution networks, provides a useful amount of data that could be beneficial to distribution companies for solving the voltage regulation problem by calculating the voltages of the customers. By only using the active and reactive power extracted from the smart meters of each customer, -this work then proposes an accurate model based on deep learning method that can determine the voltage profile of each customer in a given LV feeder.

The work presented in this thesis was done due to a research project focused on artificial intelligence application in residential LV distribution networks, taking as example a LV distribution network in Australia at the University of Melbourne, Australia with constant collaboration and consultation with my co-supervisor prof. Luis F. Ochoa.

Keywords: phase grouping, k-means clustering, voltage calculation, deep neural network model, data-driven model.

Sintesi

La quantità di impianti fotovoltaici residenziali in Australia e nel mondo è aumentata in modo significativo negli ultimi anni. I flussi di energia inversa che ne derivano stanno creando sfide significative per le società di distribuzione di energia elettrica per gestire il voltaggio, in quanto i clienti riscontrano problemi di aumento della tensione. Pertanto, le società di distribuzione riconoscono la necessità di adeguati modelli di alimentatori trifase a bassa tensione (BT) per poter realizzare studi dettagliati che identifichino potenziali impatti e soluzioni.

La crescente adozione di contatori intelligenti (o simili) in Australia e in tutto il mondo offre l'opportunità di sfruttare i dati utilizzando analisi avanzate per identificare la connettività di fase. Utilizzando unicamente i dati di serie temporali di tensione estratti da contatori intelligenti, questo lavoro, in primo luogo, propone un algoritmo accurato basato su tecniche di clustering in grado di determinare il gruppo di fase al quale ciascun cliente è collegato in un determinato alimentatore BT.

Il vasto impiego di contatori intelligenti nelle reti di distribuzione BT residenziali, fornisce una quantità di dati che potrebbero essere utili alle società di distribuzione per risolvere il problema della regolazione della tensione calcolando le tensioni dei clienti. Questo lavoro propone un modello accurato basato su metodi di “deep learning” il quale può determinare il profilo di tensione di ciascun cliente in un determinato alimentatore BT utilizzando solo la potenza attiva e reattiva estratta dai contatori intelligenti di ciascun cliente.

Parole chiave: clustering di fasi, clustering di medie k, calcolo della tensione, modello di rete neurale, modello basato sui dati.

1 Introduction

The deployment of distributed energy resources (DERs) has seen a significant rise in the last few years in Australia and around the world. Due to the increased *distributed* generation, the amount of produced power is increasing, causing different challenges for the residential LV distribution network, such as voltage fluctuations, reverse power flows, etc.

In order to determine the effects caused by the DERs, the distribution operations have a necessity to determine the topology of the residential LV distribution network and monitor the voltage of the customers.

On the other hand, the amount of data provided by the smart meters has rapidly increased in the last three years due to their large deployment, such that every household in Victoria (Australia) has one. The readings provided by these meters usually have a time resolution of thirty minutes, which on daily basis counts forty-eight readings that are stored in the data bases of the electricity distribution companies. There are different applications where this data can be used.

An area that uses data as its primary source is the Artificial Intelligence. Artificial intelligence is a set of techniques that enable machines to mimic human behaviour. It is the theory of development of computer systems able to perform tasks normally requiring human intelligence such as visual readings, speech recognition, decision making, etc.

One of the most important subsets of artificial intelligence are machine learning and deep learning methods. The concept of these methods is based on feeding a machine with data and making it learn in order to perform its own decisions. A model is trained using the machine learning algorithm for different purposes and applications. Whether the data is well labeled or not, we define different types of machine learning.

Using artificial intelligence tools for determining different states in residential LV distribution networks brings a lot of benefits related to computational time of the analysis, as well as abandoning long power flow simulations.

The main advantage is by only using data-driven models to be able to perform different analyses based on different machine learning and deep learning algorithms and explore the behaviour of the distribution network.

In case where there is a new consumer and/or prosumer joining the LV distribution network, the distribution network system provider needs to be able to determine to which phase the new consumer or prosumer will be connected, such that to avoid asset overloading on the same phase and provide better connection, reliability, security and stability for the new connected customer. In other words, accurate network and phase connectivity models are crucial to distribution system analytics. The phase to which residential customers are connected to is not initially known by distribution network system providers. Most of the time, phase connectivity data is typically missing or is erroneous. Many papers and analysis are done in order to determine the phase connection of the customers based on long power flow simulations.

1.1 LV Networks: Challenges

The amount of distributed energy resources (DER) has been significantly increasing in the last years. Frequent voltage fluctuations caused by the increased deployment of DER, demand response programs, and electric vehicles, challenge modern LV distribution networks. Electric utilities are experiencing major issues related to unprecedented levels of load peaks as well as renewable (mostly photovoltaic (PV)) penetration.

For instance, a solar farm connected at the end of a long distribution feeder in a rural area can cause voltage deviation along the feeder. Moreover, over-voltage happens during midday when the PV generation reaches its peak, while the load demand is relatively low. On the other hand,

voltage sags occur mostly during night period due to low PV generation, as well as due to the increased deployment of electric vehicle connections. This highly motivates voltage regulation. The task of maintaining voltage magnitudes, without causing violations, is critical in LV distribution networks.

The challenges that the LV distribution network is experiencing are mostly based on the absence of different models and algorithms for determining the influence of the DER. That is why, it is crucial for the distribution operators to define different models in order to determine or confirm the topology of the LV distribution network as well as to follow the voltage fluctuations caused by the increased PV penetration.

1.1.1 Modelling

One of the most important ways in estimating the effects from DER is by having suitable models for defining the topology as well as the voltage profiles of the customers in a residential LV distribution network.

So far, in the literature the existing models are mostly based on power flow simulations in which the information about the topology of the LV distribution network is already known or partially known. The number of existing models based on smart meter data is very low. Moreover, the absence of models completely based on data provided by smart meters taken as Input for different AI applications are the key motivation for the research questions elaborated in this thesis.

1.1.2 DER Effects

In order to understand the effects of the DERs and maintain voltage magnitudes, distributed network system providers (DNSPs) need the network topology information while determining the voltage is crucial. The reported topology of a power system is not necessarily always correct. As

the topological structure of the power system changes during operation as well as with the fast deployment of DER, it must be checked before any power system calculation.

One of the best ways to estimate the impacts and the effects of the DERs is by having models. The models would be able to provide information about the topology of the network and perform state estimation in a very short time period. Apart from the existing models based on power flow simulations, prospective and promising models could be built based on artificial intelligence.

1.2 Smart Meters

In the last few years, a large deployment of smart meters in residential LV distribution networks has been noticed in Australia and around the world. In Victoria, a major upgrade in the electricity structure has been done in 2016, such that all households and small businesses in Victoria had their metes upgraded to smart meters.

A smart meter is a device that digitally measures energy use. It measures when and how much electricity is used at the premises, after which it sends this information back to the energy retailer remotely. Smart meters typically record energy in 30-minute intervals or more frequently, and report at least daily. They communicate meter readings directly to DNSPs, eliminating the need for someone to come out and read meters. Also, they give notification to the DNSP in real-time if a premises' power is out.

With the large deployment of smart meters, more and more data were provided to the distribution operators. The increment of the amount of data could give a plentiful amount of opportunities for its processing and further investigations by the DNSPs. With the installation of smart meters and other smart grid sensing devices, DNSPs by extracting the active and reactive power as well as the voltage profile data per customer from the 30-minute (or more frequent) interval readings, would be overloaded with unprocessed raw data which may result with unrealistic benefits.

The best usage of the smart meter data and demonstration of its value could be by performing data analytics. Data analytics is a process of inspecting, cleaning, transforming and modelling data with the goal of discovering useful information, including conclusion and supporting decision-making. Data analytics is performed under the roots of Artificial Intelligence (AI).

By having the historical data extracted by smart meters, for all the customers connected to a residential LV distribution network, DNSPs could lean on AI applications in order to define different states of the network. Moreover, AI applications could replace power flow simulations, perform faster and get results in terms of seconds with very high accuracy, giving promising future models.

1.3 AI Opportunities

Artificial Intelligence (AI) is a technique that enables machines to mimic human behavior. It is the theory of development of computer systems able to perform tasks normally requiring human intelligence. AI is becoming significantly impactful on various sectors. The energy sector is of no exception. The implementation of the AI applications in electric power systems could be a step forward regarding distribution network performance, planning and forecasting, operation, maintenance, analysis, security assessment, etc.

The applications of AI are based on its two main subsets named Machine Learning (ML) and Deep Learning (DL). ML uses methods in which a large amount of data can be fed to a machine and make that machine learn and make its own decisions. At the same time, DL is based on neural networks that are similar to humans brain, divided in multiple layers, learning how to predict the output by training and updating their weights, such that for any Input they are able to predict/determine the output.

Due to increasing complexity, uncertainty, and data dimension in a power system, conventional methods often meet bottlenecks when attempting to solve decisions, operation and control

problems. By only having the data from smart meters, data-driven models based on ML and DL could be implemented as replacement for the long power flow simulations.

The advantage of these data-driven models is their ability to learn nonlinear problem offline with selective training and once they are trained properly, they can perform the same algorithm in terms of seconds with very high accuracy. Therefore, by using data-driven models, DNSPs could be able to determine the network topology and perform voltage calculations.

In this thesis, two research questions are presented based on the relationship between smart meter data and artificial intelligence. The first one is related to defining the phase connectivity of the customers by using ML algorithms, while the second one calculates the voltages of the customers by using deep neural network models.

The research questions are briefly introduced in the following sub-sections.

1.3.1 Phase Grouping Using Clustering Techniques

In the first part of the Thesis, an innovative approach for phase grouping is presented using clustering techniques based on unsupervised machine learning. The proposed phase grouping algorithm consists of: a feature reduction method (named principal component analysis) that extracts the main features of the voltage time-series data, an unconstrained k-means clustering algorithm that sorts those features in three groups according to their similarities, and a one-to-one matching algorithm that determines the most likely phase group for each customer. Differently from other phase grouping algorithms in the literature, the proposed approach does not a pre-defined (known) phase connection for some of the customers.

This a novel, faster method that does not require long power flow simulations in which all the information about the topology of the power system should be provided for determining the phases of the consumers. Moreover, having only the voltage time-series data for the whole day

(forty-eight voltage readings for 24 hours with 30- min. resolution) different case studies are obtained considering the night-time period (midnight until 7 AM in the morning) or low demand hours, whole day (12 AM – 12PM) and afternoon period (4 PM - 21PM) or high demand hours for validation of the algorithm. The studies are performed for one- and two-weeks of voltage time-series data, and the results are represented in the end of the section showing the performance and accuracy of the artificial intelligence tool for phase identification of customers.

Based on the phase grouping algorithm, it is possible to determine the phase to which each consumer is connected to in an Australian LV distribution network. The algorithm firstly sorts the customers according to their voltage time-series data in three different groups and then defines the correlation between the groups and the phases.

Differently from other phase grouping algorithms in the literature, the proposed approach does not a pre-defined (known) phase connection for some of the customers. This makes it even more practical and cost-effective to distribution companies.

The whole algorithm for phase grouping is done based on machine learning methods and algorithms, showing the capability of the AI in determining different states of the power system.

1.3.2 Voltage Calculation Using Neural Networks

In the second part of the Thesis, an algorithm for calculating the voltage profile of the customers is presented using neural network model based on deep learning. The deep neural network model consists of four layers: one Input, two hidden, and one output layers. Each of the layers has a different number of neurons (or nodes) that were determined based on trial and error calculations.

Differently from other voltage calculation algorithms based on neural networks in the literature, this model takes as Input only the active and reactive power extracted from the smart meter data

of the customers. By only having the data provided by the smart meters, a deep neural network model is trained in order to find the relationship between the Input data and the output (voltage of the customer). The model is trained taking 30 days of active and reactive power data extracted from the smart meter and later different tests were performed in order to define the accuracy of the algorithm, considering different days from different seasons, double net demand and PV generation.

The voltage calculations were performed taking into consideration one customer and two customers (connected to the same phase) in Australian LV distribution feeder, and the results are represented in the end of the second section showing the performance and accuracy of the artificial intelligence tool for calculating the voltage profile of the customers.

The whole algorithm for voltage calculation is based on deep learning algorithm, showing again the capability of the AI in determining different states of the LV distribution network

2 Phase Grouping Using Clustering Techniques

The amount of residential solar photovoltaic (PV) installations has significantly increased around the world in the last few years. The resulting reverse power flows are creating significant challenges for distribution companies to manage voltages as customers are now experiencing voltage rise issues. Therefore, distribution companies are recognizing the need for adequate three-phase low voltage (LV) feeder models so detailed studies that identify potential impacts and solutions can be carried out. However, one of the major challenges in producing LV models is the absence of knowledge or confidence in the phase connection of residential customers.

The growing adoption of smart meters (or similar) across Australia and around the world gives an opportunity to exploit the corresponding data using advanced analytics to identify the phase connectivity. Using only voltage time-series data extracted from smart meters, -this work proposes an accurate algorithm based on clustering techniques that can determine the phase group to which each customer is connected to in a given LV feeder.

The proposed phase grouping algorithm consists of: a feature reduction method (named principal component analysis) that extracts the main features of the voltage time-series data, an unconstrained k-means clustering algorithm that sorts those features in three groups according to their similarities, and a one-to-one matching algorithm that determines the most likely phase group for each customer. Differently from other phase grouping algorithms in the literature, the proposed approach does not a pre-defined (known) phase connection for some of the customers. This makes it even more practical and cost-effective to distribution companies.

The performance of the algorithm is demonstrated on a realistic Australian LV feeder with thirty-one customers considering one and two weeks-worth of 30-min voltage time-series data with and without solar PV. In both cases, the results are promising as all customers were accurately allocated to their corresponding phase group.

2.1 Literature review

A variety of approaches for phase identification exist in the literature. [1] is using voltage pattern clustering in order to define the phases of the customers, meanwhile reducing the dimensionality of the time-series data by using the discrete wavelet transformation (DWT). Besides the clustering technique, this paper uses Support Vector Machine (SVM) technique for the phase correlation. In [2] spectral clustering is used in conjunction with sliding window ensemble in order to improve the scalability of the algorithm for large datasets. [3] uses a load summing approach, summing the individual customer loads and comparing the results to the load at the transformers and substation. This approach requires solving the linear equations produced by this method. Several approaches using different types of clustering with correlation coefficients have been attempted before. [3], [4] use hierarchical clustering. [5] uses a Constrained k-means implementation, and [6] uses a constrained multi-tree algorithm to do phase identification. Both of these methods use the underlying topology as constraints in their algorithms. The customer-transformer connection labelling is used as 'must-link' constraints to reduce the number of possible pairings of customers. This approach reduces the complexity of the clustering problem but requires the assumption that all of the customer-transformer labelling is correct, otherwise this approach propagates the errors introduced by building those labels into the clustering algorithm

Using only voltage time-series data extracted from smart meters, -this work proposes an accurate algorithm based on clustering techniques that can determine the phase group to which each customer is connected to in a given LV feeder.

Differently from other phase grouping algorithms in the literature, the proposed approach does not a pre-defined (known) phase connection for some of the customers. This makes it even more practical and cost-effective to distribution companies.

2.2 Machine Learning

Machine learning is one of the most important subsets of Artificial Intelligence. It is a method in which a large amount of data can be fed to a machine and make that machine learn and make its own decisions.

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T as measured by P improves with experience E. The iterative aspect of machine learning has a crucial importance, because as models are exposed to new set of data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. Resurging interest in machine learning is due to growing volumes and varieties of available data, computational processing that is cheaper and more powerful, and affordable data storage.

In figure 1, the basic concept of machine learning is represented. In order to build a machine learning model that will predict outcome, a data Input should be given for training purposes. Due to this training, many iterations are performed in order to build the best model that will have high accuracy in predicting the outcome.

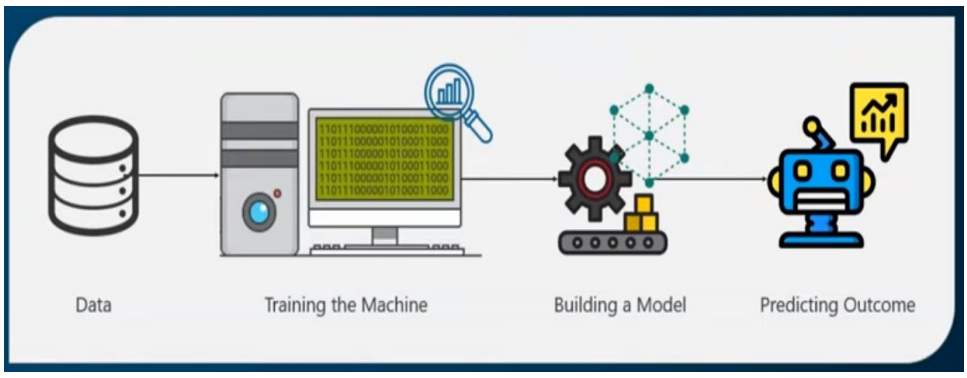


Figure 1. Machine learning model

Some popular machine learning methods are:

- Supervised learning
- Unsupervised learning
- Semi supervised learning
- Reinforcement learning

2.2.1 Supervised learning

The supervised learning algorithms are trained using labelled i.e. known examples, such as: an input where the desired output is known. The learning algorithms works in such a way that: it receives a set of inputs along with the corresponding correct outputs, and the algorithm learns by comparing its actual output with correct outputs to find errors. It then modifies the model accordingly. Through methods like classification, regression, prediction etc. supervised learning uses patterns to predict the values of the label on additional unlabelled data. It is most commonly used in applications where historical data predicts likely future events.

2.2.2 Unsupervised learning

The unsupervised learning algorithms are used against data that has no historical labels. The system is not told the "right answer". The main goal of the algorithm is to figure out by itself what is being shown to it. i.e. to explore the data and find some structure within. Popular unsupervised learning applications are nearest-neighbour mapping, k-means clustering and singular value decomposition. The k-means algorithm is used in this thesis for finding the phases of the consumers.

2.2.3 Semi supervised learning

This type of machine learning is used for the same applications as supervised learning, but it uses both labelled and unlabelled data for training. Typically, proportionally it is composed of small amount of labelled data and large amount of unlabelled data (because unlabelled data is less

expensive and takes less effort to acquire). Semi supervised learning is useful when the cost associated with labelling is too high to allow for a fully labelled training process.

2.2.4 Reinforcement learning

Reinforcement learning is used mostly for robotics, gaming and navigation. With this type of learning, the algorithm discovers through trial and error which actions yield the greatest reward. It has three primary components: agent (the learner), the environment (everything the agent interacts with) and actions (what the agent can do). The main goal is for the agent to choose actions such that it will maximize the expected reward over a given amount of time.

Some examples of applications of machine learning and artificial intelligence in general in power systems are the following:

- creation of digital twin network,
- creation of a model of high-tension cables in order to suppress false positives out of the DSO's maintenance plan,
- creation of flow rerouting feature,
- intelligent analysis and self-healing control,
- fault detection and location,
- image recognition of power lines, etc.

2.3 Methodology

The framework of the proposed phase grouping algorithm using clustering techniques is illustrated in Figure 2.

The framework explains the following:

1. In the first step, voltage time-series data is obtained as result of the power flow simulation in Python driving OpenDSS, representing the smart meter data.

2. In the second step, normalization of the customer voltage time-series is done by their standard deviations.
3. Then principal component analysis method is applied to the normalized time-series data to extract the top q components.
4. After extracting the top q components, they are normalized again and taken as an Input to the k-means clustering algorithm to partition customers into clusters.
5. At last, the phase is identified of each cluster by solving a minimization problem.

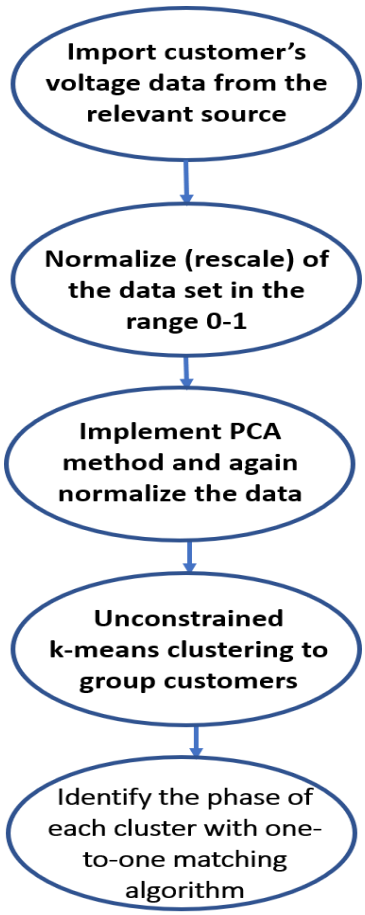


Figure 2. Algorithm for phase grouping of customer's voltages

Before starting with phase identification algorithm, the historical voltage time-series data should be taken from the smart meters. In every machine learning algorithm, the amount of data has a

huge role in its performance. Moreover, data is the main fuel for any artificial intelligence tool. Without data, it is impossible to use any tool of the area of AI.

Following are the explained steps of performing the phase grouping algorithm using clustering techniques.

2.3.1 Voltage data and measurements

The data taken as an input for the phase grouping algorithm is based on the voltage time-series data of residential LV distribution network in Victoria, Australia. For the purpose of explanation, at this stage only one feeder from the Australian LV distribution network will be taken into consideration. Moreover, one representative day is chosen for depicting the voltage profiles of the consumers fed by feeder one. (January 3rd, summer day in Australia).

Following the algorithm for phase identification of customer's voltages, in figure 3 the voltage profile of the thirty-one consumer for January 3rd is represented. The case with 100% PV penetration is chosen for better visual representation of the voltages of the consumers, as well as the different voltage profiles of the consumers are more visible.

This part of the thesis shows graphically the voltage behaviour of the consumers fed by feeder one, in the LV distribution network. The voltage time-series data obtained in this section is the input data for the phase grouping algorithm as an artificial intelligence tool for defining the phases of the consumers. These voltage profile data set is firstly normalized and centered and then used as an input data for the principal component analysis method.

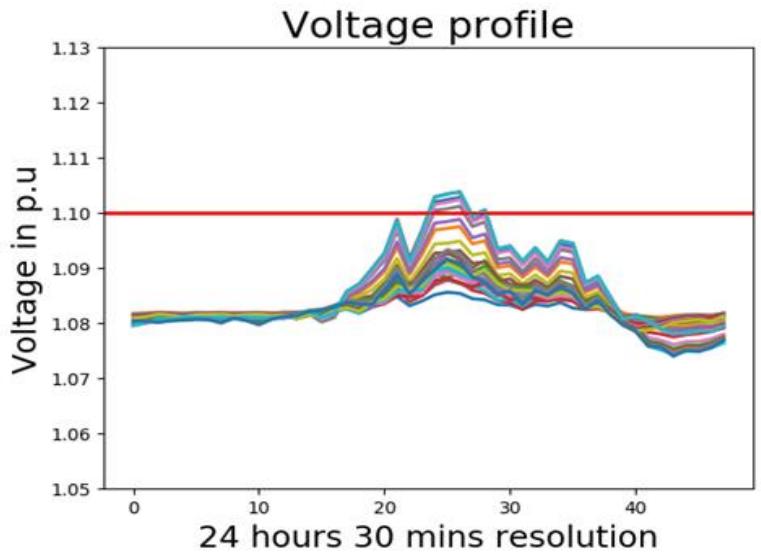


Figure 3. Voltage profile with 100% PV penetration

After importing the voltage time-series data and after normalizing and centre it in the range of -1 and 1, this voltage time-series data is taken as an Input for the PCA method explained below.

2.3.2 Principal Component Analysis Method

Principal Component Analysis (PCA) method [15] is a method of identifying features in data. It is a learning feature reduction method that is used to reduce data dimension and extract key features (patterns) of the original data in such a way as to express their similarities and differences. Since features in data with high dimension can be hard to find, and graphical representation is difficult to be represented, PCA is a powerful tool to analyse the data.

In this case, since we have thirty-one consumers fed by feeder one and each one of them has forty-eight voltage time-series readings per day, it is necessary to reduce the dimension of the voltage time-series data using PCA method, such that its output will be later used as an Input for the clustering algorithm.

The main advantage of the PCA method is that once we define the features (patterns) in the voltage time-series data and we compress the data, we are not going to lose information by reducing the number of dimensions.

This section will take you through the steps that are needed to perform a principal component analysis on a voltage time-series data, taking as a reference only one day of the year for better understanding of the method. Later, different case studies will be represented related to the number of days taken for implementing the algorithms as well as the number of voltage time-series data points per day.

Framework of the PCA method:

The framework of the principal component analysis method is explained in the following steps:

Step 1: Get data

In this step the voltage time-series data for all the consumers fed by feeder one is imported and taken as an Input. Since the voltage time-series data is taken per day every 30 minutes, that means that in a period of 24 hours, we have forty-eight readings which lead us to forty-eight dimensions of the data set. The representation of forty-eight -dimensional data set is difficult, that is why the PCA method is used for data reduction.

Step 2: Normalize the data set and subtract the mean

For the PCA method to work properly, firstly the data set is normalized and centered and the mean value is subtracted from each of the data dimensions. Since we have forty-eight dimensions for forty-eight voltage time-series data points, the mean subtracted is the average value across each dimension. After subtracting the mean value, the voltage time-series data set obtains new values.

Step 3: Calculate the covariance matrix

In this step the covariance matrix is calculated on the basis of the data set obtained in step 2. Here the dimension of the data set with thirty-one consumers with forty-eight voltage time-series data points is thirty-one x forty-eight, so the covariance matrix will have a dimension of forty-eight x forty-eight. The formula for covariance is given below:

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(Xi - \bar{X})(Yi - \bar{Y})}{(n-1)}$$

Where Xi and Yi are two different voltage readings in two different timesteps, while $\overline{\mathrm{X}}$ and $\overline{\mathrm{Y}}$are the mean values per time step. The covariance is necessary to find out how much the voltage time-series data points vary from the mean with respect to each other.

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

For defining the top principal components of crucial importance is to do an eigen decomposition on the covariance matrix (obtained in the previous step). Eigenvectors represent the vector directions of the new feature space and eigenvalues represent the magnitudes of those vectors. Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the principal components of the voltage time-series data.

An eigenvalue is a number that is equal to the sum of squared distances of all projections of the voltage time-series data points to all forty-eight possible lines i.e. forty-eight possible principal components. That sum of squared distances of the projections of all voltage data point to each one of the lines is called variance.

Step 5: Choose components and form a feature vector

Once the eigenvectors are obtained from the covariance matrix, the next step is to order them by eigenvalue i.e. by the variance from highest to lowest, which will give us the principal components (PCs) in order of significance. The PCs represent the direction of the data that explain a maximum amount of variance, i.e. the line that captures most information of the data.

The relationship between variance and information here is that: the larger the variance carried by a line – the larger the dispersion of the voltage data points along it. The larger the dispersion along a line – the more information it has. Organizing information in PCs in this way will allow us to reduce dimensionality without losing much information.

So, we order the eigenvalues from highest to lowest and we do the same with the eigenvectors corresponding to the eigenvalues. In this way we form a matrix of vectors, whose columns from left to right are eigenvectors with highest eigenvalue (variance) to the ones with lowest eigenvalue.

Step 6: Deriving the new data set

This is the final step before deriving the PCs. The final equation in order to get the matrix of PCs is the following one: We take the transpose of the eigenvector matrix and multiply on the left of the difference between the original data set and the mean values - transposed.

$$Final\ data = (Eigenvector\ matrix)^T x\ (Original\ data - Mean\ values)^T$$

In this way we get thirty-one PCs (the number of PCs always takes the lower number between the Input variables and samples which in this case is the number of loads and voltage time-series data points respectively). From the total number of thirty-one PCs depicted in figure 9, only the first two have the highest variance of the voltage time-series data with respect to the others.

It can be seen from the Scree plot in figure 4 that the first principal component has a variance equal to 82% while the second principal component has variance equal to 16%. All the following PCs do not have a significant value for their variance which means that they are not able to represent the voltage time-series data set i.e. those PCs have high loss of information related to the voltage profiles of the consumers.

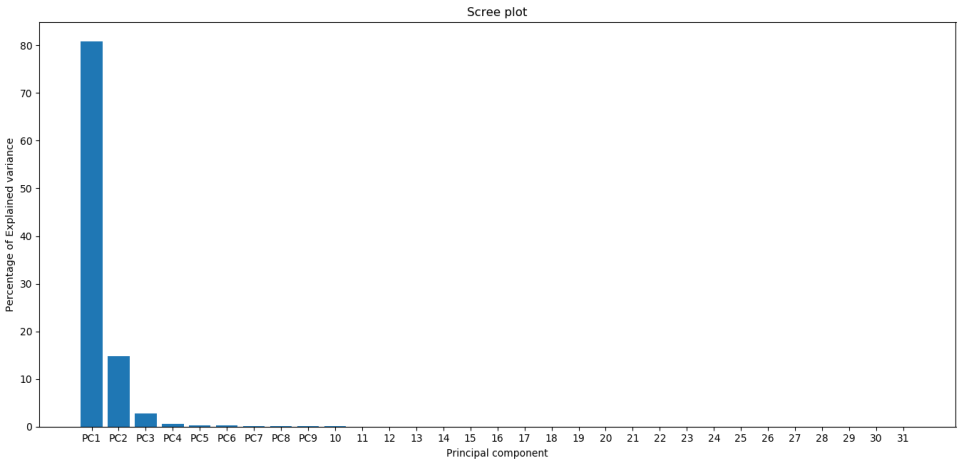


Figure 4. Scree plot of the PCs according to their variance

After obtaining the variances for all possible PCs, we can say that for a given set of data vector (voltage time-series data set) the principal component axes are those orthogonal axes onto which the variance retained under projection is maximal.

For this reason, the principal component graph depicted in figure 5 has x-axis equal to the first PC and y-axis equal to the second PC. PCA graph is a graphical representation of the consumers according to their voltages after reducing the dimensionality of their voltage time-series data. Each number in figure 5 corresponds to each one of the consumers fed by feeder 1.

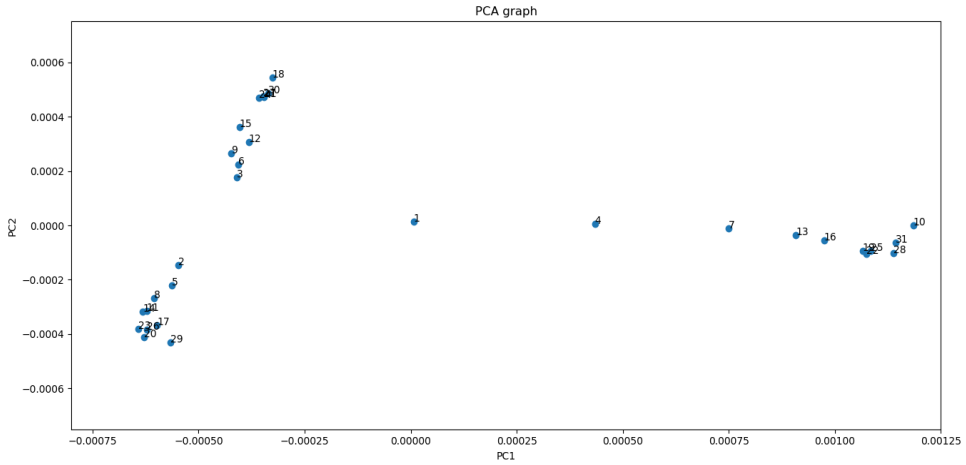


Figure 5. PCA graph of the consumers

There is a total number of thirty-one data points. It can be seen from the graph depicted above, that even after using the PCA method on the voltage time-series data, the consumers are starting to divide into three noticeable groups. However, there are still some data points i.e. consumers that are not vividly connected to one of the groups, so at this point they are considered as outliers.

The PCA method is a great starting point in the algorithm for defining the phases of the consumers considering only the voltage time-series data. The output obtained from the PCA method is taken as an Input for the clustering algorithm that is explained in the next subchapter.

2.3.3 K-means clustering

K- means is a clustering algorithm that aims to partition n observations into k clusters. The main goal of the clustering algorithm is to identify the structure in an unlabelled dataset by objectively organizing data into homogeneous groups such that the objects in the same group are more similar than those in different groups. Many clustering algorithms were developed to cluster time-series data. One of the most famous ones are the supervised and unsupervised k-means clustering or in other words the constrained and unconstrained k-means clustering respectively.

The main objective of the supervised or constrained k-means clustering algorithm is based on already known constraints that are used in the algorithm for better performing. Such an example can be if in case of defining the phase of the consumer the distribution network system provider is already familiar with some of the phases of some consumers, so that information is included in the algorithm in such a way that every single time the algorithm is called, the known phase of the consumers will be defined from the beginning. Other example can be when there are outliers i.e. data points that do not belong to any of the defined groups. In this situation the constraints are defined for those outliers such that in each calculation of the algorithm those data points have pre-defined group, so they will not behave as outliers anymore.

On the other hand, the unsupervised or unconstrained k-means clustering is based on unknown information, i.e. the algorithm has no previous information in defining the phases of the customers, so it learns by itself by improving the value of the minimum distance – the Euclidean distance. Here, even the possible outliers are always connected to one of the defined groups. In order to find the minimum distance, the k-means algorithm is sustained of steps which are explained below.

In this Thesis paper the unsupervised k-means clustering algorithm is implemented on the voltage time-series data for defining the phases of the consumers. That means that no previous information for the phases of the consumers was known and the results are obtained only by self-learning of the algorithm.

The base on which the k-means clustering algorithm is built on is the minimal sum of squared distances between the data points and the centroid (the centre of the cluster). So, the base of the k-means clustering algorithm is a distance function. There are many different types of distance functions. The first one is the Euclidean distance.

If a_i and a_j are two p - dimensional time-series, then their Euclidean distance is defined by:

$$d_E = \sqrt{\sum_{k=1}^{p} (a_{ik} - a_{jk})^2}$$

Another type of distance function is related to Pearson's correlation coefficient. For two $\mathrm{p}-$ dimensional time-series $\mathrm{a_i}$ and $\mathrm{a_j}$, their Pearson's correlation factor is defined by:

$$cc = \frac{\sum_{k=1}^{p} (a_{ik} - \mu_{jk})(a_{jk} - \mu_{jk})}{s_i s_j}$$

where μ_i and μ_j are the mean values of $\mathrm{a_i}$ and $\mathrm{a_j}$, and $\mathrm{s_i} = \sqrt{\sum_{\mathrm{k=1}}^{\mathrm{p}} (\mathrm{a_{ik}} - \mu_\mathrm{j})^2}$ (reference 9 from paper). Then, the distance between $\mathrm{a_i}$ and $\mathrm{a_j}$ can be defined based on cc as $\mathrm{d_1} = 1 - \mathrm{cc}$ or $\mathrm{d_2} = (\frac{1-\mathrm{cc}}{1+\mathrm{cc}})^{\beta}$, $(\beta > 0)$.

Smart meter time-series data are high dimensional. It is not desirable to work with that kind of data in practice, Therefore, the feature-based clustering method for the phase grouping problem is adopted. As already explained, the PC method is used for drawing features from the voltage time-series data set.

Here, we consider the Euclidean distance function in the chosen principal component's space used as distance metric in the subsequent clustering process.

After getting familiar with the principal component analysis method for data reduction and the Euclidean distance function, it is time to introduce the algorithm for clustering of the voltage time-series data.

As it was done before for describing the PCA method using one representative day only, the same day is used again for the sake of simplicity for explaining the algorithm and the steps for the clustering are given below.

Step one: Normalize and center the output data from the PCA method

The data obtained from the PCA method, i.e. the first and second principal components are normalized and centred again by their standard deviation before starting with the k-means clustering algorithm.

After many different scenarios and implementation of different pre-processing functions, the algorithm in which the data given from the PCA method is centred and normalized and taken as an Input for the clustering algorithm showed the best behaviour. In figure 6 the new graphical representation of the data points i.e. the consumers are depicted.

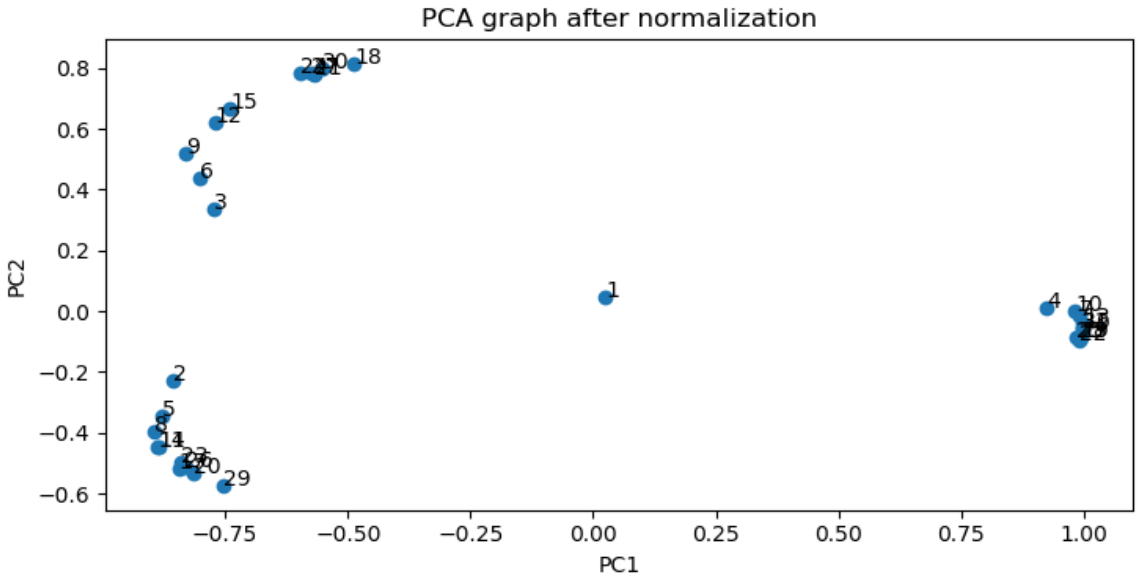


Figure 6. PCA graph after normalization

It can be seen from the figure that after centring and normalization of the data points, they are visually separated, forming a group. Moreover, it can be vividly seen that consumer number one is in the middle of the graph, so that is our potential outlier, since it is not joining any of the formed groups. From this point, the data is ready to be taken as an Input for step number two.

Step two: Determine the number of clusters in the data set

One of the crucial steps in the k-means clustering algorithms is determining the number of clusters in a data set. It is a quantity often labelled as $\mathbf{k}$ as in the k-means algorithm, which is a frequent problem in data clustering, and is a distinct issue from the process of solving the clustering problem.

The correct choice of $\mathbf{k}$ is often ambiguous, with interpretations depending on the shape and scale of the distribution of points in a data set and the desired clustering resolution of the user. In addition, increasing the number of clusters without penalty will always reduce the amount of error in the resulting clustering, to the extreme case of zero error if each data point is considered its own cluster (i.e. when $\mathbf{k}$ equals the number of data points, n). Intuitively then, the optimal choice of $\mathbf{k}$ will strike a balance between maximum compression of the data using a single cluster, and maximum accuracy by assigning each data point to its own cluster. If an appropriate value of $\mathbf{k}$ is not apparent from prior knowledge of the properties of the data set, it must be chosen somehow. There are several categories of methods for making this decision. Some of those are: Elbow method, X-means clustering, Information criterion approach, Silhouette method, Cross-validation, etc.

For the phase grouping algorithm, we already know that there are three phases: phase A, phase B and phase C to which the consumers fed by feeder one are connected, but for the sake of confirmation of the results, the elbow method is explained for graphically representing the number of clusters from the reduced voltage time-series data set in case we did not know the exact number of clusters.

The Elbow method

The Elbow method is a heuristic method of interpretation and validation of consistency within cluster analysis designed to help find the appropriate number of clusters in a dataset. This method

looks at the percentage of variance explained as a function of the number of clusters. The number of clusters is chosen in such a way that adding another cluster does not give much better modelling of the data.

More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. Therefore, the number of clusters is chosen at this point, hence the "elbow criterion" as shown in figure 7. This "elbow" cannot be unambiguously identified. Percentage of variance explained is the ratio of the between-group variance to the total variance.

In the figure below, the x-axis consists of number of possible clusters k, from 1-9, while on the y-axis the distortion is represented. It is another way for showing the variance with respect to the total variance, such that here the distortion is represented, which massively falls below 0.2 when the number of clusters reaches number equal to 3 (the elbow). For this reason, the optimal number of clusters obtained from the elbow method is three which confirms until this step the accuracy of the method using the voltage time-series data set.

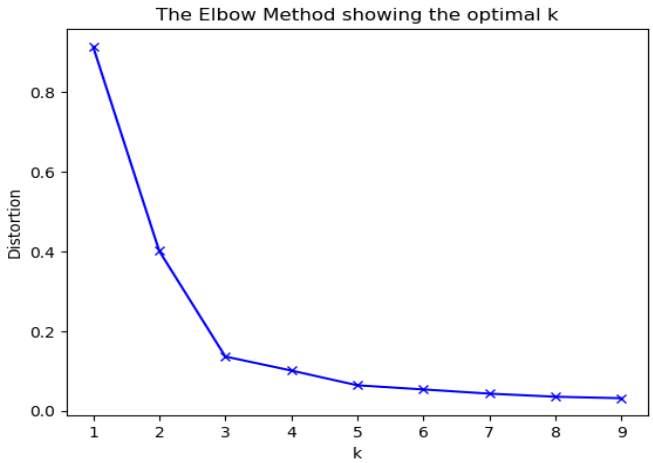


Figure 7. Elbow method defining the number of clusters

Step three: Assigning random centroids

After defining the number of clusters, $\mathrm{k} = 3$ which corresponds to the number of phases, the k-means algorithm continues with assigning three random centroids/ three random centres that are three different coordinates of three different data points i.e. three different consumers.

At this point, we still do not know whether those three different data points belong to a different group. It can happen that two out of three data points are part of the same group which should have a minor influence on the final result. Anyway, at this stage the algorithm takes as Input three different consumers and sets their coordinates to be centres of three different clusters.

When all the previously said is implemented on our representative day which is the 3rd day of January, after assigning three random centroids, we get the result shown in figure 8. As we can see from the figure below, there are three centroids defined in each group. The three centroids have bold red, green and yellow colour respectively. So, these are the initial centres from which the iterative method of defining the closest centroids for all group of data points/consumers begins.

Not all the data points in figure 8 are visible, due to the fact of their similar characteristics, which in this case is the same phase they are connected to. Some data points/consumers are overlapping, which is not a mistake.

With assigning of the centroids, the initialization of the k-means clustering algorithm begins. This is the starting point from which the Euclidean distance from each centroid to each data point is calculated and improved until convergence, explained in the next step.

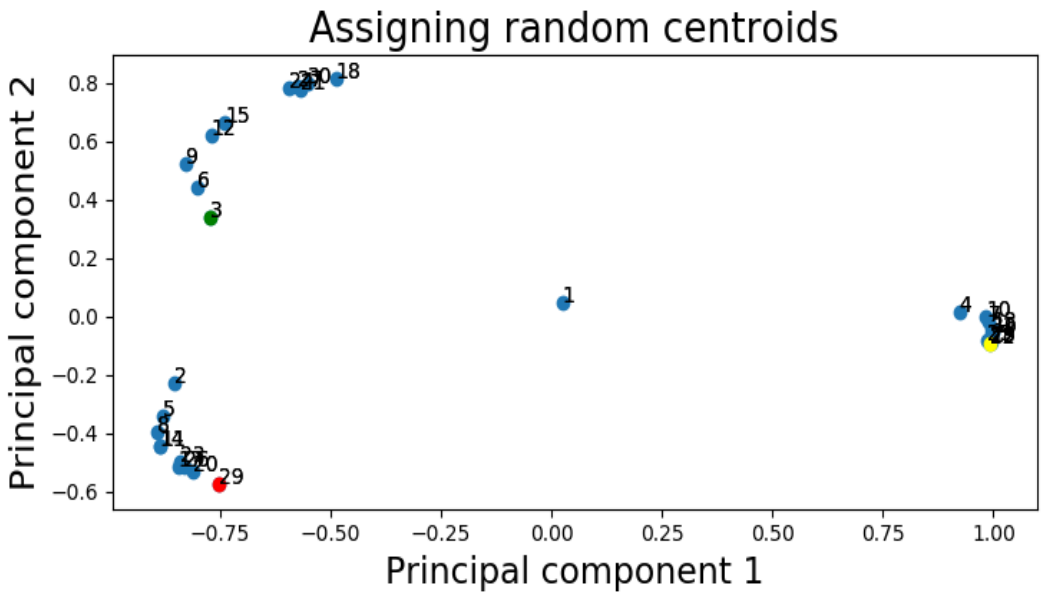


Figure 8. Assigning random centroids

Step four: Calculate each subset d_E's distance to each cluster

The distance is defined as the sum of squared distances of all the data points in the graph represented in figure 8 with the cluster centre. That is the Euclidean distance between all data points with the cluster centre, calculated according to the equation explained above. The data points have coordinates that correspond to the values obtained by the PCs method.

Step five: Assign each subset to the cluster that has the minimum summed distance

For each data point we have three calculations regarding the fact that there are three clusters with three centres. After calculating the distances from each data point to each of the three centres, the centre that is closest to the data point, i.e. the shortest summed distance between a consumer's data point and the centre defines the centre of that data point.

Step six: For each cluster Ci, update its center by averaging all the data points that have been assigned to it

After all the distances have been calculated, the next step in the k-means clustering algorithm is related to updating the centres. The centres are updated in such a way that the centre corresponds to the average value of all the data points that have been assigned to that centre.

In the figure below, it is graphically shown the update of the centres after the first iteration i.e. after the first assigning of the centroids and computation of the distances between each data point and each centre.

It can be seen from figure 9 that the change of the coordinates of the centres is significant for two of the groups, which is a result of the random allocation of the initial centroids, while for the third group (represented in yellow colour) is not very visible, since all of the data points are very close to each other and it is difficult to realize the change of the centroid.

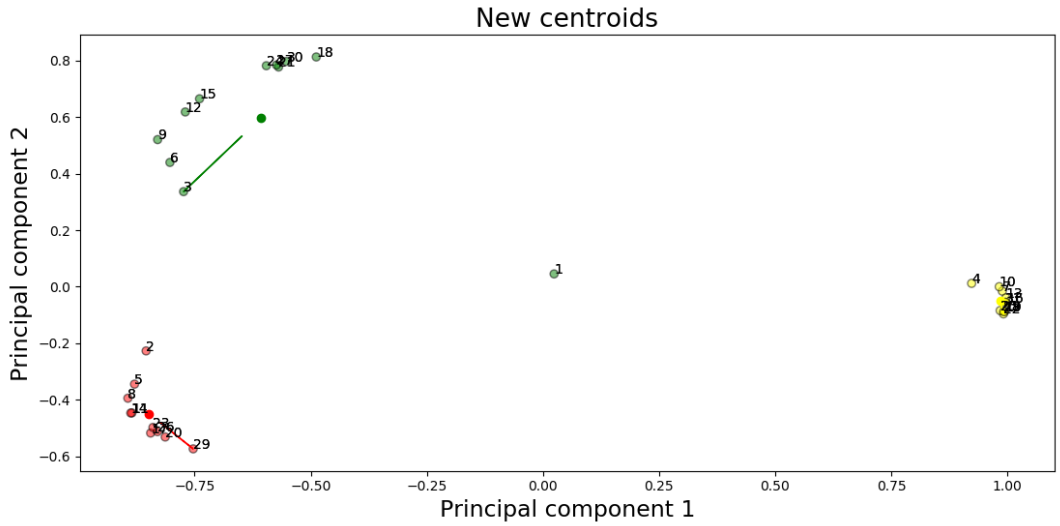


Figure 9. New centroids after first assignment

Step seven: Iterate between step (4) and (5) until convergence

The algorithm will perform as many iterations as needed until it converges. The convergence here corresponds to the result telling that there is no other centroid whose average distance between that centroid and all data points that belong to that cluster is smaller than the average distance between the current centroid and all the data points that are part of the cluster. In other words, the algorithm iterates, and the centroids are changing until the shortest distance is found between centroid an all data points.

Step eight: Return $\{\mathbf{C_1}, \mathbf{C_2}, \mathbf{C_3}\}$

In the end, the final centroids are defined and at this point each of the data points belong to one of the clusters whose centroids were just calculated. After the eight step the k-means algorithm stops with its iterations and gives a result in which all the data points are divided between the three clusters i.e. phase groups.

The result for the selected representative day is depicted in figure 10, where three groups i.e. phases can be realized with their centroids. Those are the final centroids for each of the groups and no changes are further done.

Figure 10 shows the result of the clustering algorithm after finding the final best centroids. The consumers that belong to the same group have the same number, which is the number of the group according to their shortest distance to the centroid of that particular group. After having the results and the final data frame, there is only one thing left to be done which is correlating the number of the groups with the three possible phases: A, B and C.

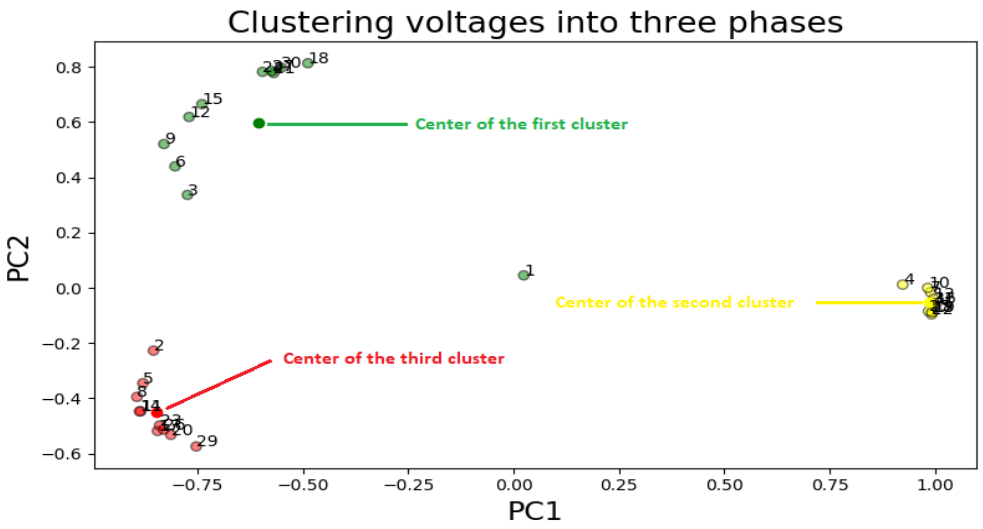


Figure 10. Clustering voltages into three groups/phases

From this stage it can be vividly seen from figure 10 that the customer number 1 is in the middle of the graph, so most probably it will have a wrong phase, i.e. it will be allocated to a wrong phase. Still, having in mind that all the other 30 customers are clearly part of a group can give us a gut feeling that the result from the algorithm implemented just for one day has high accuracy.

As previously said, once we have the three groups and all the data points allocated to those three groups (as depicted in figure 10), the final step for the phase grouping algorithm is to identify to which group each data point corresponds to. That identification is done by implementing the one-to-one matching algorithm explained below.

2.3.4 One-to-One Matching Algorithm

Once the customers are clustered as described in the previous section, the next and last step in order to define the phases of the customers is by using the one-to-one matching algorithm. Since the customers in the same cluster should have the same phase connection, the phase of each customer can be identified by taking small number of voltage time-series data as representative data and identify their phase connectivity. This is an enormous workload reduction compared with

performing phase identification algorithm on each one of the consumers. For that particular reason one-to-one matching algorithm is used between the set of clusters and the set of possible phase connections. For better understanding, in addition this algorithm will be explained firstly for general case and then it will be correlated with the example of the representative day.

The one-to-one matching algorithm can be found by solving the following minimization problem: Suppose there are $\mathbf{k}$ clusters to be identified with centroids i.e. centres C1, C2,..,Ck and there are $\mathbf{k}$ voltage time-series data points of the $\mathbf{k}$ possible phases, taken from the substation. The $\mathbf{k}$ substation voltage time-series are centred and normalized by their standard deviation and then projected onto the principal component's space used for clustering.

Let V1, V2,.., Vk be the coordinates of the three voltage time-series in the chosen PC's space and let $\mathbf{f}: \{\mathbf{C1}, \mathbf{C2}, \dots, \mathbf{Ck}\} \rightarrow \{\mathbf{V1}, \mathbf{V2}, \dots, \mathbf{Vk}\}$ be an unknown bijection between the cluster set and substation voltage set.

The solution of the following minimization:

$$\forall\ bijection\ f:\{\mathrm{C1,C2,C3}\}\rightarrow\{\mathrm{V1,V2,V3}\} \sum_{i=1}^{3} de\big(Ci, f(Ci)\big)^2$$

is the one-to-one matching for the phase grouping of customer's voltages. Where $\mathbf{de(Ci, f(Ci))^2}$ is the Euclidean distance between $\mathbf{Ci}$ and $\mathbf{f(Ci)}$.

After solving the one-to-one matching algorithm, we get the phases such that the phase of each cluster's paired voltage data is the cluster's identified phase. The minimization can be solved by exhaustive search, because there is only $\mathbf{k!}$ possible bijections where k is small.

Compared to the load matching approach in which the aggregated electricity consumption of all customers that match that of the substation is assumed, the one-to- one matching algorithm is less sensitive to the presence of unmetered customers.

Now if we implement the previously said for the case with one representative day the one-to-one matching algorithm followed step by step will lead us to the following results:

Since we have three clusters with three centres, for this algorithm only three voltage time-series data will be used from the three possible phases. Next, we take three substation voltage time-series data and normalize and centre them by their standard deviation and then project on the principal component space used for clustering.

So, we take three voltages from the three different phases and we try to find the bijection between the centroids and the data points corresponding to customer's voltage using the equation for the minimization.

After the PCA method applied to the three voltage series data, from three different phases: A, B, C their graphical representation is shown in figure 11. We can see that the phases have similar starting point as the first output of the principal component analysis shown in figure 5.

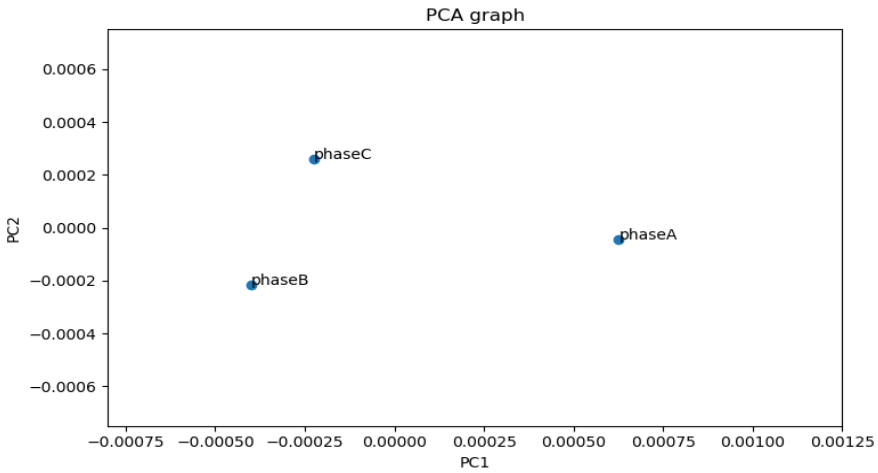


Figure 11. PCA graph of three representative voltages for the three phases

Again, after normalizing and centring the results obtained from the PCA graph, in figure 11, for the three representative voltages, the k-means algorithm for determining the best coordinates for the

centroids is implemented and compared to the results obtained with the k-means algorithm for the representative day.

According to figure 12, we obtain the final decision for the phase groups, which is:

1. All the voltage data points for the consumers that belong to group two i.e. the second cluster are connected to phase A (represented with yellow colour).

2. All the voltage data points for the consumers that belong to group three i.e. the third cluster are connected to phase B (represented with red colour).

3. All the voltage data points for the consumers that belong to group one i.e. the first cluster are connected to phase C (represented with green colour).

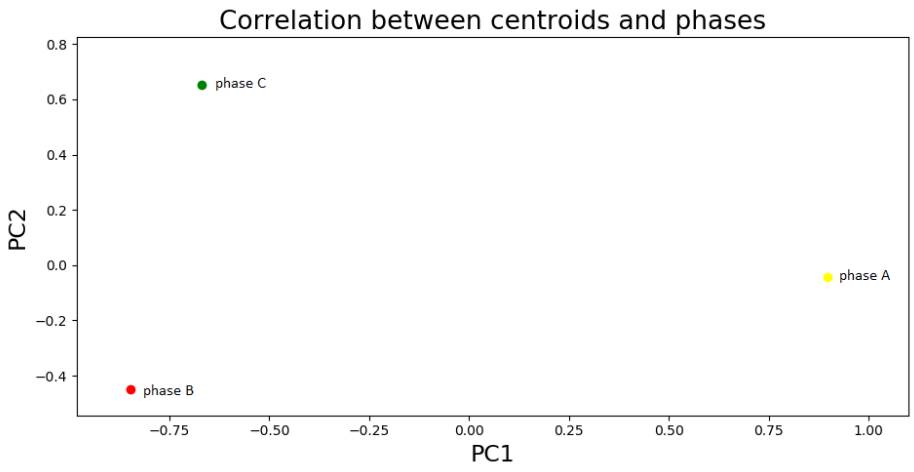


Figure 12. Correlation between centroids and phases

With the one-to-one matching algorithm, we wrap it up the phase grouping algorithm of customer's voltage time-series data. After the detailed explanation of all the methods and algorithms implemented for determining the phases of the customers for one day it is time to show the case studies done for validation of the accuracy of the method.

The reader at this stage might think that the detailed explanation is enough and that one day is good enough for phase identification. But, even though during the detailed explanation the results seem pretty good for the representative day used, which is not always the case. Moreover, this day was chosen on purpose, just for the sake of explanation for better understanding of the algorithms.

However, we will see from the case study that one day cannot be enough in order to determine the phase to which the customers are connected to. Instead, different simulation scenarios are done per week and two weeks' time considering the whole day (midnight until 11:30 PM), high-demand hours (6PM – 10PM) and low-demand hours (11PM – 8AM) for validation of the performance of the algorithm.

2.3.5 Software Tools

The main software tools used in this thesis are the following:

1. OpenDSS

OpenDSS is an open source distribution system simulator that has been developed by electric power research institute (EPRI), USA [7]. It is a multi-purpose software and different studies can be conducted, like DER planning, harmonic studies, voltage studies. OpenDSS was used for obtaining the voltage profiles of the customers (V) corresponding to the data extracted from the smart meters.

2. Python

Python is an open source programming language that has higher level of abstraction and simplicity [8]. The great advantage of Python is the also open source libraries, which are available and has a great number of applications, such as Keras.Scikit-learn, etc. Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation [9]. Scikit-learn is a free software ML

library for the Python programming language.[10] It features various classification, regression and clustering algorithms including support vector machines and k-means.

2.3.6 Summary of Methodology

The proposed phase grouping algorithm consists of: a feature reduction method PCA that extracts the main features of the voltage time-series data, an unconstrained k-means clustering algorithm that sorts those features in three groups according to their similarities, and a one-to-one matching algorithm that determines the most likely phase group for each customer.

The algorithm is performed using the programming language Python with previously calculating the voltage time-series data by running a power flow simulation in the OpenDSS software for the Australian LV distribution network, such that firstly an interface between Python and OpenDSS was made in order to run a power flow simulation. Then, in another script the algorithm for defining the phases of the customers was coded using all the steps previously explained. The voltage time series data is the only information used in defining the topology of the residential network.

2.4 Case Study

2.4.1 Australian LV Feeder

After the detailed explanation of the phase identification algorithm, the case studies are represented below. The case studies follow the algorithm and the same exact steps for determining the phases of the consumers are included here. The difference, moreover, is the amount of data used. Previously we had one representative day in summer, whereas now for the PCA and k-means algorithm time duration of one week and two consecutive weeks from each season are taken for investigation. This section consider the case study for an Australian LV feeder as part of a LV distribution network.

Voltage data and measurements

This part of the Thesis comprises the low voltage network models used for obtaining the voltage time-series data, which are used as an input for the phase grouping algorithm. The network model and profile used in this section were developed on the basis of the LV Network data provided by AusNet (Australian energy company) that owns and operates the Victorian electricity transmission network and it is one of five electricity distribution networks and one of three gas distribution networks in Victoria, Australia.

The main objective of this part is to gain understanding of the characteristics and behaviour of the low voltage distribution network(s), analysing the networks to hosting capacity of PV systems, studying the penetration levels for which voltage rise and/or asset overloading occurs.

To achieve this objective, realistic profiles and feeder models were used during this study. From the AusNet LV Network, that consists of 4 three-phase 500 kVA Transformers 22/0.433 kV/kV, to simplify analysis, Transformer 1 is chosen. The analyses are done using The Open Distribution

System Simulator (OpenDSS) developed by EPRI. Python is used to run the time-series and snapshot analysis and plotting the results.

The preliminary studies were conducted on feeder 1 fed by Transformer 1. The LV network consists of 4 Transformers that feed 18 feeders, and a total of 465 customers. The network configuration is shown in Figure 13, where feeder 1 is the one that feeds the loads that are represented as triangles (△). The characteristics of feeder 1 are provided in Table 1.

The load profiles used were obtained from a pool of different load profiles from the Smart Meter Data provided with time resolution of 30 minutes. The data used for the PV profiles was obtained from the solar profiles data for 24 hours with 30 min resolution for Australia in year 2014. These profiles are created for the same sun irradiance and same size of PV units.

To allow to understand the effects related to the integration of PV systems in distribution grids, the analysis is firstly undertaken on a feeder of LV network without any PV penetrations. Then, PV systems are added with 50% and 100% random progressive penetration. All PV panels share the same irradiance profile given the small geographical area. The size of the PV panels is fixed with 4.5 kWp (kilowatt-peak).

Table 1: Characteristics of feeder 1

No. of Customers	No. of Lines	Substation transformer	Frequency
31	24	22/0.433 kV/kV, 3 phases, 500kVA(△Y)	50Hz

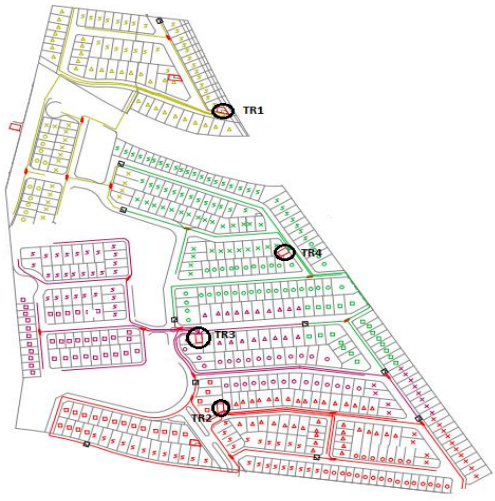


Figure 13. LV Network configuration

The figures below are showing the behaviour of the voltage where there are no PV systems and when there is 50% and 100% of PV penetration. One day is randomly chosen out of the whole year for representing the voltage profile of the consumers. In this case, summer day in January is chosen. As it can be seen from the figures for 50% and 100% PV penetration, there is a rise of the voltage during the day in the period of 10 AM untill 6 PM.

From the pool of Load profiles thirty-one load profiles are randomly chosen and allocated to all the customers fed by feeder 1 (same load profiles are used for the whole year). Then one day in summer – 3rd of January is chosen in order to show the behaviour of the feeder due to absence and presence of PV systems.

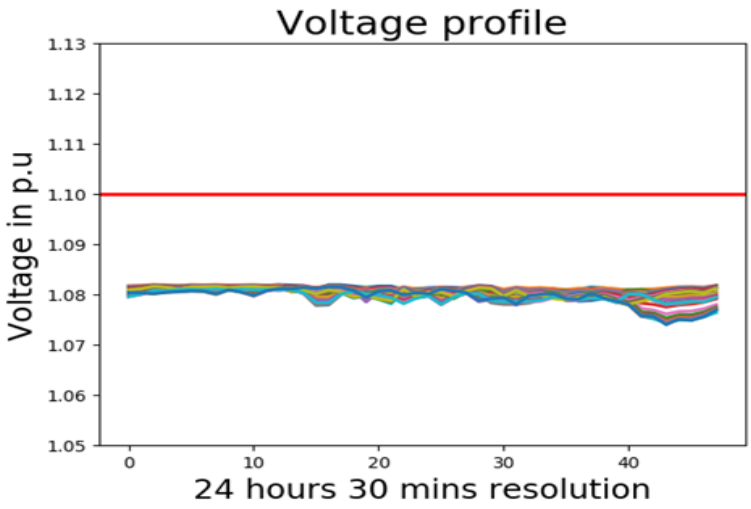


Figure 14.Voltage profile without PV penetration

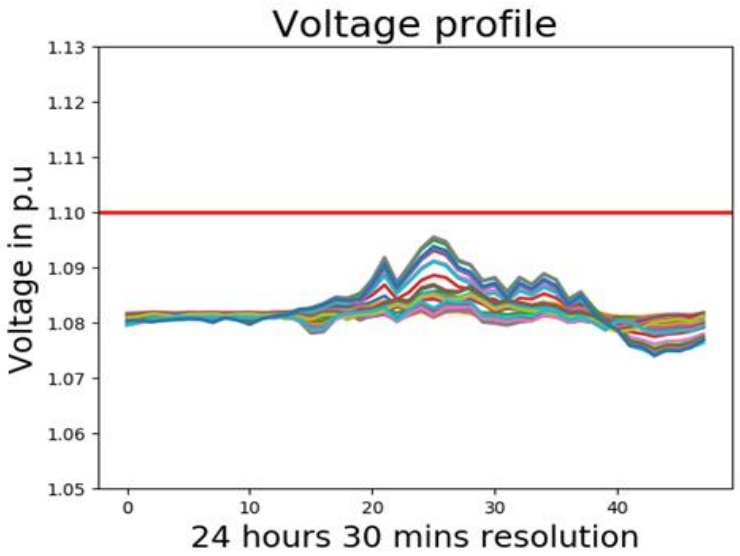


Figure 15.Voltage profile with 50% PV penetration

The results presented in Figure 14 show that the LV Network initially has no voltage issues and the substation transformer is in operation within its limits. Increasing the PV penetration to 50% results in an increase in the voltage, however, there is no issue as it can be seen from Figure 15.

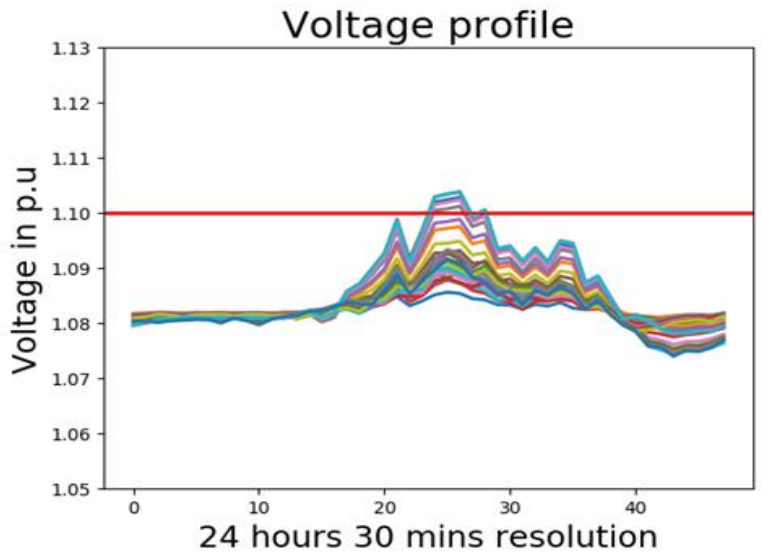


Figure 16.Voltage profile with 100% PV penetration

If the penetration is increased to 100%, as it can be seen from Figure 16, the voltage for some customers goes beyond the standard limit which is 1.10 p.u. due to peak photovoltaic generation.

From the figures above, it can be realized that the voltage for the consumers fed by feeder one has a starting value close to/equal to 1.08 p.u. which is a high value for starting point of the voltage without any photovoltaics. This is the current situation and for this reason, currently the distribution network system providers in Australia are focusing on reducing the starting value of the voltage such that an increment of hosting capacity and reduction of the reverse power flow can be performed.

Principal Component Analysis method

After getting familiar with the LV distribution network used in the case study, the voltage time-series data taken as an Input for the phase identification algorithm, as mentioned before, considers time duration of seven days and fourteen days in each of the four seasons. Since the

network topology is in Australia, the seasons are different than Europe, so at the beginning of the year firstly it is the summer season, followed by autumn, spring and winter.

Three different scenarios for one week/two weeks duration are done, taking different periods of the day, including all the forty-eight readings per day. The performance of the algorithm with and without a reduced amount of data is also represented. Therefore, it is later shown that taking specific time periods of the day reduces the computational time of the algorithm and gives better results, i.e. it is more accurate, which is an advantage related to other algorithms where not only was used the whole day, but also one to three months of data was taken for performing the calculations.[reference paper]. The work done in this thesis reduces significantly the amount of data by taking only one week or two weeks voltage time-series data.

The phase grouping algorithm starts such that firstly OpenDSS integrated with Python has been used to solve the power flow and the voltage profiles. Then the voltage profile for all the days necessary for the simulation are read. Then, according to the requirement, whether is whole day or time period of the day, the data set transforms its shape. The new version of the voltage time-series data set is then normalized and centred and proceed as Input to the PC analysis method.

Here, for graphical representation the one-week scenario during low demand hours will be represented. Then later the results will be shown in a table for all the three different time durations for both scenarios. After performing the PCA method for one week in January (summertime) the PCA graphs for each day of the week are shown below.

It can be seen from figures 17 – 23, that all the data points that represent the consumers are spread along the PCA graphs for each day. Moreover, it can be vividly seen that after performing the PCA method, all the data points are forming three different groups, such that in some days the data points are clearly belonging to a specific group, while in some days they are spread in the middle so it is a bit difficult to decide which data point belongs to which group.

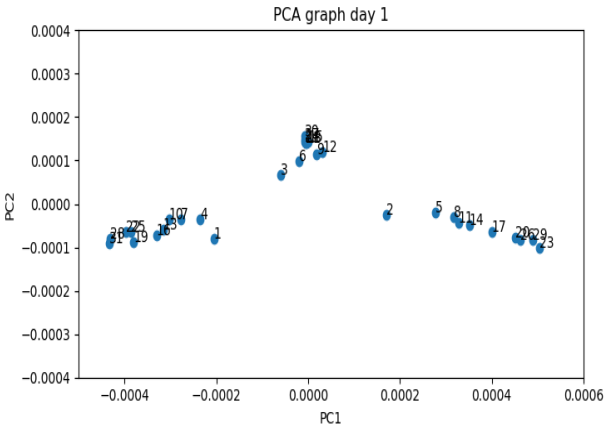


Figure 17. PCA graph day_1

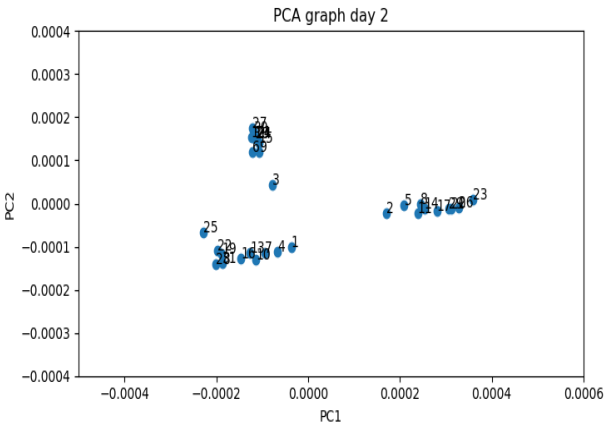


Figure 18. PCA graph day_2

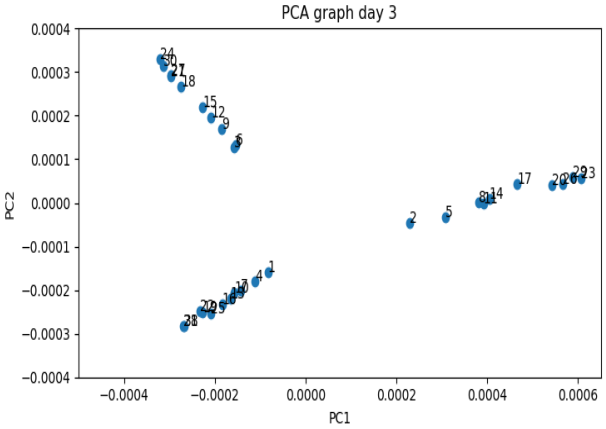


Figure 19. PCA graph day_3

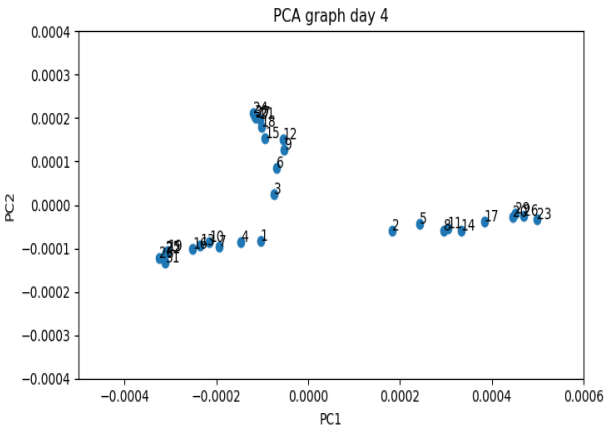


Figure 20. PCA graph day_4

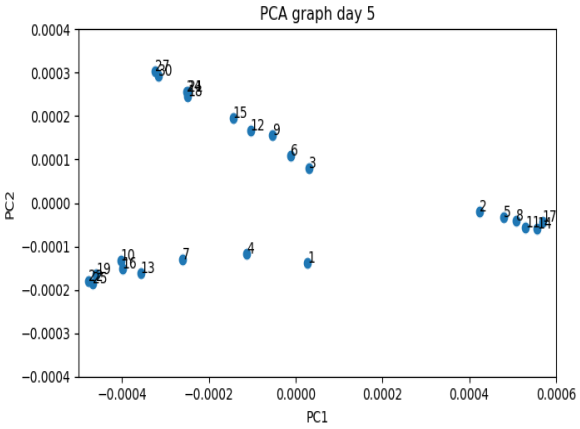


Figure 21. PCA graph day_ 5

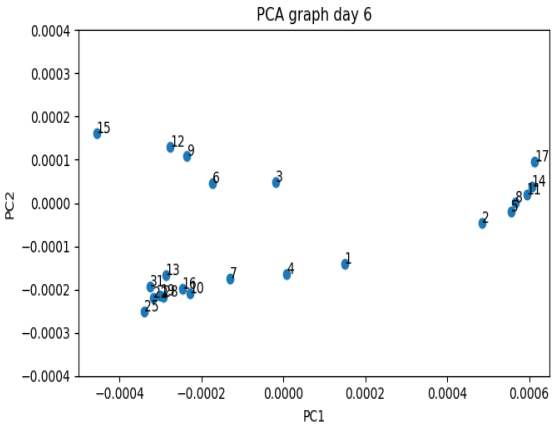


Figure 22. PCA graph day_6

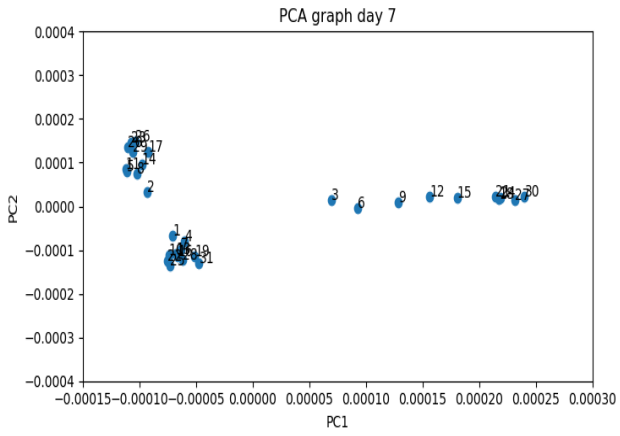


Figure 23. PCA graph day_7

After getting the results from the PCA algorithm, the data set is again centred and normalized so the range of the values of the coordinates of the data points that represent the consumers will be sorted according to their standard deviation.

K-means Clustering

The next step is to perform the k-means clustering algorithm of the output derived from the PCA method.

In this work the unsupervised k-means algorithm was implemented, regarding two different cases:

- Case 1: takes the same starting centroids for each day of the week and two weeks
- Case 2: takes updated starting centroid for each following day

The first case, as mentioned above, takes the same initial centroids for every day of the week. This is done for the purposes of labelling of the data points i.e. the consumers. That is why, at the beginning of the k-means clustering algorithm three random values are defined that correspond

to three different consumers. The coordinates of the three centroids are exactly equal to the coordinates of the those three randomly chosen consumers.

The second case, instead, updates the centroids for the following day. Here, for the first day the coordinates of the centroids are equal to the coordinates of three randomly chosen consumers, but starting from the second day and every other, the final coordinates of the centroids of the previous day are taken as starting coordinates for the next day.

In both cases explained above, the initial values for the centroids have a huge role in defining the phases of the customers. The better the initial allocation of the centroids for the first day, the faster convergence of the algorithm and its better performance. There are cases where accidentally the three values for the initial centroids belong to the same group, so the result of the algorithm is redundant, since instead of three groups sometimes only two groups are formed in the end with two centroids belonging to the same group.

That is why, in order to exclude the cases in which the randomization of the centroids will influence the result of the k-means clustering, the algorithm has been performed 30 times for the same week and the best results are taken into consideration i.e. the results in which it is vividly represented that there are three different groups.

Here, the case with updated centroids is represented, i.e. the case in which only the first day takes random centroids as Input to the algorithm while for every following day the centroids are already determined i.e. they are equal to the final centroids of the previous day. The different k-means clustering algorithm results for the whole week is depicted in figures 24-30.

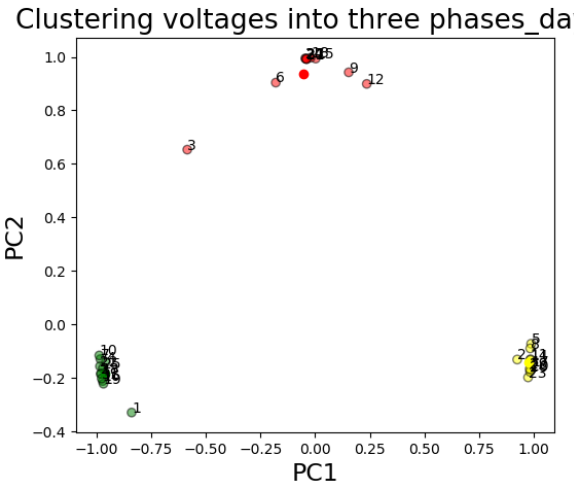


Figure 24. K-means day_1

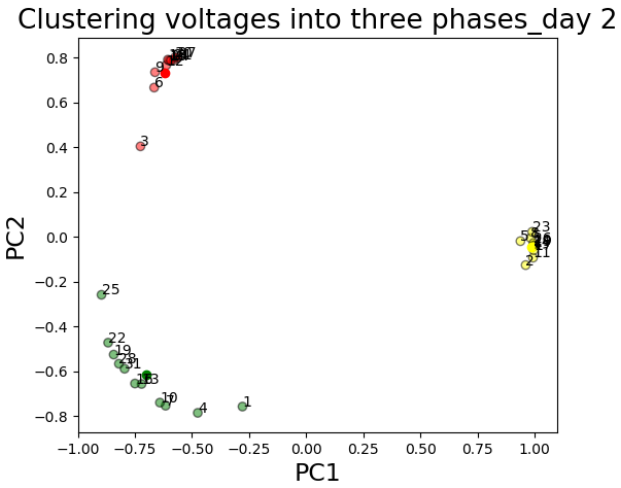


Figure 25: K-means day_2

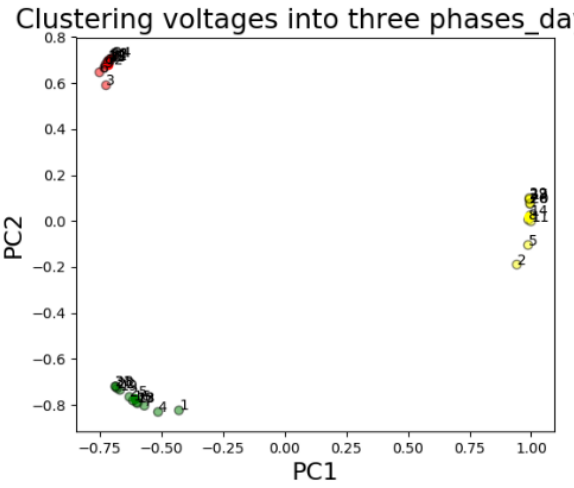


Figure 26. K-means day_3

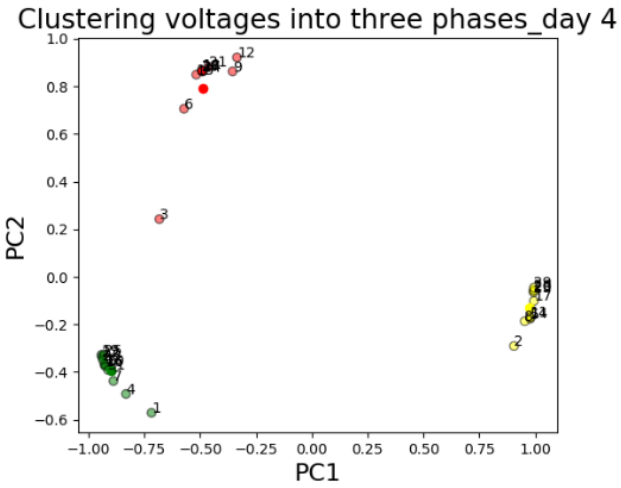


Figure 27. K-means day_4

It can be seen from the figures above that each data point belongs to a group and they are clearly noticeable, labelled with three colours. At this point, we still do not know to which phase each of the colours corresponds to. That is why, one-to-one matching algorithm, that was thoroughly explained above, needs to be implemented.

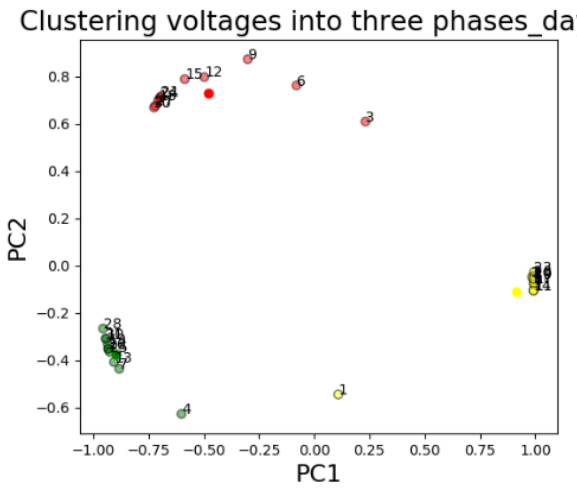


Figure 28. K-means day_5

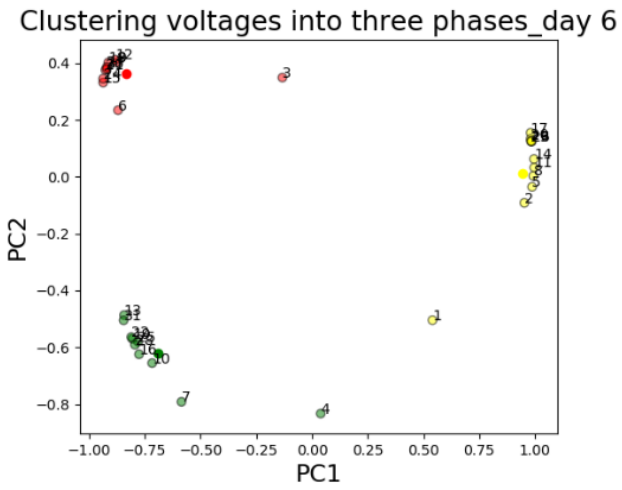


Figure 29. K-means day_6

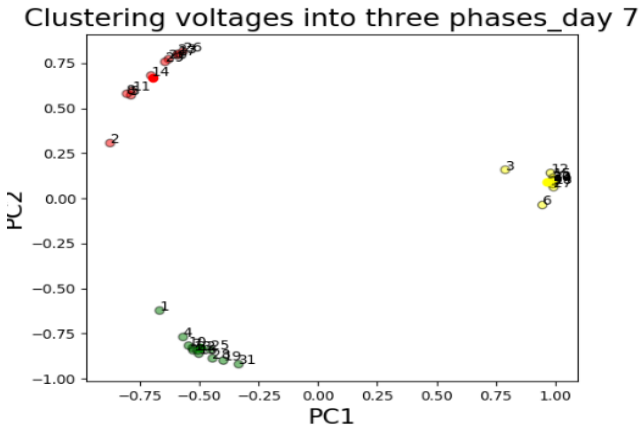


Figure 30. K-means day_7

One-to-one matching algorithm

Since we are performing the phase identification algorithm for the whole week, in order to determine which phase corresponds to which of the derived groups from the k-means clustering algorithm, the matching algorithm is performed regarding the following steps:

1. One day of the week is randomly chosen as representative day

2. Three voltage time-series data with reduced dimension, since we are working with voltage time-series data in low demand hours, from three different consumers connected to the substation are taken and the matching algorithm is performed
3. In the end we get the phases, such that each colour represents a phase

After randomly choosing a day for the matching algorithm, we get the result depicted in figure 36, where the three phases, now given with numbers, are clearly noticeable. Phase A, B, C correspond to the numbers: 1, 2, 3 that correspond to the colours: green, yellow, red.

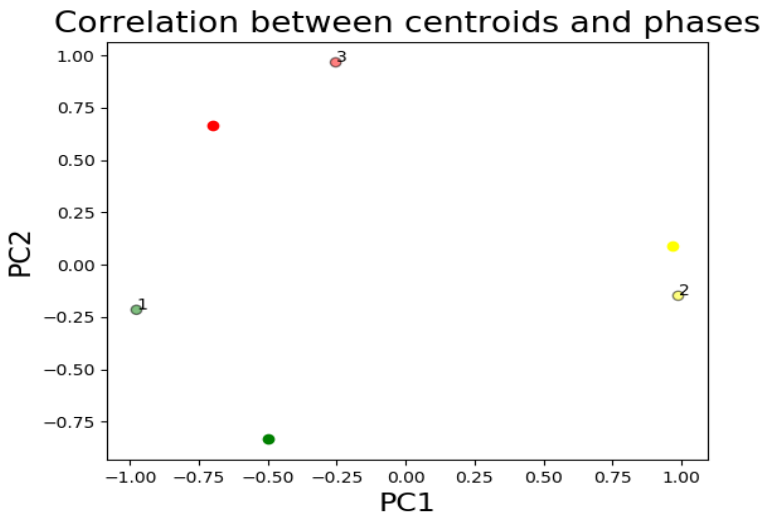


Figure 31. Matching algorithm results

Finally, after having the correlation between the groups and the phases the final step for determining the customer's phase is by selecting the most dominant group per customer during the representative week.

Depending on the voltage time-series data, after the performed phase identification algorithm, one consumer can belong to different groups in different days. The group in which one consumer is allocated in most of the days is taken as its group. This is represented in figure 32.

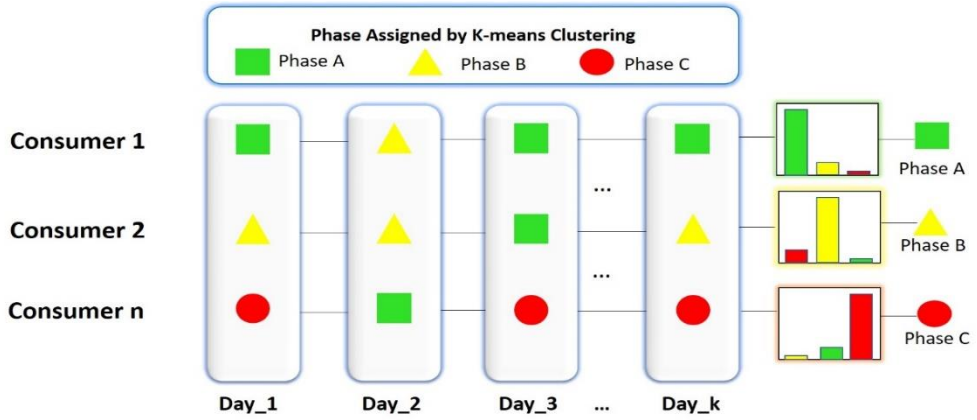


Figure 32. Assigning phases of consumers

Figure 32 represents the final method for assigning the phases of the consumers after the k-means clustering and one-to-one matching algorithm were performed. In the figure a general case is observed where the number of consumers is in the range of one to $\mathbf{n}$, while the number of days is in the range of one to $\mathbf{k}$. In this case we have thirty-one consumers and 7 days. The most dominant phase per day for one consumer defines its phase.

If we have a closer look to consumer one, from the figure above , we can see that in the second day this consumer is allocated to phase B, while in rest of the days of the representative week/weeks it is allocated to phase A, so as a result of the assigning of the phases it is determined that consumer one is connected to phase A.

In most of the cases, all the consumers fed by the same feeder are allocated to the right phase, but as mentioned before, depending on the voltage time-series readings of the day we calculated different results.

Results

The results for the phase identification algorithm performed per week and two weeks' time period for all the seasons are presented in Table 2. The table shows the number of consumers allocated

to the right phase. Since there are thirty-one consumers fed by feeder one, it can be seen that if we reduce the dimension of the voltage time-series data and we take only the voltage readings for the high demand hours or low demand hours, then the algorithm performs extremely well, such that all of the consumers are allocated to the right phase (thirty-one out of thirty-one). In other words, if we exclude certain periods of the day, the phase grouping algorithm of customer's voltages has very high accuracy equal to 100%.

On the other hand, taking into consideration the whole day, the algorithm has also very accurate results, with one mismatch i.e. one consumer out of thirty-one is allocated to the wrong phase which decreases the accuracy to 96.7% which is still very high accuracy.

Table 2. Number of consumers allocated to the right phase

Time period	Whole day (12AM-11:30 PM)	High demand hours (6PM – 10 PM)	Low demand hours (11PM – 8AM)
Week in summer	31/31	31/31	31/31
Week in autumn	30/31	31/31	31/31
Week in winter	31/31	31/31	31/31
Week in spring	30/31	31/31	31/31
Two weeks in summer	31/31	31/31	31/31
Two weeks in autumn	30/31	31/31	31/31
Two weeks in winter	31/31	31/31	31/31
Two weeks in spring	30/31	31/31	31/31

The results presented in Table 2 are in case when there is zero PV penetration in the low voltage feeder. The same analysis were performed with 50% and 100% PV penetration and the same results were obtained, which gives us the conclusion that with introduction of PV penetration to the low voltage feeder, there is no change in the results for the phases of the consumers. The phase identification algorithm is not influenced by the PV penetration on the low voltage feeder.

2.4.2 Australian LV Network

In this part of the Thesis we want to show the performance of the phase grouping algorithm if the take into consideration all feeders of the LV Distribution Network that are connected to TR1.

Previously, we saw the case where the phase grouping algorithm was performed for feeder 1 connected i.e. fed by transformer 1, whereas now we want to investigate the accuracy of the algorithm in case where all the feeders are used, i.e. the voltage time-series data of all the consumers fed by feeders 1-4 are taken as Input for the phase grouping algorithm. The steps for obtaining the phases of the consumers for all the feeders fed by transformer one are explained below.

Voltage data and measurements

This part of the Thesis comprises the low voltage network model used for obtaining the voltage time-series data, which are used as an Input for the phase grouping algorithm. The network model and profile used in this section were developed on the basis of the LV Network data provided by AusNet (Australian energy company), same as the previous case.

The main objective of this part is to gain understanding of the characteristics and behaviour of the low voltage distribution network, now with all four feeders included, analysing the networks to hosting capacity of PV systems, studying the penetration levels for which voltage rise and/or asset overloading occurs.

To achieve this objective, the same realistic profiles and feeder models as explained before were used during this study. The main difference is that the preliminary studies were conducted on feeders 1-4 fed by Transformer 1. The network configuration only for the feeders fed by TR1 is shown in Figure 33, which is extracted from the LV network configuration in figure 14. The characteristics of all the feeders are provided in Table 3.

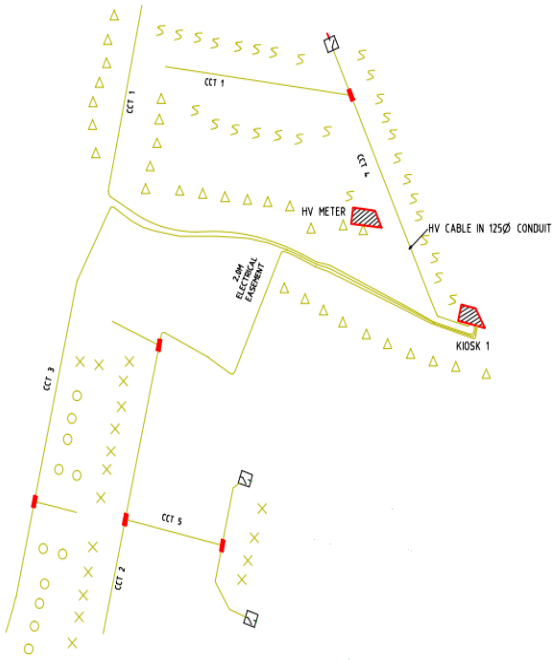


Figure 33. LV Network Scheme for all feeders fed by TR1

Table 3. Characteristics of the feeders fed by TR1

Feeder Number:	No. of Customers	No. of Lines	Substation transformer	Frequency
1	31	24	22/0.433 kV/kV, 3phases,500kVA(△Y)	50Hz
2	18	21	22/0.433 kV/kV, 3phases,500kVA(△Y)	50Hz
3	11	11	22/0.433 kV/kV, 3phases,500kVA(△Y)	50Hz
4	29	22	22/0.433 kV/kV, 3phases,500kVA(△Y)	50Hz

The same load profiles as before were obtained from a pool of different load profiles from the Smart Meter Data provided with time resolution of 30 minutes. Also, the data used for the PV profiles was obtained from the solar profiles data for 24 hours with 30 min resolution for Australia in year 2014. These profiles are created for the same sun irradiance and same size of PV units.

To allow to understand the effects related to the integration of PV systems in distribution grids, the analysis is firstly undertaken on the feeders of LV network without any PV penetrations. Then, PV systems are added with 50% and 100% random progressive penetration. All PV panels share the same irradiance profile given the small geographical area. The size of the PV panels is fixed with 4.5 kWp.

The figures below are showing the behaviour of the voltage where there are no PV systems and when there is 50% and 100% of PV penetration. One day is randomly chosen out of the whole year for representing the voltage profile of the consumers. In this case, again one summer day in January is chosen. As it can be seen from the figures for 50% and 100% PV penetration, there is a rise of the voltage during the day in the period of 10 AM until 6 PM.

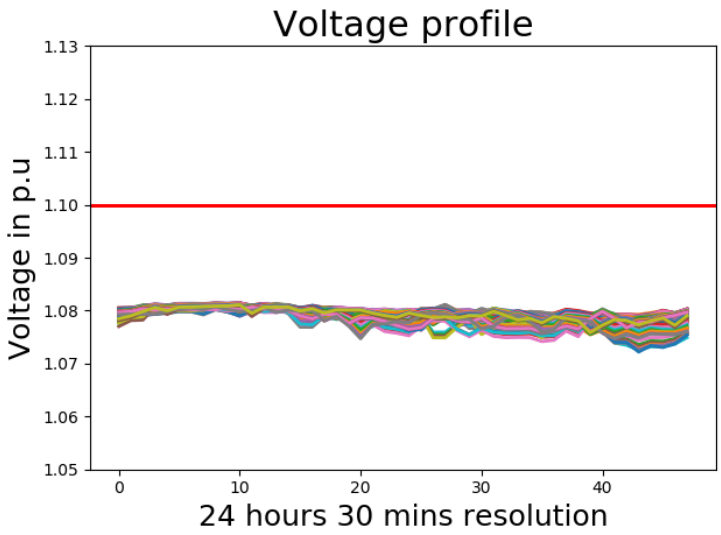


Figure 34. Voltage profile of all consumers fed by TR1 with no PV

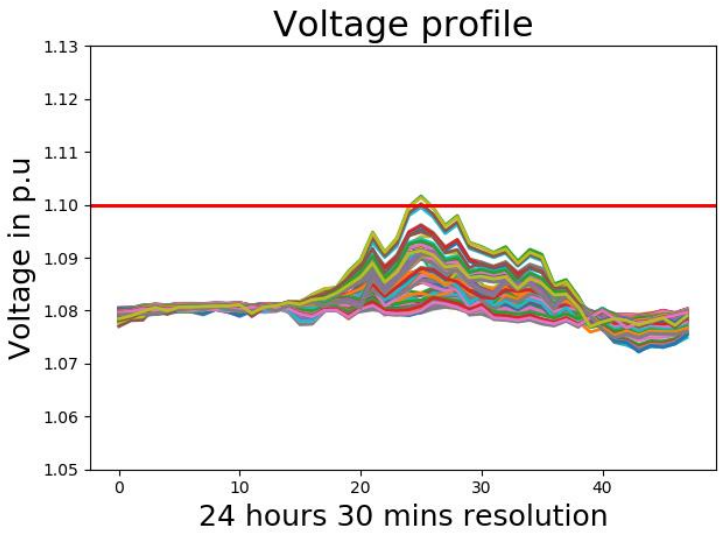


Figure 35. Voltage profile of all consumers fed by TR1 with 50% PV penetration

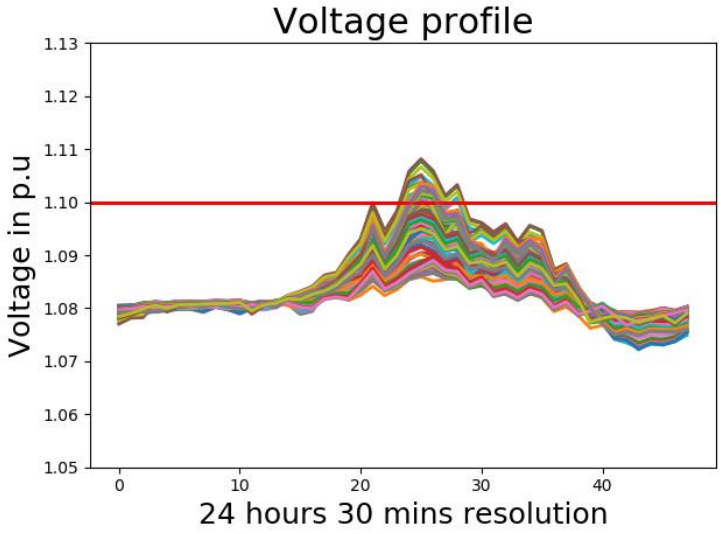


Figure 36.Voltage profile of all consumers fed by TR1 with 100% PV penetration

From the pool of Load profiles 89 load profiles are randomly chosen and allocated to all the customers fed by feeders 1-4 (same load profiles are used for the whole year). Then one day in summer – 3rd of January is chosen in order to show the behaviour of the feeders due to absence and presence of PV systems.

The results presented in Figure 36 show that the LV Network initially has no voltage issues and the substation transformer is in operation within its limits. Increasing the PV penetration to 50% results in an increase in the voltage. Opposite of what we saw in the previous case, now there is a voltage violation in the case of 50% PV penetration for some consumers, when all the feeders fed by TR1 are included, as it can be seen from Figure 37. If the penetration is increased to 100%, as it can be seen from Figure 38, the voltage for some customers goes beyond the standard limit which is 1.10 p.u. due to peak photovoltaic generation.

Again, from the figures above, it can be realized that the voltage for the consumers fed by feeders one – four has a starting value close to/equal to 1.08 p.u., as it was in the previous case, since the same load profiles and solar profiles are used for the now augmented network.

Input data

After getting familiar with the LV distribution network used in this subsection of the case study, the voltage time-series data taken as an Input for the phase identification algorithm, as mentioned before, considers time duration of seven days and fourteen days in each of the four seasons. (the same case is observed as before, now for four feeders instead of one).

As shown before, here again three different scenarios for one week/two weeks duration are done, taking different periods of the day, including all the forty-eight readings per day. The performance of the algorithm with and without reduced amount of data is also represented. Therefore, it is later shown that taking specific time periods of the day reduces the computational time of the algorithm and gives better results, i.e. it is more accurate.

The algorithm is implemented in the case with no PVs included, since there is no difference in the results. The work done in this Thesis reduces significantly the amount of data by taking only one week or two weeks voltage time-series data. The phase grouping algorithm starts such that firstly OpenDSS integrated with Python has been used to solve the power flow and the voltage profiles.

Then the voltage profile for all the days necessary for the simulation are read. Then, according to the requirement, whether is whole day or time period of the day, the data set transforms its shape. The new version of the voltage time-series data set is then normalized and centred and proceed as Input to the PC analysis method. Here, for graphical representation the one-week scenario during low demand hours will be represented. Then later the results will be shown in a table for all the three different time durations for both scenarios.

Principal Component Analysis Method

After performing the PCA method for one week in January (summertime) the PCA graphs for each day of the week are shown below:

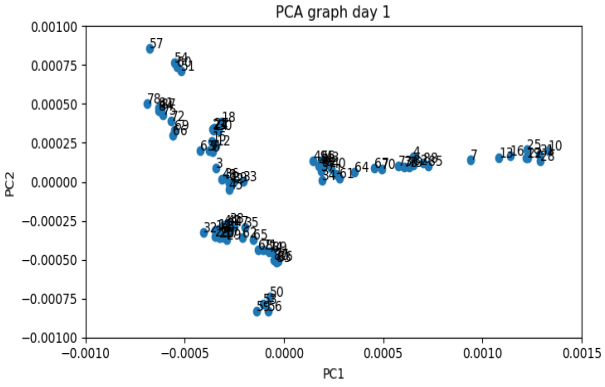


Figure 37: PCA_day_1

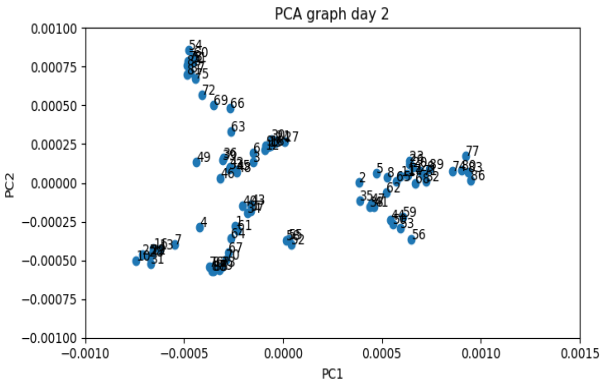


Figure 38: PCA_day_2

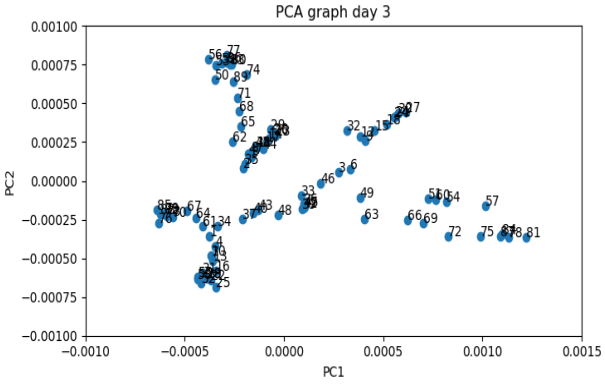


Figure 39: PCA_day_3

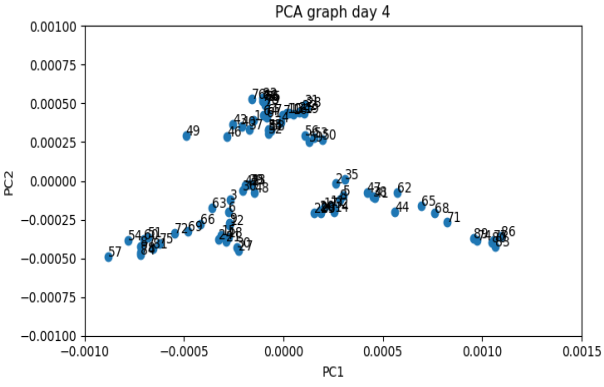
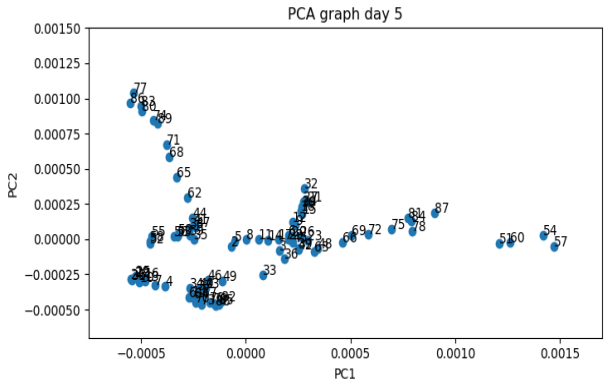


Figure 40: PCA_day_4



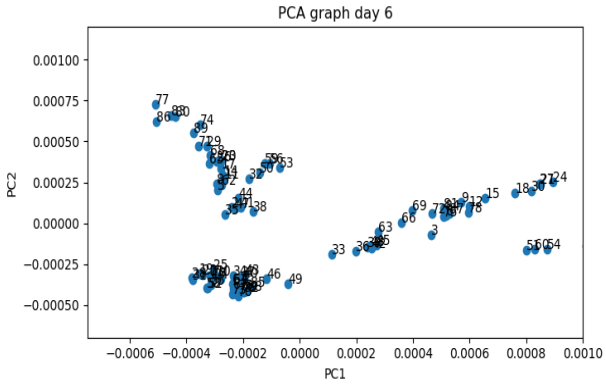


Figure 41: PCA_day_5 Figure 42: PCA_day_6

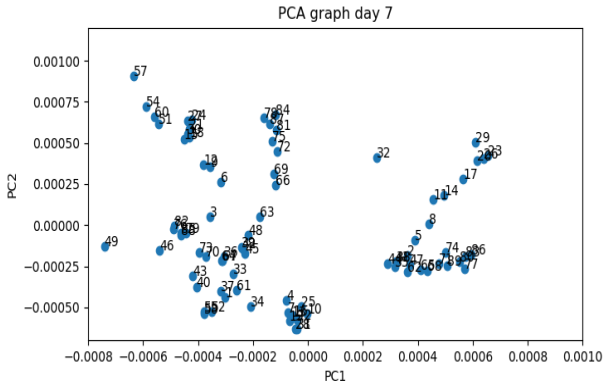


Figure 43: PCA_day_7

It can be seen from figures 37 – 43, that all the data points that represent the consumers are spread along the PCA graphs for each day. Having in mind the previous case and the results obtained from the PCA method, in this case it is difficult to notice three different groups after performing the method.

As explained before, the PCA method is used to reduce the amount of data, keeping the most of information of the data points in two new principal components.

In cases where the number of consumers is big, as this one, there is no clearly visual representation of three different groups.

After getting the results from the PCA algorithm, the data set is again centred and normalized so the range of the values of the coordinates of the data points that represent the consumers will be sorted according to their standard deviation.

The next step is to perform the k-means clustering algorithm of the output derived from the PCA method.

K-means Clustering

In this work again the unsupervised k-means algorithm was implemented, regarding two different cases, (same as when we had only feeder 1):

Case 1: takes the same starting centroids for each day of the week and two weeks
Case 2: takes updated starting centroid for each following day

In both cases explained above, the initial values for the centroids have a huge role in defining the phases of the customers. The better the initial allocation of the centroids for the first day, the faster convergence of the algorithm and its better performance.

That is why, in order to exclude the cases in which the randomization of the centroids will influence the result of the k-means clustering, the algorithm has been performed 50 times (20 times more than the first case with feeder one) for the same week and the best results are taken into consideration i.e. the results in which it is vividly represented that there are three different groups.

Here again, the case with updated centroids is represented, i.e. the case in which only the first day takes random centroids as Input to the algorithm while for every following day the centroids are already determined i.e. they are equal to the final centroids of the previous day. The different k-means clustering algorithm results for the whole week is depicted in figures 44-50.

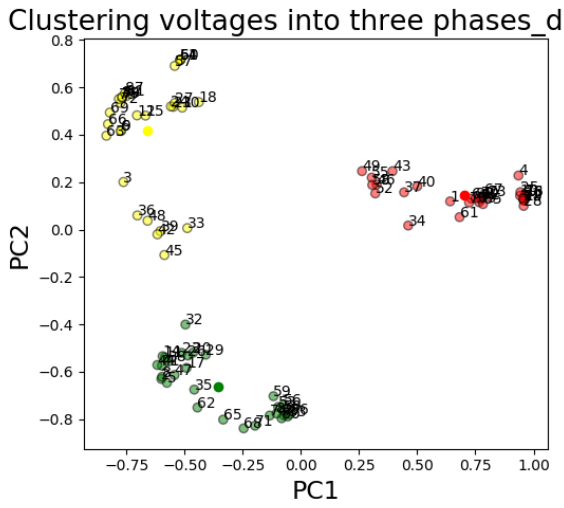


Figure 44: K-means day_1

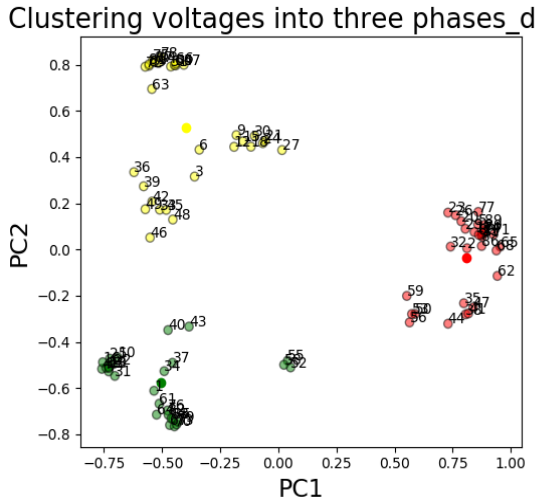


Figure 45: K-means day_2

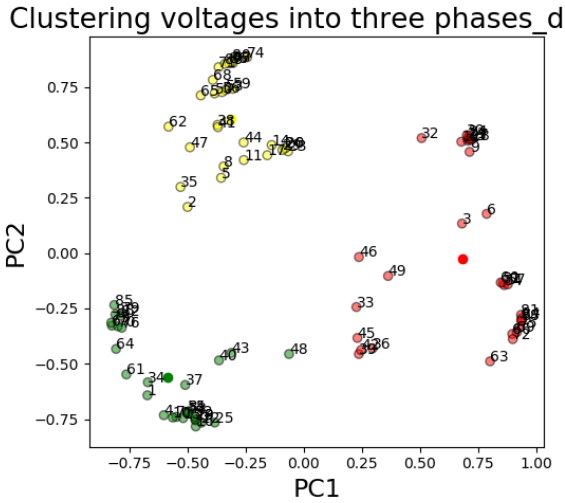


Figure 46: K-means day_3

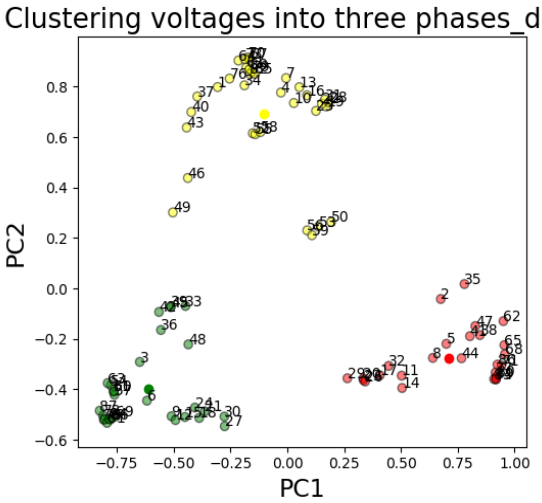


Figure 47: K-means day_4

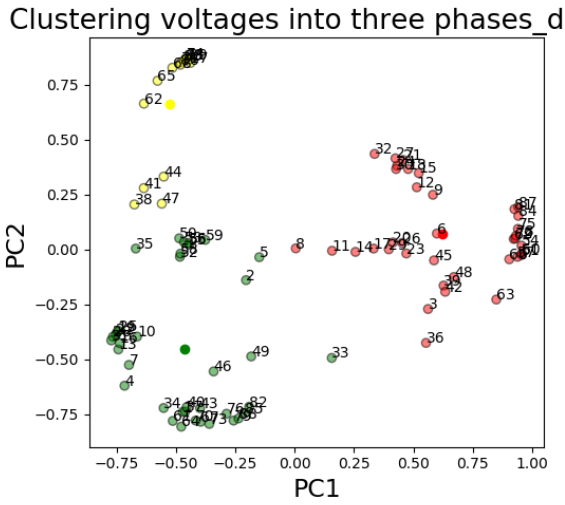


Figure 48: K-means day_5

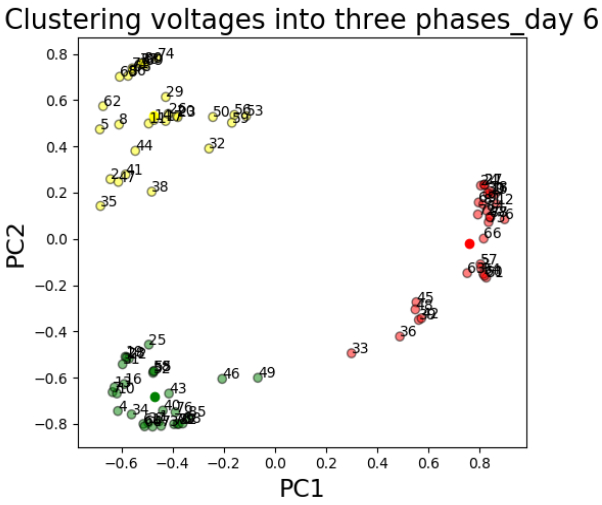


Figure 49: K-means day_6

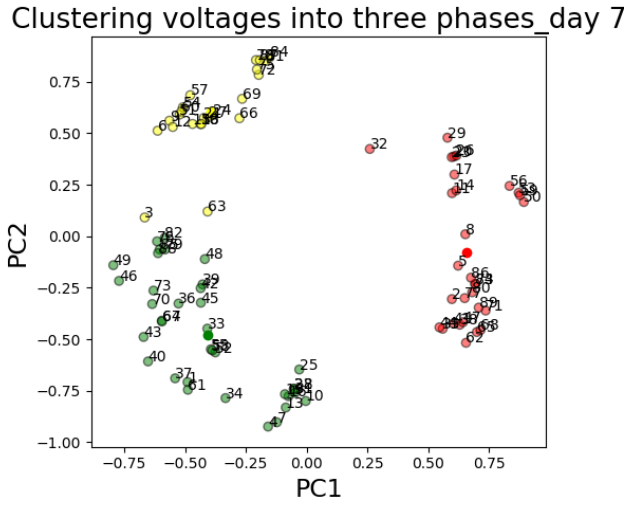


Figure 50: K-means day_7

It can be seen from the figures above that each data point belongs to a group and they are clearly noticeable, labeled with three colours. At this point, we still do not know to which phase each of the colours corresponds to. That is why, one-to-one matching algorithm, that was thoroughly explained before, needs to be implemented.

The procedure for implementing the one to one matching algorithm is the same as before, taking one representative day for defining the phases that correspond to each of the groups. After having the correlation between the groups and the phases the final step for determining the customer's phase is by selecting the most dominant group per customer during the representative week.

Depending on the voltage time-series data, after the performed phase identification algorithm, one consumer can belong to different groups in different days. The group in which one consumer is allocated in most of the days is taken as its group. This, as shown before is represented in figure 37 in which the final method for assigning the phases of the consumers after the k-means clustering and one-to-one matching algorithm was performed. In this case we have 89 consumers and 7 days. The most dominant phase per day for one consumer defines its phase.

In most of the cases, the majority of consumers fed by the four feeders connected to TR1 are allocated to the right phase, but as mentioned before, depending on the voltage time-series readings of the day we calculated different results.

The results for the phase identification algorithm performed per week and two weeks' time period for all the seasons are presented in Table 4. The table shows the number of consumers allocated to the right phase. Since there is a total number of 89 consumers, it can be seen that if we reduce the dimension of the voltage time-series data and we take only the voltage readings for the high demand hours or low demand hours, then the algorithm gives better results.

This is exactly what we obtained in the first case, with feeder one. So, if we exclude certain periods of the day, the phase grouping algorithm of customer's voltages has better accuracy.

As we can see from Table 4, the results obtained for all four feeders are different when the algorithm is performed for just one feeder.

Table 4. Number of consumers allocated to the right phase

Time period	Whole day (12AM-11:30 PM)	High demand hours (6PM – 10 PM)	Low demand hours (11PM – 8AM)
Week in summer	69/89	84/89	84/89
Week in autumn	71/89	85/89	86/89
Week in winter	82/89	87/89	86/89
Week in spring	83/89	87/89	87/89
Two weeks - summer	68/89	87/89	85/89
Two weeks - autumn	75/89	86/89	86/89
Two weeks in winter	86/89	85/89	85/89
Two weeks in spring	84/89	86/89	86/89

The main reason for having these results is the increased number of customers, due to including all the feeders. Still the algorithm in this case shows average 93% accuracy, which is still good percentage of accuracy, but this is clearly for the high demand and low demand hours, whereas for the whole day – the algorithm has lower average percentage of accuracy equal to 87%.

The results presented in Table 4 are in case when there is zero PV penetration in the low voltage feeder. The same analysis were performed with [20%, 40%, 60%, 80% and 100%] PV penetration and the same results were obtained, which gives us the same conclusion that with introduction of PV penetration to the low voltage network, there is no change in the results for the phases of the consumers. The phase identification algorithm is not influenced by the PV penetration on the low voltage feeder.

After analysing the both cases for the Australian LV distribution network, we can conclude that the phase identification algorithm gives better results when it takes the calculations feeder by feeder, then the case when it takes all the feeder fed by the same transformer.

The case in which all the feeders of all four transformers are included in the phase grouping algorithm is not represented in this Thesis, since the behaviour of the algorithm is very similar to the case with four feeders and due to avoiding repetition it was decided to be excluded.

2.5 Chapter Summary

The phase grouping algorithm was based on unconstrained k-means clustering method for defining the groups of the phases, without having any pre-defined (known) connection or topology of the LV distribution network.

In the first case, the performance of the phase grouping algorithm using clustering techniques was demonstrated on a realistic Australian LV feeder with thirty-one customers considering one and two weeks-worth of 30-min voltage time-series data with and without solar PV. In both cases, considering three different time periods: whole day, high demand hours and low-demand hours, the results are promising as all customers were accurately allocated to their corresponding phase group.

In the second case, the performance of the phase grouping algorithm using clustering techniques was demonstrated on a realistic Australian LV network with all feeders fed by one transformer with total number of 89 customers considering one and two weeks-worth of 30-min voltage time-series data with and without solar PV. Same as the previous case with only one feeder, the results are promising as all customers were accurately allocated to their corresponding phase group.

3 Voltage Calculation Using Artificial Neural Network

The increased amount of DER (mostly PV systems) around the world in the last few years gave a lot of challenges to the LV distribution networks. Voltage fluctuations are caused mainly by the increased deployment of PV generation, demand response programs and electric vehicles. Electric utilities are experiencing major issues related to unprecedented levels of load peaks as well as PV penetration. For instance, a solar farm connected at the end of a long distribution feeder in a rural area can cause voltage deviation along the feeder. Moreover, over-voltage happens during midday when the PV generation reaches its peak, while the load demand is relatively low. On the other hand, voltage sags occur mostly during night period due to low PV generation, as well as due to the increased deployment of electric vehicle connections. This highly motivates voltage regulation. The task of maintaining voltage magnitudes, without causing violations, is critical in LV distribution networks.

Conventionally, DNSPs rely on on-load tap changers (OLTCs) and fixed or switched capacitors to maintain the voltage profile across the network in corresponding limits. However, they are limited by number and speed of operations, and insufficient to adapt to highly variable PV production to provide the relevant voltage regulation.

The large deployment of the smart meters in residential LV distribution networks, provides a useful amount of data that could be beneficial to DNSPs for solving the voltage regulation problem by calculating the voltages of the customers. By only using the active and reactive power extracted from the smart meters of each customer, -this work proposes an accurate model based on deep learning method that can determine the voltage profile of each customer in a given LV distribution feeder.

The proposed deep neural network model consists of one input layer, two hidden layers and one output layer. The inputs of the input layer are formed from the active and reactive power of the customer, extracted from the smart meter data, such that for one customer there are only two inputs, for two customers – four inputs and so on. However, the outputs are the voltage profiles of the customers and their quantity is determined by the number of customers. The number of neurons in the hidden layers is determined through trial and error calculation and has different value for different number of inputs.

Differently from other deep neural network models for voltage calculations in the literature, the proposed approach is based on smart meter data only, without knowing the topology of the LV distribution network and without running power flow simulations for covering different scenarios. This makes it even more practical and cost-effective to distribution companies.

The performance of the model for voltage calculation is demonstrated on a realistic Australian LV feeder, by training it for one and two customers considering eighty consecutive days and seventy days of data, respectively, with and without PV systems. After training the neural network, the testing is performed for different days in different season, both with and without PV systems, giving accurate and promising results.

3.1 Literature review

Deep learning (DL) neural network models have so far demonstrated impressive results for a range of highly complex tasks, especially where an accurate mathematical representation of the problem cannot be obtained. The combination of deep learning models with reinforcement learning is becoming more and more attractive in the literature. In [7] deep reinforcement learning (DRL) is applied to coordinate smart inverters with continuous outputs. After proper training, the algorithm can generate timely control decisions to produce actions of smart inverters. The suggested smart inverter actions will be executed to leverage the fast response speed of smart inverters to accommodate PV generation fluctuations. In this work the transient state is not

regarded, as well as in order to perform the training of the DRL algorithm, the authors perform massive offline power flow simulations for calculating the voltage.

The authors of [8] employ a neural network voltage estimator to calculate the voltage profile along a feeder. Remote terminal units (RTUs) send the resulting voltage profile to a master controller aiming at enhancing the operation of an on-load tap-changer (OLTC) transformer for voltage regulation (they have the topology of the network: R, X, length etc). In [9] an approach for estimation of the voltage at specific LV buses is proposed by use of NNs trained on voltage and power measurements from substation level only They estimate the phase-neutral voltage magnitudes (Ua, Ub, Uc) at a downstream bus of a distribution feeder. [10] estimates the voltage drop with taking the topology information as Input to the algorithm. Also, the neural network for estimating the voltage drop is performed on only three nodes LV distribution feeder.

Most of the cases in the literature depend on multiple power flow simulations for obtaining results. Also, for some of them the topology of the LV distribution network is known. Smart meter data is rarely used. Almost all of the cases in the literature are training the neural network models in Matlab toolbox package. The models in this Thesis are based on smart meter data exclusively. Moreover, the active and reactive power of the smart meters are extracted and used as Inputs for the deep neural network, performed in the programming language Python.

3.2 Deep Learning

Deep Learning (DL) is a subset of ML that uses a model of computing that is very much inspired by the structure of the brain, i.e. DL mimics the way our brain functions – it learns from experience. In DL the feature extraction happens automatically. DL will understand which feature or which variable is important in predicting the output. It is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost.

Lately, DL is getting lots of attention and for a good reason. It is achieving results that were not possible before. DL models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. In DL, models are trained by using a large set of labelled (known) data and neural network architectures that have many layers.

After properly training the neural networks, the algorithms can be used for testing on unlabelled (unknown) data in order to predict the output. DL performs "end-to-end learning" – where a network is given raw data and a task to perform, such as classification, and it learns how to do it automatically. A key advantage of DL networks is that they often continue to improve as the size of the data increases. DL is a collection of a ML technologies used to learn feature hierarchies based on the concept of Artificial neural network (ANN).

3.2.1 Artificial Neural Network

Artificial neural networks (ANNs) are computing systems that are highly inspired by the human brain. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in classification, they might learn to classify different types of fruit such as apple, banana, pear by analysing examples that have been manually labelled as "apple", "banana" and "pear". After the learning process, ANN can easily do the classification. They can also do the classification without labelling, such that after training with many examples that will learn to separate the different types of fruit based on their different characteristics.

An ANN is based on collection of connected units of nodes called artificial neurons (or just neurons), which are correlated to the neurons in a human brain. Each connection, like in a biological brain, can transmit a signal to other neurons. The signals in the ANN are numbers.
The structure of the ANN consists of input layer, hidden layer and output layer (as shown in figure 51). In each of the layers there is a specific number of neurons. All neurons are connected between themselves giving a fully connected neural network. In order to reach the neuron of the next layer,

a weight (w) and a bias (b) is added to the neuron. The weight is adjusting during the learning procedure.

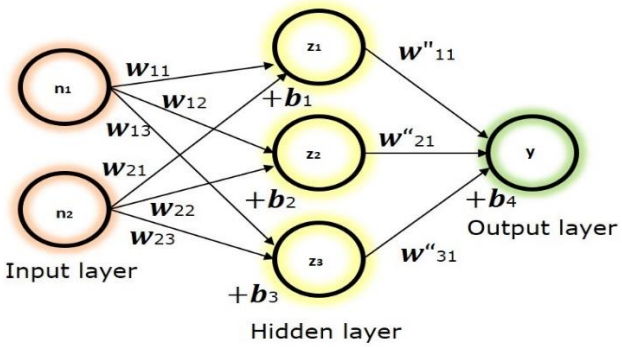


Figure 51.Artificial Neural Network

The sum of the weighted neurons and the bias goes through an activation function and reaches the next neurons. Therefore, the first neuron of the second layer (z_1) is the output of the sum of all the weighted neurons from the first layers and the bias, which is given in the following equation:

$$\boldsymbol{z1 = f[\sum (n1 * w11 + n2 * w21) + b1]}$$

The same procedure applies to the second and third neuron of the second layer (hidden layer) of the ANN and repeats the same for obtaining the output. The process of going from the input layer by adding the weights and biases through activation functions to reach the output is called feedforward method. During the feedforward method the weights are not updating, they are used to predict the output. In order to determine or predict the output of the DNN, an activation function is used.

The activation function determines the accuracy and the computational efficiency of the training. It has the major effect on the NNs ability to converge and the convergence speed of the network. Also, in some cases activation functions might prevent neural networks from converging in the first place. There are three groups of activation functions that might be used in the training of the DNN, such as step functions, linear functions and non-linear activation functions. However, depending on the nature of the input and output values, the mostly used activation functions in

deep learning are the non-linear activation functions. There are different non-linear activation functions, from which only two are used in this work – explained in the methodology section.
Once the output is predicted, the estimated value from the ANN is compared to the real value. The comparison between the estimated and real value of the output is performed by using objective (cost, loss) function.

The loss function is the indicator for the performance of the ANN. The lower the loss function the better the training of the neural network. There is no one-size-fits-all loss function to algorithms in machine learning. There are various factors involved in choosing a loss function for specific problem such as type of machine learning algorithm chosen, ease of calculating the derivatives and to some degree the percentage of outliers in the data set.

There are different types of loss functions used in the literature such as mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean squared logarithmic error (MSLE), cross-entropy loss, multi-class cross-entropy loss, hinge loss, etc.

The loss function used in this work is the MSE, given with the equation:

$$\boldsymbol{MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{pred} - y_{real})^2}$$

where, $\mathbf{y_{pred}}$ is the predicted value from the algorithm while $\mathbf{y_{real}}$ is the real value of the output.

After calculating the loss function, the weights are updated using the following equation:

$$\boldsymbol{w_{new} = w_{old} - \propto (y_{pred} - y_{real})}$$

where, $\mathbf{W_{old}}$ is the old value of the weight, $\mathbf{W_{new}}$ is the new – updated value of the weight and $\propto$ is the learning rate. The learning rate defines the size of corrective steps that the model takes to adjust for errors in each observation. Also, it defines the speed of the training process, such that a high learning rate shortens the training time, but with lower accuracy, while a lower learning rate takes longer with the potential for greater accuracy. In most of the cases, the value of the learning rate is between 0 and 1.

After reaching the output with the feedforward method and updating the weights, the neural network back-propagates by using the backpropagation algorithm explained in the subsequent section.

3.2.2 Backpropagation algorithm

Backpropagation (or "backward propagation of errors") is the most commonly used algorithm for supervised learning of ANNs using gradient descent. Gradient descent is an iterative optimization algorithm of first order that finds a local minimum of a differentiable function.

In this work a classical backpropagation algorithm is used such that once the weights are updated, after performing the feedforward method, the algorithm goes back starting from the output layer. In order to reach the hidden layer, the sum of the weighted neuron of the output (the weight is already updated) and the bias, now goes through the derivative of the activation function and reaches the neurons of the hidden layer.

The same procedure is applied to the neurons of the hidden layer for reaching the input layer. The process is called backpropagation, in which the main difference from the feedforward method is in calculating the derivatives of the activation functions.

The training of the algorithm lasts until two different conditions are reached. One is related to the loss function, such that the algorithm stops training and converges at the same time, when the loss function reaches its pre-defined value, set at the beginning of the training. The second one is related to the number of epochs set at the beginning of the training (in this case the convergence of the algorithm is not guaranteed). The duration of the training of the ANN mostly depends on these conditions and the value of the learning rate introduced in the previous subsection.

When the training of the neural network is done and the algorithm has converged, the ANN is able to give an output for a given input in a very short time.

3.3 Methodology

The methodology for determining the voltage profile of a customer in a LV distribution feeder is based on supervised machine learning. Moreover, the subset of machine learning named deep learning (DL) is used for building the deep neural network (DNN) model. The main idea behind the DL algorithm is that firstly a DNN must be trained with labelled output in order to learn the path for obtaining that output. Later, after properly training the neural network, it should be able to predict any output for any given Input. The backpropagation algorithm explained above is used for the purpose of training of the DNN.

3.3.1 Deep Neural Network Model

Artificial Neural Network that has more than one hidden layer is called Deep Neural Network (DNN). The DNN model for performing the voltage calculation of the customers consists of one input layer, hidden layers and one output layer (as illustrated in figure 52).

The DNN model's training was performed following four steps:

1. Assembling the training data
2. Creating the network
3. Training the network
4. Simulating the network response to new Inputs

In order to create the neural network model, firstly the relevant data was gathered and the number of neurons in each of the DNN layers was defined. The active and reactive power per customer obtained from the OpenDSS software corresponding to smart meter data with 30-minute resolution were taken as an input for the neural network. However, the number of neurons (nodes) for the input and output layer are pre-defined with the number of inputs (P,Q) and

output(s) - V, while the number of neurons in the hidden layer is mostly defined by performing trial and error calculations. After multiple training of the DNN, the structure of the neural network is the following:

One customer: 2 – 15 – 10 – 1

Two customers: 4 – 20 – 10 – 2

Between each of the layers a weight matrix (w) and bias vector (b) are added to each of the neurons. The neural network is fully connected such that all the neurons from the previous layer are connected to all the nodes of the following layer (as illustrated in figure 52).

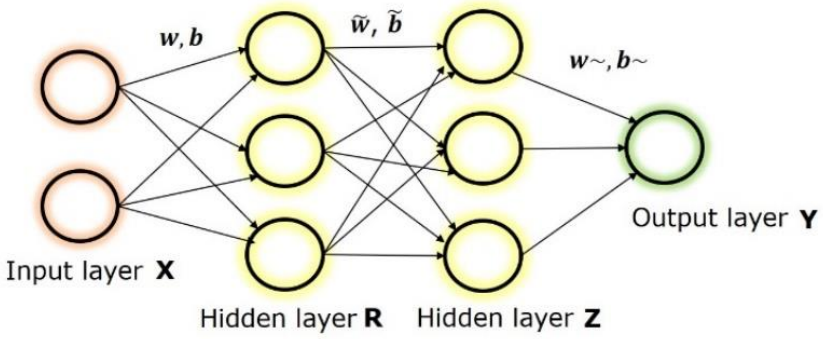


Figure 52.Deep Neural Network Model

In order to get valuable results for given inputs, the DNN firstly must be trained. DNN are part of the supervised ML algorithm, which means the computer is shown in order to learn, i.e. the inputs and the corresponding outputs are known at the moment of training.

The training of the neural network using backpropagation algorithm has the following steps:

1. Firstly, the sum of the weighted input neurons and the biases is calculated
2. Then, that sum goes through an activation function and we get the output of the input (first hidden layer)
3. The same process is repeated for the neurons of the first hidden layer for obtaining its output (second hidden layer) and the second hidden layer for calculating the output

4. After reaching the output, the loss function is calculated, and the weights are updated
5. Then, the backpropagation algorithm is performed, calculating the derivatives of the activation functions
6. The feedforward and backpropagation algorithm are repeated until convergence or until reaching the number of epochs
7. Finally, the neural network is trained

The equations for obtaining the output neurons of each layer are given bellow:

$$\mathrm{R}\mathbf{j} = \mathbf{f}[\textstyle\sum_{\mathbf{j=1}}^{\mathbf{n}}(\mathbf{Xj} * \mathbf{w}) + \mathbf{b}] - (\textbf{output of first layer})$$

$$\mathrm{Z}\mathbf{j} = \mathbf{f}[\textstyle\sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{Rj} * \widetilde{\mathbf{w}} + \tilde{\mathbf{b}}] \ - (\textbf{output of second layer})$$

$$\mathrm{Y}\mathbf{j} = \mathbf{f}[\textstyle\sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{Zj} * \mathbf{w}\sim + \mathbf{b}\sim] \ - (\textbf{output of third layer})$$

where $\mathrm{Rj}, \mathrm{Zj}, \mathrm{Yj}$ are the outputs of the input layer, first hidden layer and second hidden layer, respectively.

The activation functions used for calculating the voltage profile of the customers are sigmoid and rectified linear unit activation function (ReLU) (depicted in figure 53). Both sigmoid and ReLU are non-linear activation functions. They determine the accuracy and computational efficiency of the trained model and have a major effect on the DNNs ability to converge.

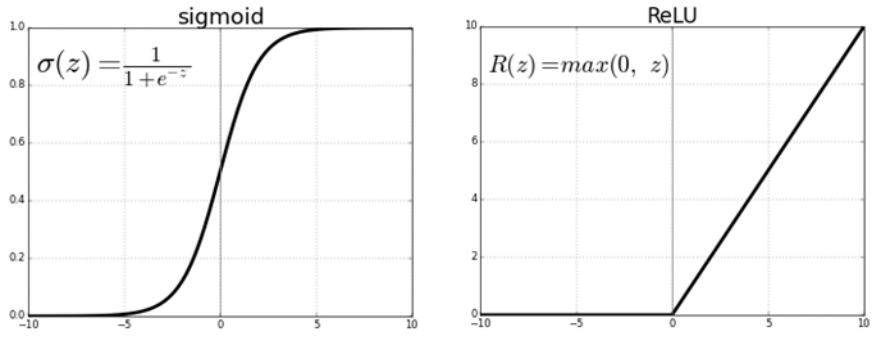


Figure 53. Activation Functions

The process of going through all the layers from the input layer to the output is the feed-forward algorithm. After the feed-forward process the value of the predicted output is compared with the value of the real output, and the loss function is calculated. If predictions deviate too much from actual results, loss function would cough up a very large number. Gradually, with the help of some optimization function, loss function learns to reduce the error in prediction. Here, the mean squared error is used. Since the large number of the loss function in the first iteration, the algorithm goes back computing the derivatives of the activation functions and updating the weights (the algorithm backpropagates).

For training the neural network, the value of the learning rate was 0.09 in the case for one customer with 400 epochs, while for the case with two customers, connected to the same phase, the learning rate was 0.001 with 300 epochs.

The number of samples used for the training are 3840 and 3360, equivalent to eighty and seventy days respectively, from which half of the days are in case where there are no PV systems and half of the days are with PV systems. Each of the days used for the training and testing of the algorithm has different load profile and random power factor assigned in range of 0.9 to 0.98 for each time step. The PV profiles used were extracted from the pool of profiles for the year of 2014.

3.3.2 Software Tools

The software tools used for the second research question are the following:

1. OpenDSS

OpenDSS is an open source distribution system simulator that has been developed by electric power research institute (EPRI), USA [7]. OpenDSS was used for obtaining the active and reactive power (P, Q) of the customers corresponding to data extracted from the smart meters.

2. Python

Python is an open source programming language that has higher level of abstraction and simplicity [8]. Even though there are libraries for building deep neural network models, such as Keras, still in this work a full code written in Python was used, in order to track the mean squared errors in each epoch for all of the training samples used.

3.3.3 Summary of Methodology

The proposed deep neural network model consists of four layers that correspond to one input layer, two hidden layers and one output layer. The number of neurons in the input and output layers depends on the number of customers for which the voltage calculation is performed, while the number of neurons in the hidden layer is determined due to trial and error calculation.

Two non-linear activation functions were used in order to predict the output of the neural network: sigmoid and rectified linear unit activation function, while the backpropagation algorithm was used for training the model.

The training of the DNN model is performed in the environment of the programming language Python, after extracting the values for the active (P) and reactive (Q) power derived from the OpenDSS software for power flow analysis. P and Q of the customer were used as input data for the model.

The amount of data used for the training of the model covered thirty consecutive and sixty days, from which half of them were without PV systems and the other half were with included PV systems. The performance of the algorithm is tested on Australian LV distribution feeder and explained in the following case study.

3.4 Case Study

In this section the case study for the proposed deep neural network algorithm is presented. It consists of two scenarios. The first one regards the case when only one customer is connected to the LV distribution feeder and the voltage profile for that customer is calculated by using the deep neural network model, whereas the second one regards the case when two customers are connected to the LV distribution feeder and the voltage profile for both of the customers is calculated using the proposed model.

The case study is performed on the same Australian LV distribution feeder, as shown in figure 33, such that in order to obtain the input and the output time-series data for the neural network, firstly a load flow analysis was done using the OpenDSS software "driven" by the programming language Python, that corresponds to the smart meter data of the customers.

Firstly, in the following subsection the input data and output data, as well as the parameters of the grid are presented, following by the two case studies observed for one customer and two customers connected to the residential LV distribution feeder.

3.4.1 Input data and parameters of the LV network

In this subsection the parameters of the residential LV distribution network are provided, followed by the input data used for the training of the neural network.

The residential LV distribution network model and profiles used in this section were developed based on the LV Network data provided by AusNet (Australian energy company). The main objective of this part is to gain understanding of the characteristics and behaviour of the LV distribution network. To achieve this objective, realistic profiles and feeder models were used during the study.

From the AusNet LV Network, that consists of 4 three-phase 500 kVA Transformers 22/0.433 kV/kV, to simplify analysis, Transformer 1 is chosen. Moreover, only feeder 1 from TR1 is taken into consideration, from which the analyses are done for one and two consumers. The total number of consumers fed by feeder one is thirty-one, from which two cases with only one and only two customers (connected to the same phase, as shown in figure 54) are observed.

The LV network's configuration is shown in figure 54, where two loads are emphasised as representative loads. The characteristics of feeder 1 are provided in Table 5.

Table 5. Characteristic of feeder 1

No. of Customers	No. of Lines	Substation transformer	Frequency
31	24	22/0.433 kV/kV, 3 phases, 500kVA(△Y)	50Hz

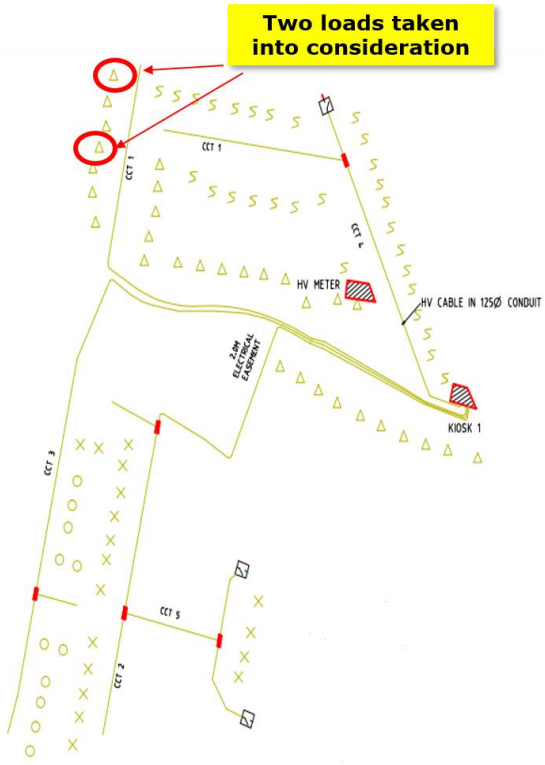


Figure 54. Network configuration

The load profiles used for the customers were obtained from a pool of different load profiles from the smart meter data provided with time resolution of 30 minutes. The data used for the PV profiles was obtained from the solar profiles data for 24 hours with 30 min resolution for Australia in year 2014. These profiles are created for the same sun irradiance and same size of PV units.

For the two case studies represented, the active and reactive power profiles of the customers for time period of eighty and seventy days, respectively were used as input, for calculating the voltage profile of the customer(s) for the same time duration as the input profile.

In the figures below, the load profiles used for the training are represented for the case with only one customer (last customer fed by the feeder – furthest from the head of the feeder). Figures 55 and 56 depict the active and reactive power respectively of the customer for time period of thirty days. The load profiles used for the training are taken for one month in summer. It can be seen from figure 55 that the demand that last customer can have is in range of 0 – 3 kWs.

Figure 55. Active power profiles – last customer

The reactive profiles of the customers are obtained by calculating the product of the demand multiplied with a random power factor per time period in the range of 0.9 to 0.98. Therefore, the reactive load profiles of the customer are in range of 0 to 1.2 kVars, as depicted in figure 56.

Figure 56. Reactive power profiles – last customer

It can be seen from the figures above for the load profiles, that the maximum consumption is during the evening, while the minimum consumption is during the night-time period, for the last customer fed by the feeder.

For the case with two customers, connected to the same phase, fed by the same feeder, the input data consists of load profiles for sixty consecutive days in summer, for both customers. In the figures below, the load profiles for both customers are depicted. In figures 57 and 58, the active power profiles for the first and the second customer, respectively are depicted. The first customer is closer to the head of the feeder with respect to the second customer.

Same as the previous case, the reactive power profiles for both customers are depicted in figures 59 and 60, where the reactive profile is a product of an active profile and the random power factor applied for each time step during the day, in the range of 0.9 to 0.98.

Figure 57. Active power profiles - first customer

Figure 58. Active power profiles - second customer

Similarly, as in the previous case where there was only one load, the active profiles in case with two loads are in range of 0 – 4 kWs.

Figure 59. Reactive power profiles - first customer

Figure 60. Reactive power profiles - second customer

The reactive profiles for both customers, are following the behaviour of the demand multiplied with the random power factor.

Table 6. Input data for the neural network

P_load31	Q_load31
0.9280	0.4056
0.9300	0.4065
1.0320	0.4511
1.0920	0.4773
1.0400	0.4546
0.3720	0.1626
0.1520	0.0664
0.1040	0.0455
0.0480	0.0210
0.0480	0.0210
0.0580	0.0254
0.0700	0.0306
0.0700	0.0306
0.0680	0.0297
0.4260	0.1862
0.0680	0.0297
0.3060	0.1338
0.2520	0.1101
0.1640	0.0717
0.0680	0.0297
0.2060	0.0900
0.1800	0.0787
0.6660	0.2911
0.6160	0.2693
0.2700	0.1180
0.2100	0.0918
0.2620	0.1145
0.1500	0.0656
0.1420	0.0621
0.1400	0.0612
0.1380	0.0603
0.1380	0.0603
0.0600	0.0262
0.0680	0.0297
0.0780	0.0341
0.0840	0.0367
0.0820	0.0358
0.4820	0.2107
0.6600	0.2885
0.6520	0.2850

P_load1	Q_load1	P_load2	Q_load2
0.3480	0.2062	0.1140	0.0517
0.3040	0.1423	0.1160	0.0348
0.4940	0.1199	0.1260	0.0366
0.9120	0.1326	2.2240	0.9498
1.3940	0.1628	1.8580	0.8172
1.3940	0.2923	0.1660	0.0395
1.3400	0.3615	0.1800	0.0493
1.3700	0.4012	0.5500	0.1454
1.0500	0.0743	0.1200	0.0525
1.2000	0.1158	0.1200	0.0428
1.3340	0.1906	0.1180	0.0397
1.4120	0.2211	0.1160	0.0376
1.2280	0.2458	0.1200	0.0261
1.6500	0.1682	0.1000	0.0458
1.0900	0.0555	0.1280	0.0481
0.3480	0.1051	0.1300	0.0467
0.4500	0.0704	0.1660	0.0632
0.8340	0.1770	0.3120	0.0984
2.0160	0.1326	0.9520	0.3199
1.8320	0.0521	1.8740	0.4110
1.5860	0.1322	1.3120	0.3052
1.6960	0.0371	1.3760	0.2830
2.0400	0.1943	1.3320	0.5098
1.6500	0.1193	0.2140	0.0897
1.5500	0.1188	0.1220	0.0526
1.4340	0.0727	0.1420	0.0382
0.9580	0.0684	0.1360	0.0298
0.3180	0.0905	0.1240	0.0544
0.2700	0.0802	0.1240	0.0319
0.4260	0.1356	1.2320	0.3164
0.2040	0.1955	0.5660	0.2442
0.3160	0.1094	0.7160	0.2386
0.2360	0.0758	0.4640	0.1763
0.5220	0.0477	0.3380	0.0771
0.4540	0.0606	0.3320	0.1013
0.3480	0.0689	0.2900	0.1316
0.3040	0.1412	0.2940	0.1019
0.4940	0.0757	0.1160	0.0264
0.9120	0.0607	0.1260	0.0543
1.3940	0.0921	0.1120	0.0401

0.5340	0.2334
0.1580	0.0691
0.1500	0.0656
0.1480	0.0647
0.0740	0.0323
0.0760	0.0332
0.2840	0.1241
1.1460	0.5009

1.3940	0.1102	0.1160	0.0401
1.3400	0.1313	0.1160	0.0511
1.3700	0.1042	0.1160	0.0479
1.0500	0.1075	0.2700	0.1287
1.2000	0.3382	0.1920	0.0780
1.3340	0.6116	1.3460	0.6306
1.4120	0.6189	0.9180	0.4314
1.2280	0.3711	1.3900	0.3276

The input data for the neural network is gathered by taking the active and reactive profiles per customer. For the case with only one load connected, there are only two inputs representing two columns of the active and reactive profiles for a 80 days duration with 30-minutes resolution from which one day of data is shown in table 6 (left part).

Furthermore, for the case with two customers, there are four inputs representing four columns of the active and reactive profiles of both customers for seventy consecutive days in summer, with the same resolution as for one customer, from which one day is shown in table 6 (second part).

3.4.2 Case 1: One Customer Connected to LV feeder

The training for the case of one customer fed by feeder one starts after getting the active, reactive power (as explained in the previous subsection) and the voltage of the last customer connected to the LV feeder, for a time period of eighty days, from which forty are without PV systems and forty are with PV systems. Since the data resolution is 30-minutes, there are 3840 samples per input variable. The number of neurons in each of the layers is 2 – 15 – 10 – 1 , for the input, first hidden, second hidden and output layer respectively, while the activation functions used for obtaining the neurons in the hidden layers and the output layer are sigmoid, sigmoid and rectified linear unit activation functions, respectively.

In the results subsection, firstly the training results of the deep neural network are given followed by the mean squared error (MSE) and then the testing results show the performance of the neural

network for taking as input different days in different season, as well as cases when the demand is doubled. Later, the training without PV systems and the testing with PV systems only is shown.

Results

The results for the training of the deep neural network are depicted in figure 61. The training of the neural network is done for one customer with different load profiles per day and random power factor in the range of 0.9 – 0.98. The total duration for the training performed was thirty minutes.

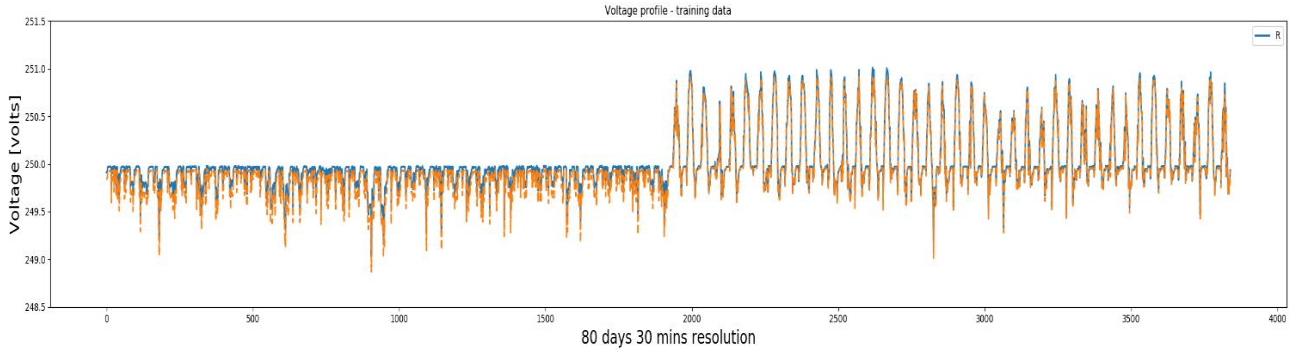


Figure 61. Training results of the neural network

From the figure above, with blue colour is represented the real value of the voltage profile of the customer, while with orange (dashed line) the estimated voltage profile obtained from the DNN model. The figure shows overlapping of both real and estimated value of the voltages of the customer, and the calculated mean squared error from the training is given in figure 62.

Figure 62 shows the MSE of all the samples in each epoch. The total number of epochs is equal to 400, but it can be seen from the figure that the algorithm converges sooner than reaching the total number of epochs.

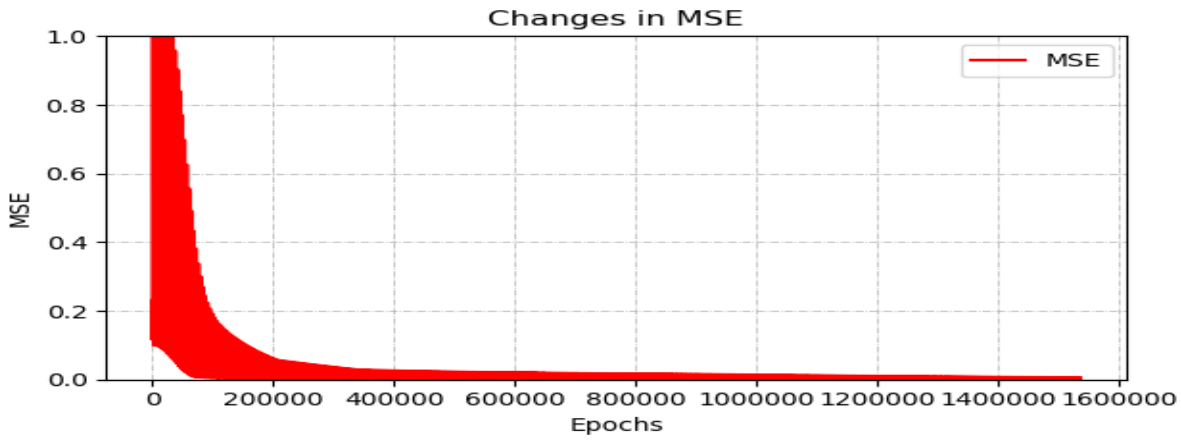


Figure 62. Mean Squared Error

However, the difference between the results obtained from the training and the real values of the voltages is shown in figure 63.

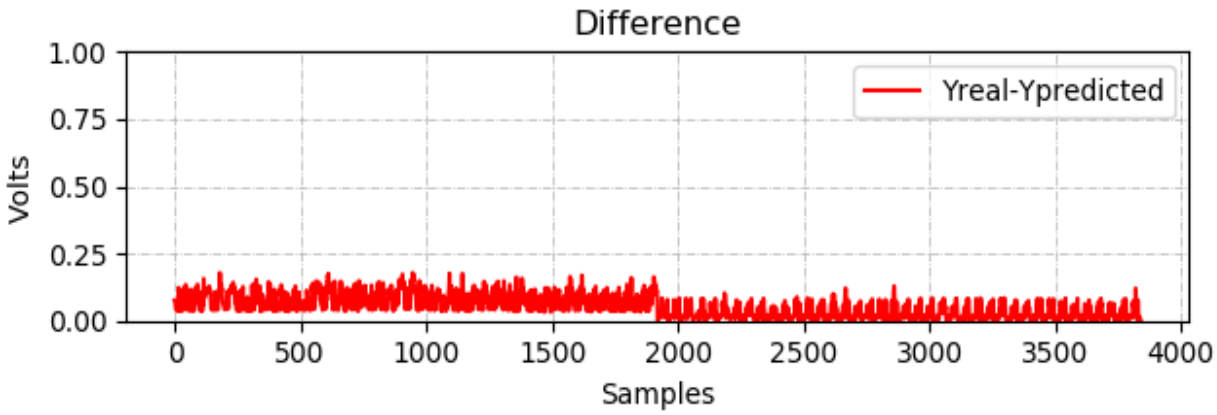


Figure 63. Difference of real and estimated values

Since the value of the MSE is less than 0.01 (close to zero) and the difference in volts is less than 0.25 per sample, we can say that the neural network has been trained well and further testing is performed, keeping the final weights from the training.

The first testing is done for forty days from a different season, dividing the days in half with and without PV systems. The results of the testing are shown in figure 64, while the difference in volts between the real values and the predicted values is depicted in figure 65.

The days from the different season used in the testing have a random power factor in the range of 0.9 – 0.98, and the MSE in this case has a value lower than 0.1, which confirms the results depicted in figures 64 and 65.

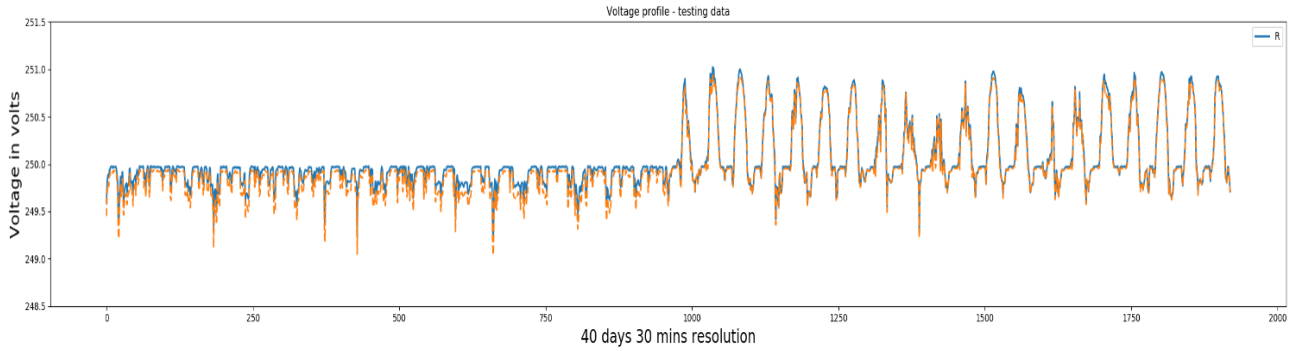


Figure 64. Testing with different days

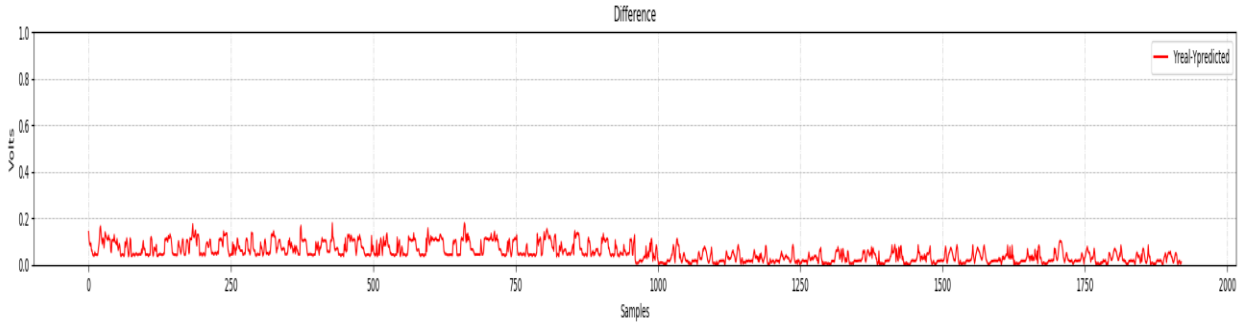


Figure 65. Difference of real and estimated values

The second performed testing using the trained DNN is done for the case when the demand has a double value. The results of the testing are depicted in figure 66, while figure 67 depicts the difference in volts between the real and estimated value.

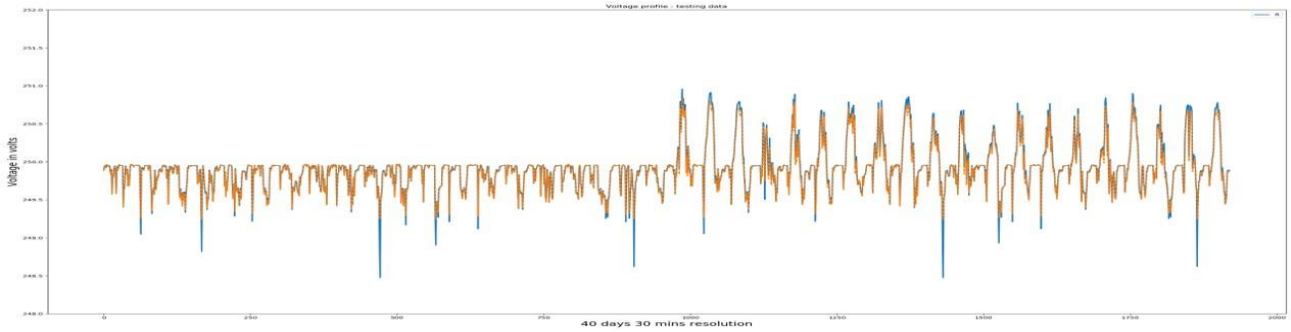


Figure 66. Testing with double P

The testing of the model in case with double demand is done for forty days, from which twenty are with and twenty are without PV systems. It can be seen from figure 66 that there are a few data samples in which the real value and the calculated value of the voltage are not exactly the same, which is due to the fact that the neural network has not seen a case before in which the input has such high value. That can be also seen from figure 67, where the difference in volt is depicted.

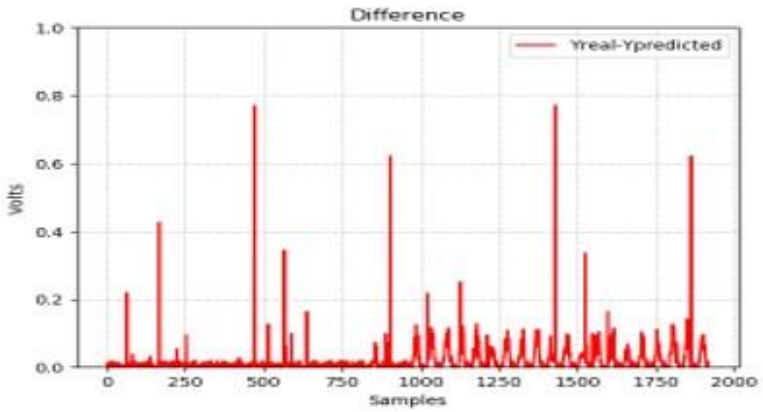


Figure 67. Difference between real and estimated value

Even though there is a clear difference between certain samples, the MSE has a value around 0.2, which is considered as an acceptable error.

In order to note the importance of using days both with and without PV systems, a training for the last customer fed by the LV distribution feeder is done without any photovoltaics. The results from the training are represented in figure 68.

Figure 68. Training data results without PV systems

Since there are no PV systems included, the number of days is significantly reduced because the DNN needs less days and epochs to reach convergence. The MSE in this case is zero, which can be also seen from the figure, since both the real (blue colour) and the estimated (orange colour) values for the voltages are overlapping.

After training the neural network without any PV systems, a testing is performed for forty days but this time with PV systems only. The results from the testing are depicted in figure 69.

It can be seen from the figures below, that the performance of the neural network for the testing with PV systems only, gives unsatisfactory results. There is a clear mismatching between the real and estimated values of the voltage of the customer as shown in figure 69, while figure 70 depicts the difference in volts. The behaviour in this case is due to the absence of PV during the training of the neural network. The DNN performs better and predicts with higher accuracy when different scenarios are given for the training process.

Figure 69. Testing with PV systems only

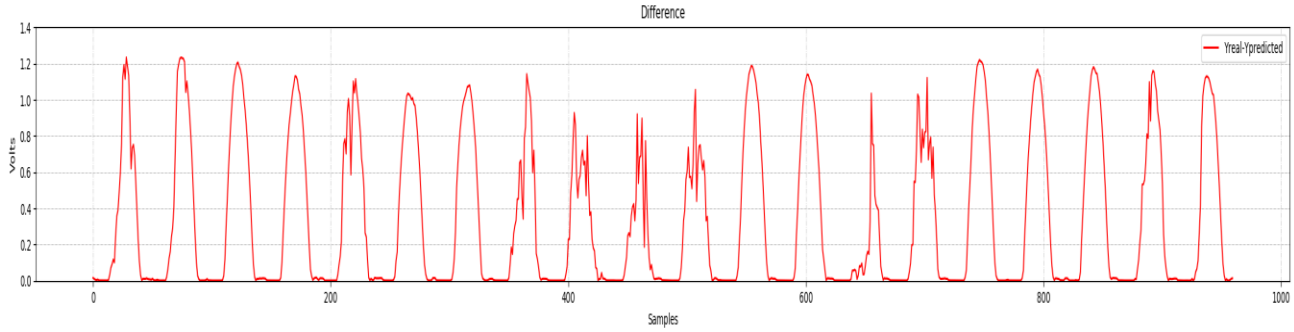


Figure 70. Difference of real and calculated value

3.4.3 Case 2: Two Customers Connected to LV feeder

The second case regards the training of the neural network when two customers connected to the same phase are fed by the LV distribution feeder, as shown in figure 71. The training of the DNN takes four inputs: P and Q for the first customer and for the second customer, for a time period of seventy days, from which half days are with and half without PV systems and two outputs – the voltage profiles of the customers for the same time period.

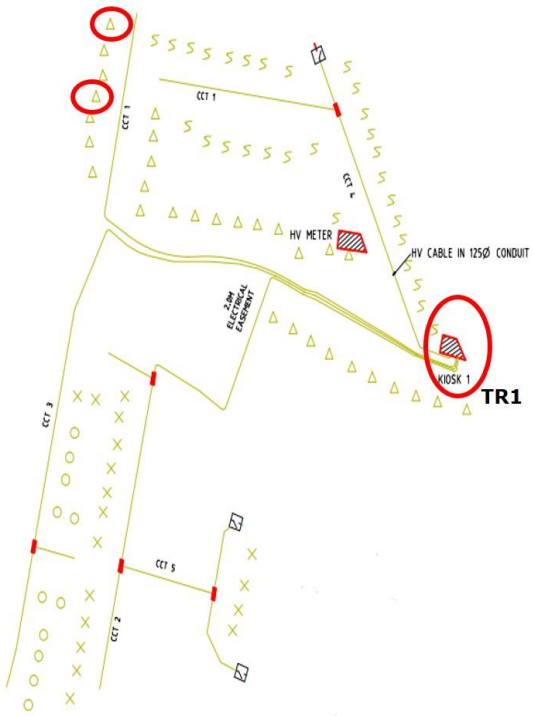


Figure 71. Two loads fed by LV feeder

Since the data resolution is 30-minutes, there are 3360 samples per input variable. The number of neurons in each of the layers is 4 – 20 – 10 – 2 , for the input, first hidden, second hidden and output layer respectively, while the activation functions used for obtaining the neurons in the hidden layers and the output layer are tanh, sigmoid and rectified linear unit activation functions, respectively.

In the results subsection, same as the previous case, firstly the training results of the deep neural network are given followed by the mean squared error (MSE) and then the testing results show the performance of the neural network for taking as input different days in different season, as well as cases when the demand is doubled. The case when the training is done without PV and tested with PV systems is excluded, since its performance was shown in the previous case and no better performance was expected in this case.

Results

The results of the training of the DNN for the case with two customers at the end of the feeder, connected to the same phase, for seventy days with different load profile per day and random power factor in the range of 0.9 to 0.98 per time period are depicted in figures 72 and 73.

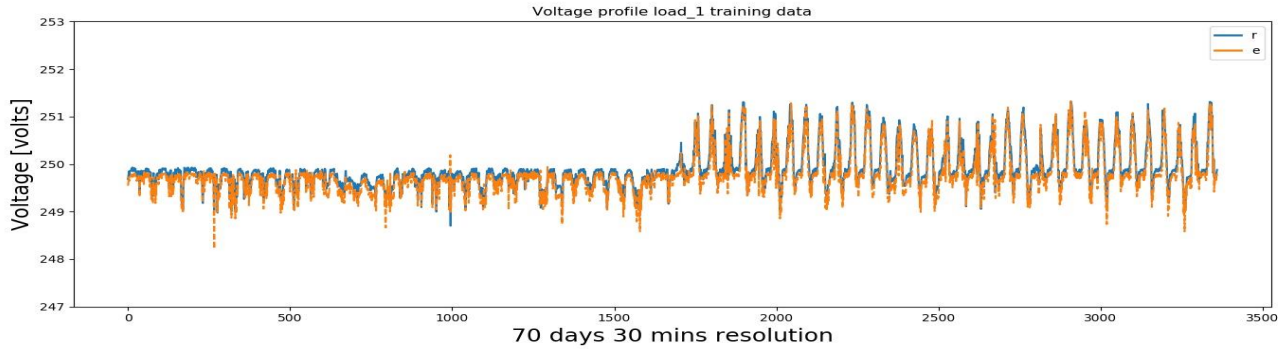


Figure 72. Training results - first customer

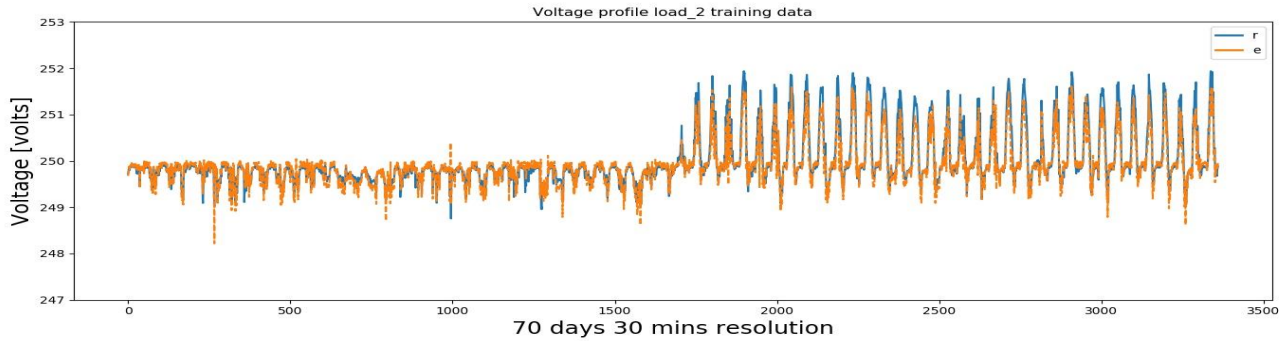


Figure 73. Training results - second customer

The learning rate used for the training was 0.001 for total number of epochs equal to 300. The total duration of the training was 30 minutes, while the MSE after the 300 epochs for all 3360 samples was less than 0.15 (as shown in figure 74).

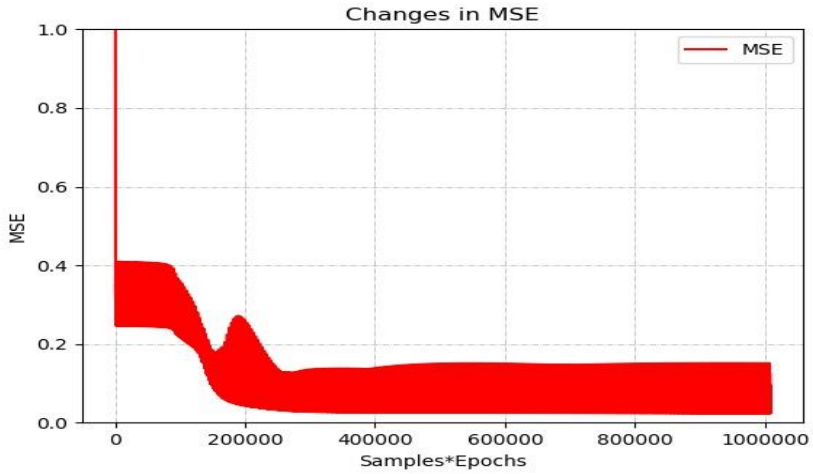


Figure 74. MSE for the training

The first testing is done for forty days from a different season, dividing the days in half with and without PV systems. The results of the testing are shown in figures 75 and 76, while the difference in volts between the real values and the predicted values is depicted in figures 77 and 78 for both customers.

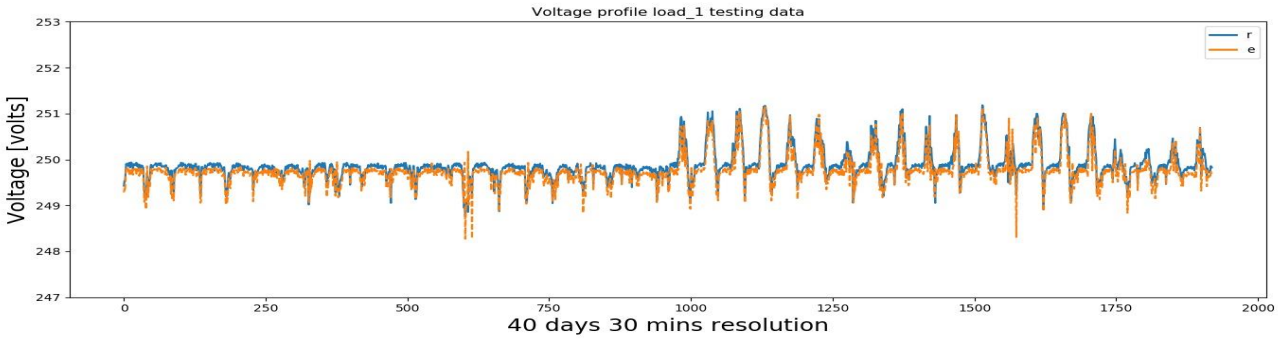


Figure 75. Testing results for the first customer

The results from the testing done for both customers fed by feeder one, for forty different days from different season show good performance of the trained neural network, with MSEs equal to 0.02835 and 0.03128, respectively as shown in figures 77 and 78.

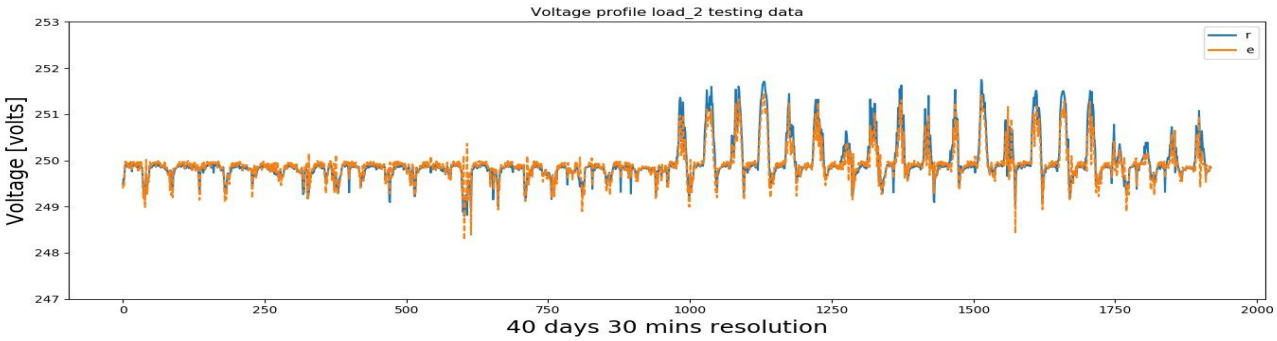


Figure 76. Testing results for the second customer

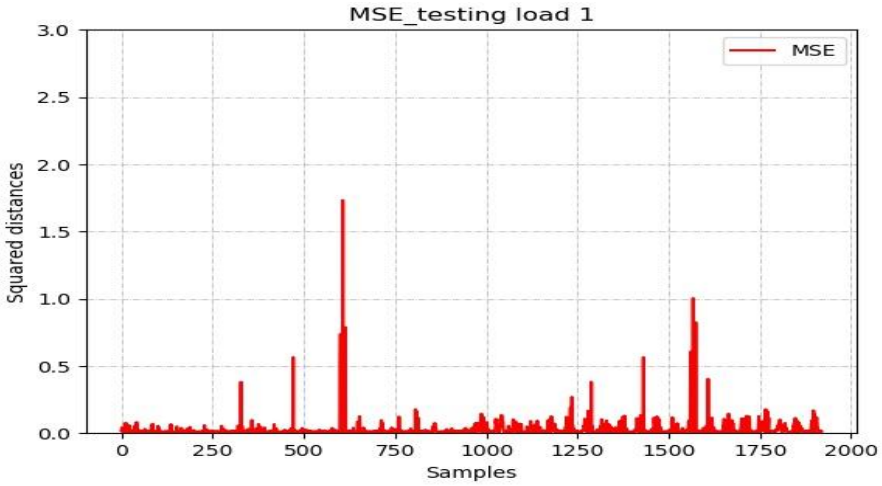


Figure 77. Difference in volts for load 1

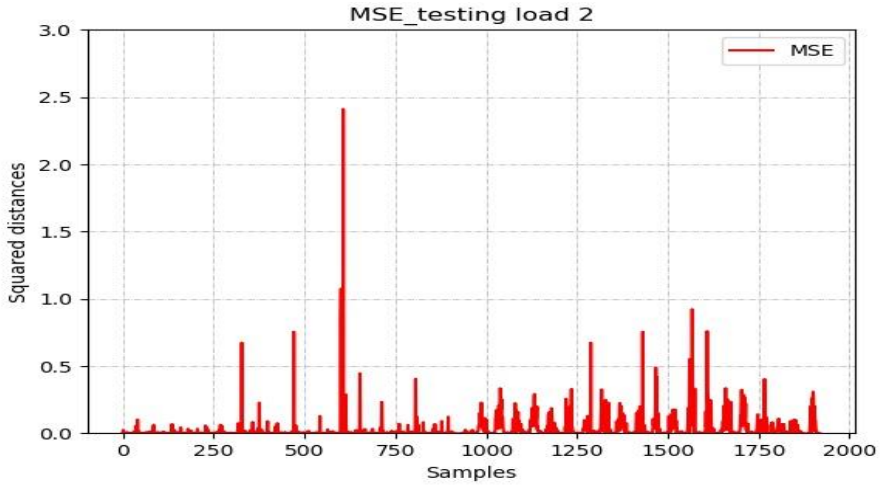


Figure 78. Difference in volts for load 2

The last testing performed for the case with two customers is when the demand has a double value. Same as before, for both customers forty days are observed, divided in half with and without PV systems. The results obtained from the testing are shown in figures 79 and 80, respectively.

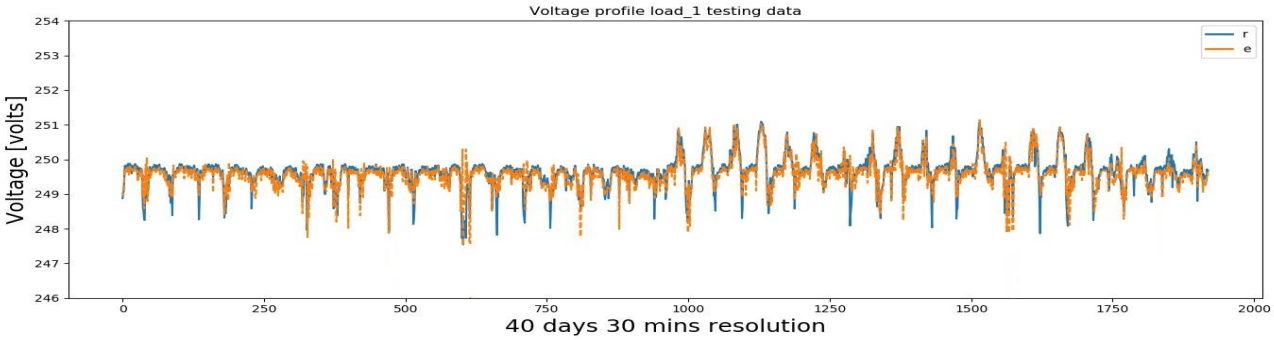


Figure 79. Testing results with double P

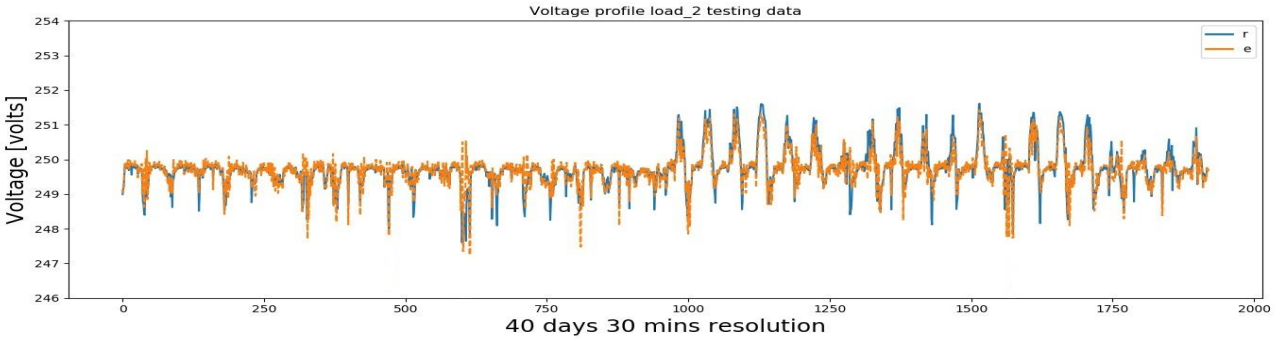


Figure 80. Testing results with double P

The results in figures 79 and 80 show good overall performance of the algorithm in case when the demand is doubled, with couple of mismatches for both customers. The MSEs in this case are equal to 0.11178 for customer 1 and 0.12227 for customer 2.

3.5 Chapter Summary

The calculation of the voltage profile of a single customer and two customers fed by the same feeder was performed by using a deep neural network (DNN) model. The DNN consisted of four layers – one input, one output and two hidden layers. The inputs taken for the DNN depended on the number of customers, such that for each customer the active and reactive power data was firstly extracted from the smart meter and then taken as an input to the neural network.

For the training of the neural network in the first case - for one customer fed by the LV distribution feeder, two different activation functions were used: sigmoid and rectified linear unit activation function. The number of neurons in each of the hidden layers was 15 and 10 respectively, obtained by trial and error calculation. The training was done for a time period of 80 days, dividing the days in such a way that half were in case where there are no PV systems, while the other half was with PV systems included. Different load profiles were used for each day, while the power factor was taken randomly in a range of 0.9 to 0.98 for each time step.

The testing was performed regarding two different inputs: in the first one 40 days from a different season were taken as an input, while in the second case the input was in case when the demand has a double value. In both cases, the model performed well, by accurately calculating the voltage profile of the customer.

For the second training with two customers fed by the same feeder, connected to a same phase, three different activation functions were used: tangent hyperbolic, sigmoid and rectified linear unit activation function. The number of neurons in the hidden layers was 20 and 10, respectively, defined by trial and error calculations. The training of the network regarded 70 days for both customers, with different load profiles and random PF in range of 0.9 to 0.98 for each time step.

The testing in the second case regarded the same scenarios as per one load (different days and double demand), showing good performance by accurately calculating the customers' voltages.

4 Conclusions

Artificial Intelligence (AI) applications based on smart meter data could be used for solving some of the LV distribution network problems. In this work the subsets of AI, the well-known machine learning and deep learning were used in order to build algorithms for defining the phase of the customers and calculating their voltage profiles.

The first chapter is dedicated to the phase grouping using clustering techniques which showed good results, allocating the customers to their corresponding phase, in both cases presented: for customers connected to a single feeder, and for all customers fed by feeders connected to the same transformer. Using the phase grouping algorithm, distribution network service provides could be able to better understand the topology of the residential LV distribution network, by only having the voltage profiles of the customers. Still, further research in this area is needed in case when two phases are the most used phases.

On the other hand, in the second chapter promising initial results for voltage calculation of a single customer and two customers fed by a LV distribution feeder were provided. A deep neural network was trained by taking as input data the load profiles of the customers extracted from the smart meter data. The main idea behind calculating the voltage profiles of the customers is for building an application for future calculation, for any given load profiles, where the DNN could be able to extrapolate the voltages of the customers.

Further research in this area will consist of training the deep neural network for voltage calculation for all customers fed by the feeder and expanding it for the whole residential LV distribution network, as well as what would happen if PV inverters are adapting to voltages (volt-watt and volt-var functions).

5 References

1. K. Y. Khumchoo and W. Kongprawechnon, "Cluster analysis for primary feeder identification using metering data," 2015 6th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES), Hua-Hin, 2015, pp. 1-6.
2. *L. Blakely, M. J. Reno and W. Feng, "Spectral Clustering for Customer Phase Identification Using AMI Voltage Timeseries," 2019 IEEE Power and Energy Conference at Illinois (PECI), Champaign, IL, USA, 2019, pp. 1-7.*
3. V. Arya et al., "Phase identification in smart grids," in 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2011, pp. 25–30.
4. R. Mitra et al., "Voltage Correlations in Smart Meter Data," ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 1999–2008, 2015.
5. W. Wang, N. Yu, B. Foggo, J. Davis, and J. Li, "Phase Identification in Electric Power Distribution Systems by Clustering of Smart Meter Data," in 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), 2016, pp. 259–265.
6. F. Olivier, A. Sutera, P. Geurts, R. Fonteneau, and D. Ernst, "Phase Identification of Smart Meters by Clustering Voltage Measurements," Power Syst. Comp.u.t. Conf. PSCC, 2018
7. R. C. Dugan and T. E. McDermott, "An Open Source Platform for Collaborating on Smart Grid Research," in IEEE Power and Energy Society General Meeting, Detroit, USA, 2011
8. Python Software Foundation, "Python," [Online]. Available: https://www.python.org/. [Accessed 2019 11 21]
9. Home – Keras Documentation [Online]. Available: https://keras.io/
10. Scikit -Learn, Machine Learning in Python [Online]. Available: https://scikit-learn.org/stable/
11. *C. Li, C. Jin and R. Sharma, "Coordination of PV Smart Inverters Using Deep Reinforcement Learning for Grid Voltage Regulation," 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 2019, pp. 1930-1937.*
12. *R. Kamali, R. Sharifi, H. Radmanesh, S. Fathi "Online voltage estimation for distribution networks in presence of distributed generation" Indian J. Sci. Technol., 9 (18) (2016), pp. 1-5, 10.17forty-eight 5/ijst/2016/v9i18/71forty-eight 4*
13. *Pertl, M., Douglass, P.J., Heussen, K., & Kok, K. "Validation of a robust neural real-time voltage estimator for active distribution grids on field data" (2018)*

14. *Z. Wacławek, "Application of artificial neural networks for estimation of some quantities in electrical networks," 2010 9th International Conference on Environment and Electrical Engineering, Prague, 2010, pp. 1forty-eight -150.*
15. *W. Svante "Principal component analysis" Chemometr. Intell. Lab. Syst., 2 (1987), pp. 37-52, 10.1016/0169-7439(87)80084-9, 1987*