Faculty of Electrical Engineering
Wroclaw University of Science and Technology

Doctoral thesis

Microgrid energy management systems with hybrid optimizers, embedded deep learning forecasters and e-vehicle charging stations

Doctoral student:
Vishnu Suresh
Supervisor:
dr hab. inż. Przemysław Janik
Co – supervisor:
dr inż. Zbigniew Wacławek

Keywords: Energy management systems, solar forecasting, meta-heuristic optimization, MIDACO, electrical vehicle charging station, optimal location, microgrids, minimisation of cost, LCOE.

Wroclaw 2021

Acknowledgements

I would like to thank everyone who was involved directly and indirectly in the author's doctoral journey.

Most importantly dr hab. inż. Przemysław Janik for looking after the interests of the doctoral student throughout the studies and providing guidance regarding scientific research and cultural integration.

dr inż. Zbigniew Wacławek for being the co-supervisor and providing guidance and assisting in various administrative procedures.

prof. dr hab. inż. Zbigniew Leonowicz for taking a keen interest in the author's activities and raising his potential to the international level of conferences and journals and constantly providing opportunities for improvement in terms of research, self-development and international collaboration.

Colleagues and friends who have helped relentlessly when it comes to access to resources, administrative procedure and learning the local language and culture.

Finally, Parents of the author for providing financial assistance during times of difficulties without which the path for a doctoral degree would not have been possible

Abstract

The modern electrical power system is characterised by increased complexity and dynamics. This outlook is partly motivated by the society due to increased motivation to move to a carbon neutral environment. The pathway to such an environment has led to higher penetration of renewable energy sources, research into improving efficiency of energy systems, optimizing the operation of existing assets, e-mobility and decentralisation of the power system itself.

The above-mentioned approaches to a net zero emissions environment creates additional challenges such as the increased share of renewables in the energy mix which leads to uncertainty in the power outputs making the problem of energy balancing more complex. Optimizing the power system involves deployment of optimization algorithms (conventional and intelligent) modelled specifically for intended power networks. E-mobility while decarbonising the transportation sector represents a significant load with a stochastic nature. Finally, the aspect of decentralisation leads to questions regarding energy security and independence from the power system operators.

This thesis attempts to address a few of these challenges. It explores the use of deep-learning based forecasting algorithms such as the Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), CNN-LSTM and other similar architectures in-order to model the stochastic nature of power outputs from renewable energy sources. It also deploys certain statistical tools such as the Markov chains for uncertainty analysis of the forecasts made by such algorithms. For the purpose of optimization hybrid optimizers such as the Mixed Integer Distributed Ant Colony Optimization (MIDACO) – MATPOWER amongst others are used in-order to leverage both the global solution search capabilities of intelligent optimization algorithms and the speed of MATPOWER. It also explores the use of Genetic Algorithm (GA) for energy management. Problems concerned with E-mobility are addressed by way of deploying co-ordinated charging in Electrical Vehicle Charging Stations (EVCS) in order to manage the sudden increase in energy demand and finding the optimal location for such an EVCS within the existing network. Finally, all the above-mentioned methods are applied to networks of different configurations under different scenarios in order to understand, extend and generalise their applicability.

This study uses a combination of both real and modelled data obtained from installations available locally and from collaborating institutions abroad. All the power flows calculated are based on the alternating-current model

Contents

Acknowledgements 2

Abstract 3

1. Introduction 6

1.1 Microgrid 7

1.2 Forecasting 10

1.3 Optimization 22

2. Solar Forecasting 30

2.1 Data pre-processing 30

2.2 Evaluation metrics 31

2.3 Convolutional Neural Networks (CNNs) 31

2.4 LSTM – Autoencoder (L-AE) 41

2.5 ARIMA 44

2.6 Results 46

Conclusions 50

3. Energy Management (Optimal Power Flow) 52

3.1 Power flow 52

3.2 Optimal Power flow 63

3.3 Hybrid meta-heuristic optimization approaches 67

3.4 Genetic – Algorithm (GA) 68

3.5 Particle Swarm Optimization (PSO) 72

3.6 Mixed Integer Distributed Ant Colony Algorithm (MIDACO) 74

3.7 Political Optimizer (PO) 77

3.8 Lichtenberg Algorithm (PO) 79

3.8 Performance evaluation of meta-heuristic approaches to OPF 82

Conclusions 85

4. Energy Management (Economic Dispatch) 89

4.1 Generators 89

4.2 Economic analysis 93

4.3 Microgrid Layout 95

4.4 Problem Formulation 96

4.5 Economic dispatch 97

Conclusions: 102

5. Electrical Vehicle Charging Station (EVCS) and optimal location 104

5.1 Microgrid layout 104

5.2 Energy management strategy 106
5.3 Electrical vehicle charging station model 106
5.4 Analysis and optimal location of the EVCS 107
Conclusions 112
Summary 113
Appendix 115

1. Introduction

In recent times, there has been a shift in the outlook of the global energy scenario. There are considerable efforts being made in-order to shift from fossil fuels to renewable energy sources which is evident from the International Energy Agency (IEA) global energy review 2019 [1]. Over the last decade despite an increase in overall energy demand, the share of energy from coal in the energy mix has fallen by 2% from 2010 to 2019, the share of nuclear energy has fallen by 1% but the share of renewable energy has increased by 2%. The increase in the share of renewable energy is mainly due to significant growth in both solar Photovoltaics (PV) and wind power [1]. These contributions from renewable energy sources are mainly directed towards the power sector which represents a rapidly changing landscape in terms of sustainability and energy efficiency.

This momentum of renewable energy sources in the power sector is expected to increase due to the societal attitude to remain eco-friendly and the political will to do the same which is evident from the recently struck agreements such as the Paris agreement [2], the European green deal [3] and the Sustainable Development Goal 7 (SDG7) [4] which all promote the use of renewable energy sources, improved energy efficiency, e-mobility and the need to reduce CO2 and other greenhouse gases emissions in order to mitigate climate change.

In order to achieve the targets included in the numerous agreements mentioned above an efficient and sustainable power system has to be adopted. The Microgrid is one of the approaches of achieving such a power system. A crucial part of a microgrid is its Energy Management System (EMS) embedded with forecasters and optimizers. It is responsible for:

- Optimal resource allocation based on maximising a fixed objective.
- Account for the unpredictability of the renewable energy-based generators in its network.
- Maintenance of electrical network parameters such as voltage magnitude, frequency and others within specified limits according to standards
- Implement specified programs such as demand response and flexibility according to the needs of the local situation.

The importance and the need of this research work is to improve upon the current optimization and forecasting methods implemented in EMSs. The core thesis of this research work is as follows:

“Application of hybrid optimization algorithms embedded with deep learning forecasters enables optimal energy management of electrical networks. This can be used for microgrid optimal operation and to find the optimal location of microgrid elements such as an electrical vehicle charging station”

The thesis is realised as follows:

- Firstly, literature review is made of the existing approaches to optimization and forecasting.
- Secondly, new approaches to the same are proposed improving upon the existing approaches and providing an explanation as to why they will be better
- This is followed by implementation of the proposed approaches and application of the same to pre-existing electrical networks and related data

- Then the proposed methods are compared with the existing methods from the literature to establish superiority
- Finally, they are used to obtain optimal operational conditions of the microgrid based on real data from Wroclaw University of Science and Technology and optimal location of the EVCS in the microgrid.

The contributions of the thesis can be summarised as follows:

- The development of hybrid optimization algorithms which leverage the superior solution search capabilities of meta-heuristic algorithms and conventional optimization algorithms
- Deep learning neural network models to achieve better performance than conventional forecasting models for short term forecasting needs.
- An EMS tailored for optimal operational management of the Microgrid present at Wroclaw University of Science and Technology.
- Identification of an optimal operational scenario of the EVCS taking into account the interest of both the consumer and the power system. Then finding its optimal location in the microgrid.

The rest of this document is organised in the following manner. The introduction section introduces the microgrid concept and provides the literature review behind the existing methods for optimization and forecasting involved in power systems. Chapter 2 presents the proposed forecasting alternatives using deep learning models and a comparison with the existing ARIMA model. Chapter 3 presents optimal power flow as a first step towards energy management, introduces the hybrid optimization algorithms and then makes a comparison with the existing approaches. Chapter 4 presents energy management by way of economic dispatch of microgrids, it involves calculation of levelized cost of energy for all sources of energy present in the microgrid and aims to minimise the same during its operation. Chapter 5 investigates numerous charging scenarios for charging stations and identifies the best scenario based on which the optimal location of the charging station is decided in the microgrid. This is followed by the general conclusions and summary of the thesis.

1.1 Microgrid

The typical microgrid is defined in many ways. The most widely accepted definitions provided by the US department of energy [5] and Conseil International des Grandes Réseaux Électriques (CIGRÉ) [6] are as follows:

"A group of interconnected loads and distributed energy resources within clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid. A microgrid can connect and disconnect from the grid to enable it to operate in both grid-connected or island mode"

"Microgrids are electricity distribution systems containing loads and distributed energy resources, (such as distributed generators, storage devices, or controllable loads) that can be operated in a controlled, coordinated way either while connected to the main power network or while islanded"

While other definitions exist, they are similar to the definitions provided above. All emphasise on the following characteristics:

- The microgrid is visualised as a single controllable entity
- It is capable of operating in either a grid connected mode or isolated mode or both
- It is typically composed of numerous distributed energy resources
- In most cases it involves a high penetration of renewable energy resources
- The system is local in nature considering the needs of the society it is built for
- It is controllable and its operations can be co-ordinated

Microgrids enable the concept of active distribution networks and realise the implementation of smart grids. Active distribution networks are usually characterized by a higher penetration of Renewable Energy Sources (RES), Energy Storage Systems (ESS), bi-directional flow of power between the consumer and the Distribution System Operator (DSO), Demand Side Management (DSM), smart metering and it enables all players to make decisions in real-time that improves sustainability and profitability. It is characterised by the presence of modern Information and Communication Technology (ICT) and control systems. Smart grids on the other hand are defined as power networks that are capable of handling all stakeholders in an integrated manned. The stakeholders being the generators, system operators and consumers. This results in power being managed in a sustainable and economic manner. It is also characterised by modern ICT, control and monitoring systems [7].

The difference between microgrids and regular distribution networks with RES is mainly with regard to controlling the operations of the network. Microgrids quite often are equipped with Energy Management Systems (EMS) that are capable of managing both generators and loads in a way that is economical and sustainable. Moreover, they also interact in a smart manner with the upstream distribution network. They perform these duties while maintaining all electrical parameters of the network within their specified limits as required by power quality standards. This includes voltage magnitudes, voltage angles, harmonics and any other parameters of necessity. They provide decision making power to both the generators and consumers.

The typical microgrid layout is shown in Fig. 1. It consists of numerous components such as generators which could be solar PV panels, wind turbines, micro hydro plants, diesel generators, fuel cells etc. It includes different types of load demands such as critical load and common load. It includes different storage systems such as battery storage systems, pumped hydro storage systems, fly-wheel storage systems, compressed air storage systems etc. At the core of the microgrid is the central controller which is responsible for monitoring, co-ordinating and controlling the operations of the microgrid. It receives information from different components through a network of sensors and then moves into decision making depending upon a certain objective that is set by the user.

In general, the central controller or the EMS can be of two types, centralized and de-centralized [8]. In the centralized scheme the EMS receives information from all different components which includes meteorological data, load profiles of all consumers, information from energy storage systems, current output of generators and forecasted outputs of generators etc. based on which the EMS prepares a schedule for the microgrid and it's components and the decisions made are sent to the individual local controllers. On the other hand, in case of the de-centralized scheme the local controllers have more power, where each local controller sends a future possible generation or consumption schedule to the central controller which is analysed by the central controller and an optimal scheduling scheme is prepared and the decision is sent

to the local controllers. The local controllers may refuse to implement the decision made and bargain until the local or global objectives of the system are obtained.

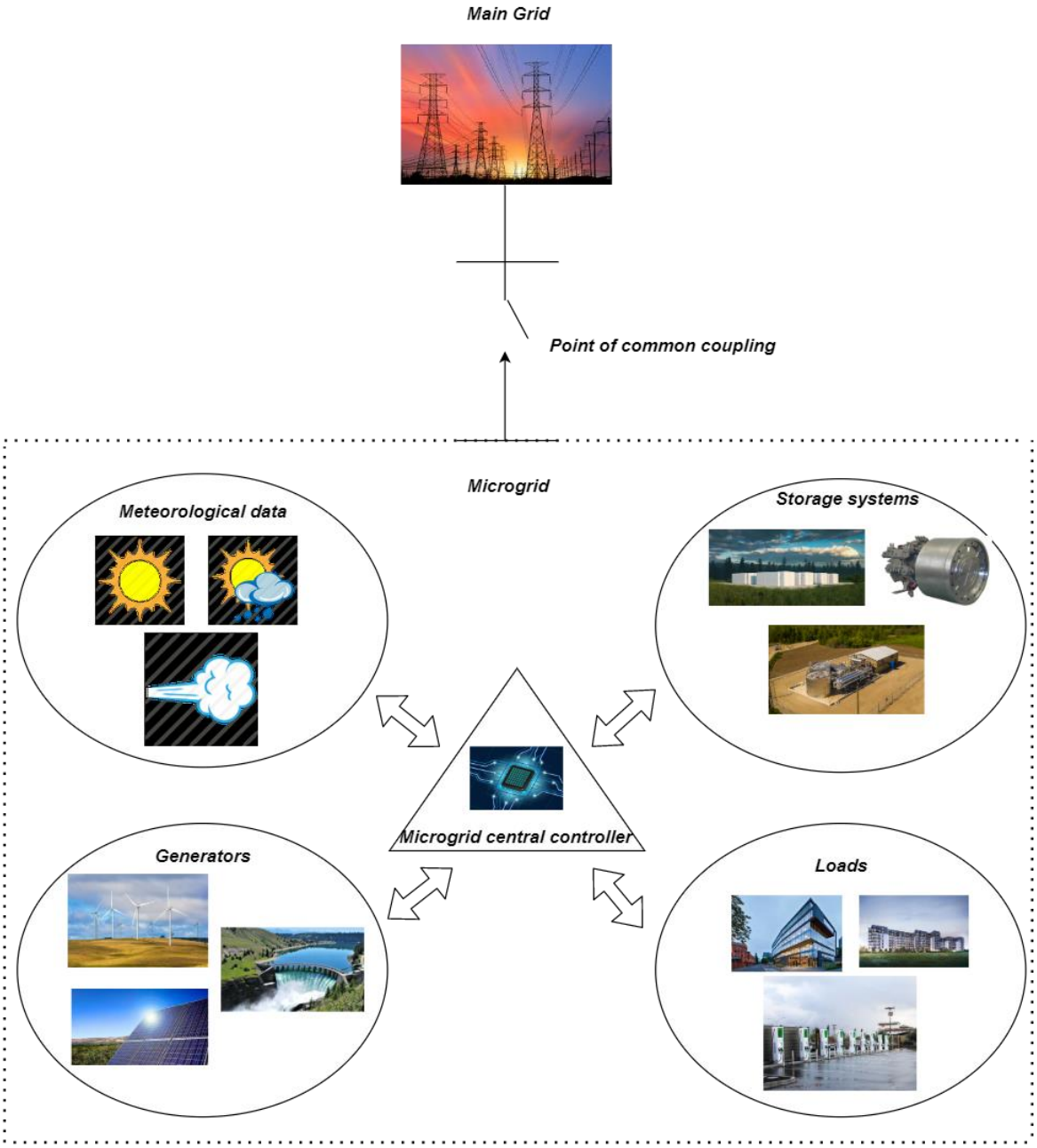

Figure 1 Typical microgrid layout

A typical EMS of a microgrid can be seen in Fig 2. As mentioned earlier the EMS receives information from all components present within the microgrid which is called as data acquisition. Once this is obtained the next step is data processing which involves standardisation of all data sets into a format that is applicable to the EMS. In this step the data is also manipulated as required which could involve normalization of the data, imputation replacing missing data, using transformations such as wavelets or Fourier's etc. This step is followed by forecasting which takes into account the meteorological data and past generation data in order to predict the future generation profiles of generators within the microgrid. It also involves load forecasting. These forecasted variables are then used by the optimization algorithm of the EMS to create a suitable schedule of the operations of the microgrid and its

components also known as Economic Dispatch (ED). Based on this schedule real-time control decisions are made wherein errors between the forecasted and actual values are accounted for by the EMS in real time.

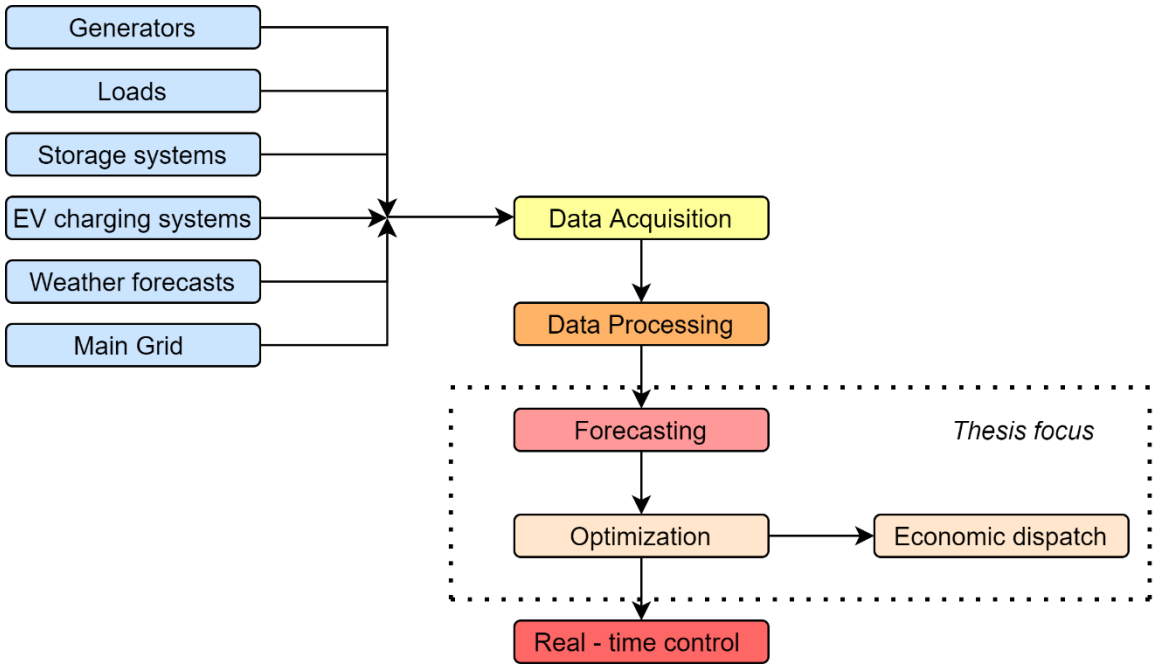

Figure 2 Typical energy management system

1.2 Forecasting

Forecasting in microgrids is mainly concerned with a few components. The generators whose output's depend upon meteorological factors and the load demand of the consumers which fluctuates regularly. This thesis is concerned with the forecasting of mainly the solar PV systems, individual wind turbines and load profiles of the consumer. The literature concerning the same is analysed below.

1.2.1 Solar power forecasts

The main challenges to wide spread adoption of PV systems are its variability and uncertainty [9]. In any given time-horizon from seconds to years the power output of a PV system is difficult to ascertain. In order to overcome this challenge numerous forecasting algorithms have been proposed. It is crucial to have forecasts of sufficient accuracy in order to deploy the PV systems in a large scale without threatening the stability of the power system and promote their participation in the electricity markets.

Broadly, forecasting algorithms for solar PV systems can be classified as physical or statistical. The physical forecasting model includes creating an appropriate model for the PV system taking into account not only the historical output values of the system but also other variables such as Irradiation (W/m^2), Wind speed (m/s), Ambient temperature (^oC) and PV Module temperature (^oC) which could affect the output of PV system. The selection of variables for the forecasting model are of utmost importance to achieve an acceptable accuracy [9]. The statistical model on the other hand takes into account only the historical values of the power output of the system. Both approaches have been used widely and the selection of an approach depends upon the availability of data and the local conditions of the PV system and needs.

The actual forecasting models used can be classified into 5 classes: time series models, regression models, Numerical Weather Predictions (NWP) based models, machine learning models and image-based models [10]. Each class is explained briefly.

Time series models: The models under this class fall mainly under 3 sub-classes. They are Auto Regressive Moving Average (ARMA) based models, Exponential Smoothing (ETS) models and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models. There are numerous models which are also a combination of the sub classes and in combination with other classes of forecasting models such as in combination with machine learning forecasting models. ARMA class of models such as auto regressive integrated moving average (ARIMA) , seasonal Auto Regressive Integrated Moving Average (SARIMA) are the most popular forecasting models used when it comes to solar PV output forecasts [10]. This is also because they serve as standards when it comes to testing and validating the results of other forecasting algorithms and have been in use for quite some time. They also have dedicated modules in most statistical software such as R or MATLAB making their application easier. The biggest drawback of such models is the fact that they are statistical and mainly only focus on the historical data of the output without considering factors such as irradiation, module temperature etc which affect the output of all PV systems.

Regression: Regression is defined as a statistical method of ascertaining the relationship between an output and an input variable or variables. When more than one input variable is used to determine the output, it is called Multiple Linear Regression (MLR). The regression model is written as follows: $Y = \beta_o + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$, where Y is the output/dependent term, X_k are the input/independent terms, β_0 is the constant term and β_k is the slope of each input/independent term, and ϵ is model's error, also known as residuals.

One of the simplest regression models is the first order Auto Regressive model (AR1) which is defined as $Y_t = \beta_o + \beta_1 Y_{t-1} + \epsilon$ Where, the forecasted output at *t* (Y_t) is calculated based only on the output value at the previous time step (Y_{t-1}). Considered to be a very naive method of creating MLR forecasting models is to include all possible variables that affect the output such as ambient temperature, module temperature, wind speed, humidity, cloud cover etc. Hence, for this method, as well as in the case of many others it is imperative to choose the right combination of input variables to make an accurate forecast.

Numerical weather prediction: NWP models are able to determine the irradiation fluxes across numerous levels of the atmosphere. Until recently the Global Horizontal Irradiance (GHI) and the Direct Normal Irradiance (DNI) were not provided by NWP models due to lack of appropriate application especially when compared to other variables such as temperature, humidity, precipitation and wind. This situation has now changed due to the enormous adoption of solar energy-based generators and devices which benefit from the availability of such information.

There are many NWP models that exist such as the Weather Research and Forecast (WRF) and especially the WRF – Solar [11] which is regarded as the first NWP model for forecasting solar power. Integrated forecasting system (IFS) developed by the European centre for medium-range forecasts. Advanced Regional Prediction Systems (ARPS) developed by the university of Oklahoma. The Global Environmental Multiscale (GEM) which is an integrated forecasting and data assimilation system created by Environment Canada. Global Forecast System (GFS) created by the National Centres for Environmental Prediction (NCEP) among others. All such models solve fundamental equations of fluid motion but their approach to the same might

differ. They might differ in terms if the numerical scheme used and in terms of models for sub-grid physical processes [10]. The usage of such NWP models for solar radiation simulations can be a challenging task and would require the need of domain expertise to use them appropriately.

Machine learning: One of the more popular approaches to forecasting in general and solar power forecasting is the use of Machine Learning (ML) approaches. ML methods include the use of neural networks such as Artificial Neural Networks (ANN) which incorporates Multi – Layer Perceptrons (MLPs) with back propagation. The general idea is to teach such neural network-based models the relationship between input and output variables using already available data which characterise a particular system. Over time there has been significant improvement in the structure and complexity of such neural networks which has led to the development of deep learning models which exhibit superior performance in terms of speed and accuracy of producing forecast models. This is because in comparison with the traditional ML models the deep learning models do not saturate in terms of performance when greater data and higher computational power is available [12], Another reason for their improved performance is the ability of such models to extract features from raw data utilising feature learning which helps the model to understand unknown representations within the input data [13].

Apart from ANNs and deep learning models there are numerous other ML methods which are used for different purposes within the field of forecasting. Support Vector Machines (SVMs) and K- nearest neighbours (kNN) are used for classification of data. Gradient boosting and random forests regressions represent ensemble learning methods for regression which combine the prediction from numerous forecasting models with the objective of producing more accurate forecasts. While the world of ML is too big to describe in a few paragraphs, all methods have a common approach to creating models which is to initially split the data into training and test sets. To fit a model according to their own working algorithm and validate the functioning of the model against the test set. All results are then evaluated using a common set of evaluation metrics such as the Root Mean Square Error (RMSE), BIAS, Mean Absolute Error (MAE) etc. in order to determine the superior models.

Image based models: The accuracy of models quite often can be improved with the use of satellite or sky imagery. Such images provide information regarding the approach of clouds beforehand which can be incorporated in the model improving its performance[10]. These images are used in tandem with other approaches mentioned above. The image containing data pertaining to the clouds is used to train regression models, deep learning models etc. These approaches extract data from the images only along a line or a sector upwind of the sun and apply regression or machine learning methods only at a select number of pixels in order to derive the GHI or DNI at the imaging device's location.

A few examples from the literature are as follows, [14] presents a model which is based on a Recurrent Neural Network (RNN) utilising Genetic Algorithm (GA) and Ensemble methods for day ahead forecasting for regional solar power plants in Thailand. The data utilised for model training includes historical data of the power generated by solar power plants. Weather data at the location of the power plants obtained using measuring devices and weather forecast data obtained from 3 free weather models which are PVwatts, Pysolar and NCEP forecasts respectively. PVwatts developed by the National Renewable Energy Laboratory (NREL) is able to provide information regarding solar irradiance upon entering the latitudinal and longitudinal co-ordinates. Pysolar which is a python library provides information regarding

solar irradiance, azimuth angle and altitude angle of the sun at all locations on earth. NCEP forecasts developed by National Centre for Environment Prediction (NCEP) provides 4 times a day weather forecasts every 6 hours. The evaluation criteria used for assessment of the results obtained was the RMSE. Due to utilisation of various data sources the number of variables taken into account are quite high which are the GHI, Diffuse Horizontal Irradiation (DHI), altitude degree, ambient temperature, module temperature, downward and upward short-wave radiations at the surface and finally solar irradiation. The results have indicated an improvement of 1.71 % with regard to the RMSE when compared with a single model instead of a combination of an RNN based on GA and Ensemble methods.

Forecasting concerned with the largest solar plant in Russia, the Sakmarskaya solar power plant can be found in [15] wherein a combined physical and statistical approach to forecasting has been adopted. The forecasts were carried out utilising solar radiation flux density, relative air humidity, wind speed and temperature. The meteorological variables were obtained from an onsite weather station. The comprehensiveness of the model arises from the detailed physical model which involves assessment of changing flux density of the solar radiation as it moves through the atmosphere. This is made possible by the use of a clearness index which accounts for absorption of solar radiation by various atmospheric elements such as the ozone layer, dust particles, clouds and water vapour. The evaluation metrics used in this study are the RMSE and the Mean Absolute Percentage Error (MAPE). The analysis of the forecasting model is quite rigorous and the authors have made concrete efforts to identify weak points that result in high accuracy of forecasts. In this manner recommendations have been made to improve the accuracy of the forecasts to an error value of 3-4 % of the output power.

A purely statistical approach to forecasting can be found in [16]. The study uses a novel Mycielski – Markov method in order to make forecasts. Mycielski refers to the signal processing involved and it is used to make forecasts in a deterministic way using recurrence of data points. Markov chain on the other hand is responsible for calculating the probabilities of different solar energy states and produce a forecast based on these probabilities. The data for this study was local and was taken from monocrystalline PV panels installed therein. The method is claimed to be novel since it accounts for both the deterministic component through Mycielski and the probabilistic component through Markov. The evaluation metrics used in this study are the RMSE, coefficient of determination, MAPE and Relative RMSE (RRMSE).

A novel analog ensemble approach to hourly day ahead forecasts is described in [17]. The model uses both statistical and physical data. It utilises freely available open source data (weather forecasts), measured data such as historical output data, irradiation, temperature and some astronomical data such as the declination angle and solar time. The approach was applied to a zone in South east Massachusetts to demonstrate its performance. The open data sources for weather forecasts were obtained from 3 widely available NWP models which are the North American Mesoscale Forecast System (NAM), the Global Forecast System (GFS) and the Short – Range Ensemble Forecast System (SREF). All of the above-mentioned models provide the GHI, DNI, DHI and ambient temperature. The NAM and GFS models are run 4 times a day, every 6 hours at 00, 06, 12 and 18 UTC time. The SREF is run every 6 hours at a 3-hour lag compared to the NAM and SREF at 03, 09, 15 and 21 UTC time. In order to download information from all the sources mentioned a web-crawler was deployed by the authors. Once all the data is collected, data cleaning is deployed to remove missing data and invalid data. The approach in its core has two algorithms which are the Earth Declination Angle Change (EDAC) limit algorithm and the Historical Day Change (HDC) limiting algorithm. In brief both the

algorithms have 3 steps that are: building a data set from past values that have similar declination angles and solar times, within this said data set identify the points most similar to the weather conditions of the time for which the forecast is to be made and predict the power at the time for which the forecast is to be made using the available information. The advantages of using such an approach is that it doesn't need detailed information regarding the PV system for which it used, it can be set up with minimum work effort and labour costs, much of the data used is available freely to the public and it is a novel approach to forecasting since it used declination angles and solar times. The evaluation metrics used were the Normalized MAE (NMAE) and Normalized RMSE (NRMSE). The results of the study were compared with three baseline models based on the Persistence model, the NWP model and a Support Vector Machine model (SVM). The results showed a significant improvement over all baseline models used with regard to both the NRMSE and the NMAE.

An improvised probabilistic method of forecasting solar power is described in [18]. The model is based upon a correlated weather scenario generation. The process utilises copula which is a popular method for ascertaining the dependency amongst variables. It is used for re-writing multivariate joint distributions in the form of univariate marginal distribution functions with a copula that defines the dependency amongst the variables. The steps involve fitting Probability Distribution Functions (PDFs) of the historical actual and historical forecasted weather data. This is then followed by obtaining the joint Conditional Distribution Functions (CDF) of the historical actual weather data utilising the forecasted historical weather data and Copula. Following this is the generation of numerous weather scenarios through Gibbs sampling. The process described above is utilised to produce numerous weather scenarios which help in obtaining probabilistic forecasts. In order to do so an initial deterministic forecasting process still has to be in place which is done in this article through a data-driven multi-layered multi-model ensemble machine learning system. The input variables used for this particular system include historical values of solar power output from the solar farms and weather data such as air temperature, solar radiation, solar zenith angle and wind speed obtained from the National Solar Radiation Database (NSRDB). The data has a 30-minute temporal resolution. Given the input data the forecasts are obtained using an ensemble wherein the first step involves using the input data to obtain forecasts from different machine learning algorithms such as random forests, gradient boosts, SVMs and neural networks. This is followed by a blending algorithm which utilizes the individual forecasts to produce a final forecast. The evaluation criteria used for the forecasts are the NRMSE and the NMAE for the initial deterministic forecasts and Pinball loss for the final probabilistic forecasts. For comparison of the results four benchmarking methods are used which are Quantile Regression (QR), fixed-date, shifted-date and bootstrap methods. The results have shown improvement over all other benchmarking techniques used.

1.2.2 Wind power forecasts

The challenges to wind power integration into power systems are similar to that of the solar PV systems which are variability and uncertainty. In order for increased wind power penetration, forecasting is deployed so that appropriate scheduling and reserve capacity can be planned by the power system operators and regulators [19].

Again, similar to the solar power forecasts, wind power forecasting can also be categorised under physical and statistical methods of forecasting. The physical methods take into account numerous physical factors affecting wind speed such as temperature, physical terrain of the surroundings, pressure and obstacles. Moreover, numerous such models employ NWP models

that solve conservation equations numerically at the location [19]. The statistical models are time-series models which are dependent on analysis of historical recorder data. It involves pattern identification, parameter characterization and model checking. The models can be categorized as Autoregressive (AR), Moving Averages (MA), ARMA and ARIMA models.

The forecasts can also be classified based on the time horizon. There are 4 categories namely, very short term ranging from a few seconds to 30 minutes, this time horizon is useful in electricity market clearings and wind turbine control. Short-term ranging from 30 minutes to 48/72 hours ahead which is useful in economic load dispatch and load increment/decrement decisions. Medium term ranging from 48/72 hours to 1 week ahead, these are useful for generator online/offline decisions, planning maintenance schedules and unit commitment problems. Long term ranging from 1 week to a year or more, this is useful during feasibility studies of wind farms and maintenance scheduling for optimal operational cost [20].

This thesis mainly focuses on forecasting at the short term. Hence, certain examples from the literature are as follows. [21] presents a comparison of popular short-term forecasting methods especially between conventional time-series methods – ARMA and intelligent methods which are ANNs and Adaptive Neuro-Fuzzy Inference Systems (ANFIS). The data used in the study were collected from the wind atlas of South Africa. It included wind speed measurements at various heights, wind direction data, atmospheric pressure, relative humidity and temperature. The results in this study have shown that for very short-term forecasting (10 – min ahead) the intelligent methods outperform the ARMA model and for the short-term forecasting (1- hour ahead) the ARMA model performs better. The evaluation criteria used in the study are the MAE and the RMSE.

An improved statistical approach to short term wind speed forecasting can be found in [22] wherein they have proposed the use of a Repeated Wavelet Transform – ARIMA (RWT – ARIMA) which has shown to be superior to both the ARIMA model, a hybrid Wavelet – ARIMA (WT – ARIMA) model and the persistence model. The performance of the model has been evaluated at different time horizons such as 1 min, 3 min, 7 min and 10 min. This article provides an in-depth analysis into the usage of wavelets, ARIMA and WT – ARIMA for the purposes of short-term wind speed forecasts. The evaluation criteria used in this study are the MAE, RMSE, MAPE and Sum of Square Error (SSE). It has been shown that the reason for the better performance of the (RWT – ARIMA) model is due to its ability to decompose the higher frequency time series into detailed coefficients and course approximations in addition to the low frequency time series decomposition. This way it is expected that the RWT- ARIMA will be able to capture all time variations in the wind speed data.

An intelligent approach to short term wind speed forecasting is presented in [23]. The approach used in this case is the Jaya – SVM model. It is different from the regular SVM approach in the fact that the input data is analysed and only the most representative features of the same are utilised by the model. The Jaya optimization algorithm is used to optimize the hyper parameters of the SVM. This particular optimization algorithm was chosen by the authors due to its nature of finding the optimal solution of a given problem while avoiding the inferior solutions. Moreover, its simplistic approach avoiding the need to use control parameters and work only on the basis of common parameters such as the number of iterations and population size merit its use. The results of using this model have been compared with the same data used on 7 other models which are the deep belief network, multi-layer perceptron regression model, least absolute shrinkage and selection operator, gaussian process regression, extreme gradient boosting, granular computing method and stacked sparse autoencoder. The data used in this

study includes wind speed data alone collected in Jilin, China. The evaluation parameters used include the MAPE, R^2, MAPE and MAE. The results have shown the Jaya – SVM model outperform all the above mentioned 7 algorithms with regard to all evaluation parameters. Moreover, it is also shown that the model displayed higher reliability within the same confidence levels. It is believed that this improvement in performance of the SVM is due to the Jaya algorithm.

A comprehensive approach to forecasting has been adopted in [24] wherein the approach involves considerable data pre-processing, a modified multi – objective optimization algorithm and various forecasting models. A novel data pre-processing technique called Improved Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (ICEEMDAN) has been used in order to overcome the disadvantages of more regularly used data pre-processing techniques such as the Empirical Mode Decomposition (EMD) and the Ensemble EMD (EEMD). EMD has weaknesses such as mode-mixing and EEMD has problems with residual noise. ICEEMDAN is able to overcome such problems, remove noise fluctuations and extract the main features within the wind speed data. The prediction model used is a combination of both conventional and intelligent algorithms, they are the Back Propagation Neural Network (BPNN), Ensemble Neural Network (ENN), Extreme Learning Machine (ELM), Generalized Regression Neural Network (GRNN) and the ARIMA model. The reason for using both types of models is because the ARIMA is efficient in characterising linearity and the intelligent algorithms are good at dealing with non-linear characteristics. The weight coefficients of all above mentioned models were determined by a modified multi – objective dragonfly algorithm. The original algorithm was modified by adding to it an exponential step size strategy and an elite reverse learning strategy. This enables the algorithm to hop out of local optimum solutions when entrapment happens. The multi – objective optimization algorithm used focuses on two objectives which is accuracy of the result and stability. The data used in this study is the wind speed data obtained at an interval of 10 minutes from 4 adjacent wind farms located in Shandong, China. The evaluation metrics used in this study are the MAE, SSE, MAPE and the RMSE. The results of the study were compared with forecasting results from individual models of ARIMA, ELM, ENN, BPNN and GRNN. The proposed model has outperformed all the above-mentioned individual models.

A hybrid forecasting model based on intelligent algorithms is presented in [25]. The model is composed in total of 4 different modules which are the Crow Search Algorithm (CSA), Long Short-Term Memory (LSTM) deep learning networks, Wavelet Transform (WT) and Feature Selection (FS) depending upon entropy and Mutual Information (MI). The procedure of the model is as follows, first the erratic and fluctuating characteristics of the time series data is removed using WT which splits the wind speed data used into 4 sub – series. This is followed by FS which discards inputs that are redundant and selects the most appropriate inputs for forecasting. This not only contributes towards increased accuracy of the model but also improves the run time. The outputs from this step are then used to train an LSTM model which makes the forecasts. The LSTM is optimized using the CSA. The data used in this study was obtained from two different sites, one from Spain and the other from Iran. In the former the data was recorded every 10 minutes which was then converted to hourly wind speed data and from the latter the data available was hourly. The evaluation criteria used in this study are the RMSE, MAE and MAPE. The results of the study have been compared with various other models such as the WT-FS-LSTM model, MLP model, WT-FS-MLP model, basic LSTM model, FS-LSTM model, WT-LSTM model, WT-FS-LSTM model and WT-FS-LSTM model optimized by particle swarm optimization (PSO) and it had outperformed them all.

A turbine specific approach to forecasting wind speeds is presented in [26]. In this approach firstly, data is collected at hub height from a cluster of turbines and then two different stochastic models are built in-order to capture two different characteristics that are responsible for wind propagation in a wind farm which are the temporal and spatial dependencies along with the high-magnitude and high-frequency erratic fluctuations that have an impact on model accuracy. The two models are then coupled together to produce probabilistic forecasts of wind-speeds for each turbine. These forecasts are then used to make wind power forecasts using the power curves of the turbines. The first characteristic which are the temporal and spatial characteristics are modelled as a gaussian process with a specific non-separable covariance model. This is different in comparison with regular approaches which usually tend to separate the models for spatial and temporal dependencies. This is followed by adding another gaussian process which is used to capture the second characteristic which is the high – frequency and high – magnitude variations. The data used in this study is obtained from a wind farm located in the United States. It includes wind speed and wind power data that are recorded hourly at a height of 100 m for about 200 turbines. The wind power data is scaled in the range of [0,1] and wind direction measurements are made in three different locations of the wind farm. The wind power curves are created using the binning method. This approach is compared with several other models such as the persistence model, Regime – switching ARIMA model, spatio – temporal gaussian process, calibrated regime-switching gaussian process and Farm-level ARIMA model. The evaluation criteria used in this study includes the RMSE based on forecast per time horizon and RMSE based on forecast per location. There is an overall improvement of the proposed method despite the benchmark models performing satisfactorily at certain instances.

1.2.3 Load forecasting

The need for load forecasting depends from scenario to scenario[27]. For example, short-term forecasting of load is crucial for power balancing in the electrical grid, in the power exchanges and control of power plants. Long-term forecasting on the other hand is useful in capacity planning and expansion, creating long term maintenance schedules, making capital investments and carrying out revenue analysis. Moreover, load forecasting can also be carried out over varying electrical boundaries, it can be made for a building, a local neighbourhood, a district or a country.

There is a plethora of methods when it comes to load forecasting but in a broad manner, they can be classified under 4 classes as explained below.

Artificial neural networks: The conventional time series models that are often employed are unable to characterize the complex non-linear relationships that exist within data [27]. Moreover, at times they produce results that are unsatisfactory. With the increase in computational power availability and improvement of neural network-based models, ANNs and deep learning-based models are becoming increasingly popular. These class of models are based on the functioning of the human brain. ANNs consist of a network of neurons in many different arrangements and are interconnected to one another wherein they can send and receive signals that are normalized. The training process for any ANN involves characterization of weights which can be seen as the threads that connect these neurons. They are known to be quite powerful in patter recognition applications. One of the most popular and commonly used ANN is the MLP model. It is based on the theory of back-propagation wherein the error between the output value of the ANN and the actual value helps determine the weight values of the neural network. ANNs are quite often data-driven and unlike time series models which saturate in performance, ANNs improve with increased data availability and increased

computational power. The drawbacks of ANNs and deep learning models are that, they are essentially black boxes which means it is impossible to understand the exact relationship by which they relate the inputs and outputs, they have high training times, they are computationally expensive, they are data intensive, as in they need a certain minimum amount of data to function desirably and the models may never converge to a satisfactory relationship at times.

Time-series models: Some of the most popular methods of forecasting belong to this class of models, as already explained in the previous sections it includes the ARMA and ARIMA set of models amongst a few others. The ARMA and ARIMA were introduced by George Box and Gwilym Jenkins in 1970 [27]. The difference between ARMA and ARIMA is the differencing of the time series data which is done in ARIMA and not in ARMA. This is because these class of models are applicable only to stationary signals. Autoregression (AR) can be defined as a linear relationship between a current output value and past values whereas a Moving Average (MA) can be defined as a linear relationship between the present value and past white noise or error values. The models can also be adapted to account for seasonality and such an adaptation is called as Seasonal ARIMA or a SARIMA model. In general, the Box – Jenkins class of methods are adaptable and seasonal. The drawbacks of such methods are that they are unable to characterize complex non-linearity present in the data, they are not suitable for long term forecasting and at times needs strong expertise in order to understand the underlying statistical data.

Bottom up end-use approach: This approach is called so because it involves creating a representative model of the entire system by aggregating the demand of the individual components of the said system. The most commonly used bottom up approach is the Capasso model. This model calculates the probability of an appliance working throughout the day (at every time step) taking into consideration the appliance's and the members of the household's characteristics. This probability is then calibrated based on whether the activity attached to the appliance can be performed by one person or more and if the activity can be carried out along with another activity. Once the final probability is attained, socio-economic criteria is used in-order to decide the penetration of various appliances of a given household. The power requirements and duration associated with each appliance is calculated and it is ascertained if those parameters can be correlated with a specific activity and building. For this to be possible the minimum duration of every appliance has to fit the probability of the activity in which it is involved and the required power has to be lower than P_{limit} which is the maximum load demand of the household taking into account that other appliances might also be functioning at the same time instance. This leads to construction of an appliance and household daily load profile. This model is quite comprehensive and accounts for behavioural differences amongst different households. It characterizes the relationship between the members of a household and their appliances. The disadvantages of this approach is the massive exercise to be carried out in terms of surveys and accumulation of data, the lack of data in terms of behaviour of the consumers in the long run and the fact that the model assumes a linear relationship between end use and load consumption.

Support vector machine: SVMs were originally introduced by Vladimir Vapnik and later the soft margin classifier was put forth by Vapnik and Cortes. SVMs were initially used for pattern classification problems such as facial recognition and text classification. In 1995 this class of algorithms was extended to also include SVM regression which led to its application to time series forecasting. SVMs create a hyperplane which in a higher dimensional feature space

classify incoming observations into different subsets. In practice however it may not be possible to classify real data easily. Keeping this in mind a soft margin classifier was introduced. SVM is useful in the fact that it is able to create non-linear decision boundaries, this is possible since it is able to linearize such boundaries by using appropriate transformations on a higher dimensional feature space. The drawback in this process is the high computational power required to make calculations in a higher dimensional space which makes proper selection of hyper-parameters crucial. It is possible to improve the computational efficiency by using something called a "kernel trick". These are in fact functions which utilise the inner products between observations instead of the observations themselves. Thus, it changes how similarity between two observations is calculated, making it more flexible and possible to solve a non linear problem linearly in a higher dimensional space.

Regression: Regression here is the same as explained before in the sub-section of solar forecasting. It is a popular statistical method used to establish a relationship between a dependent variable and other independent variables. The most common regression methods used are the linear and polynomial regression methods though other exist.

A few examples from the literature are as follows:

A unique approach to load forecasting is described in [28]. For all practical purposes it can be viewed as top down approach. The reasoning behind such a model is that the geographical area of a bulk power system can be massive. The study conducted in China, takes into account a large geographical area and creates sub districts within the same region and attempts to establish different forecasting models for each of these sub districts based on their individual meteorological conditions and historical consumption data. The individual models for the sub-districts are called subnet models and every such model is created using the Mining Default Rules Based on Rough sets (MDRBR) algorithm. The MDRBR is an approach which uses the mining policy in a bottom to top approach. Every rule with every condition is used to form different layers one by one. For example, given 3 rules with 3 conditions, a layer is first created with one rule having one condition and when this is fulfilled the next layer is built with the second rule and second condition and when this is fulfilled a third one is built. In this manner the procedure keeps moving until all rules containing all conditions are satisfied. The rules are in fact simple "IF" statements such as, if the temperature is above 35^0C then the load is high. The area under investigation for the study is divided into 3 subnets which are the Beijing subnet, Tianjin subnet and Tangshan subnet. The total number of attributes considered include meteorological factors such as temperature, humidity, rate of change of temperature, rate of change of humidity and the characteristic of the day, if it is a holiday or a weekday and historical load consumption data. The evaluation criteria of the study used is "Accuracy", a term defined by the national electric power dispatch centre of china. Similar to other metrics it is dependent upon the error value between predicted and actual value.

A comprehensive study regarding the application of deep learning models for hour ahead and day-ahead load forecasting is presented in [29]. The study involves modelling a sophisticated model involving data pre-processing, feature selection and hyperparameter fine-tuning. The data pre-processing used is quite standard involving imputation, detection of outliers and normalization of the data. Feature selection involves optimal choosing of features needed to train the model, this is done so that redundant data can be avoided which reduces computational burden and improves accuracy. The study has used Pearson correlation and analysis of the same to select the optimal features needed. 9 deep learning models are used by the study which include the regular LSTM, BiLSTM, BiLSTM with attention, CNN + BiLSTM, CNN + LSTM,

Encoder – Decoder LSTM, ConvBiLSTM, BiLSTM with attention and ConvLSTM. The performance of all the above-mentioned algorithms are dependent on the hyperparameters that define them, this study has optimized the selection of a few of the hyperparameters which are in fact common to all the models. They are the optimizer, the number of layers and neurons, the best learning rate, the activation function and the number of epochs and batch size. The evaluation metrics used for the study include the RMSE, MAPE, Coefficient of Variance (CV) and a new Root Mean Square Logarithmic Error (RMSLE). RMSLE was introduced as a new metric because the RMSE was not a suitable candidate for comparison of results in this study since it involves a number of buildings with different load capacities and the buildings with higher load capacities will automatically see their RMSE values ballooned. The forecasting has been carried out for 5 different types of buildings in 5 different places. The building types are a grocery store, a school, an office, a research laboratory and an academic building. The buildings are spread across Bangkok, Thailand, Hyderabad, India, Virginia, Massachusetts and New York in the USA. Such a wide variety of buildings from different locations were chosen in order to show the general applicability of the forecasting method irrespective of the building type and the place. The data used in the study include meteorological data such as temperature, humidity, air pressure, wind speed and solar irradiation and load consumption data of all buildings. While a number of input variables exist, feature selection ensures that the optimal number of variables are chosen. The study also investigates into the performance of Ensemble learning based clustering wherein 2-3 clusters of different combinations of features were used to train and evaluate models. The best results were achieved by the ensemble of a cluster consisting of historical load data and hourly meteorological data along with a BiLSTM with attention network for low load conditions and the same cluster with an LSTM with attention for high load conditions. Finally, a sensitivity analysis was also carried out in-order to understand the effect of data with different resolutions on the forecasting model. It was concluded that data with a higher resolution does improve the forecasting model but it also increases the computational burden in terms of time and memory.

Load forecasting using the ARIMA model has been adopted in [30]. The study presents a short-term load forecasting model for a 400 kV substation. The study has also investigated the effects of different data sets with different sizes on the accuracy of the forecasting model. The software used for this study is R, which is a widely used tool in data science and data analytics. The Auto.arima function in R is used to fit the data and make the forecasts. They have considered two cases wherein the accuracy of forecasting has been evaluated when the input data contains data related to temperature and when the input data does not contain any information related to the temperature. The evaluation criteria used in this study is the RMSE and it was evaluated for both types of data sets (with and without temperature) and for two different dataset lengths (one day and one week). It was found that the RMSE was the lowest when the dataset considered had included data regarding temperature and the data points available were for the whole week making it the most accurate model.

Load demand while being stochastic for whole of the year still follows a seasonal and daily trend which is also influenced by meteorological factors. It is this combination of data that form the basis of most forecasting models. During a year there are certain days (festivals, events etc.) during which the regularly used forecasting models come up short and make highly erroneous forecasts. In order to make accurate forecasts for these special days a model is presented in [31] which is based on SVM hybridized with a novel Grey Wolf Optimizer (GWO) algorithm to forecast the load for 3 different festival days in Assam, India. The need for such a forecasting model also comes from the fact that the particular region is energy scarce and

during the festival period the energy demand increases so rapidly that in-order to maintain operations, accurate forecasting methods are required. The data used in this study for model training is mainly historical hourly data for the festival days since the data in any given hour of the holidays under study change rapidly due to the many rituals that are scheduled for certain parts of the day. Moreover, from observation it has been noticed that every calendar year during these days the demand increases due to the addition of new electrical appliances with different characteristics and ratings. In order to account for this change an hourly annual demand growth of 4.4% - 7.6% has been added from the previous calendar year. Apart from this data the ambient temperature forecast is also utilised by the model which is taken from an external service provider of such data. The GWO which is based on the grey wolfs' social hierarchy and hunting pattern is used to optimize and find the best combination of parameters for the SVM. The evaluation metric used in this study is the MAPE and the results are compared with four other methods which are the SVM, ANN, PSO-SVM and GA-SVM. The proposed method has been shown to be faster, reliable and more accurate than the other algorithms.

Load forecasting for multiple households using a multivariate model utilising past consumption values, socio-economic data and meteorological data is presented in [32]. The study utilises Bayesian Networks (BNs) defined as a probabilistic graphical model to perform the forecasting. It is defined as a directed acyclic graph wherein the model depends on the graph and the conditional probabilities, both of which have to be learnt. The learning of the graph can be constraint dependent, score dependent or a combination of the two. The constraint-based approach utilises independent tests whereas the score-based approach uses heuristics to build structures that are evaluated using metrics such as the Akaike Information Criteria (AIC) or Bayesian Information Criteria (BIC). The final model includes the part described above and a quantitative part. Once that is obtained it is possible to calculate the conditional probability distribution function of an unobserved variable given certain observed variable states. The data utilised in the study is publicly accessible and it is the Irish smart meter data which is available due to a project executed by the Irish Commission for Energy Regulation. The data is comprehensive and is obtained from 5000 Irish homes for a period of 17 months with a sampling rate of 30 minutes. The data captured is active power consumption. Along with this there is also data from questionnaires regarding the number of household occupants, customer classification, tariff schemes, tariff incentive stimulus and socio-economic data such as the social class of the household based on the occupation of the bread winner, the electricity used for cooking and house heating. The data once obtained was pre-processed, initially a filter was used so that data pertaining to only those households which also filled the questionnaires regarding their socio-economic status would be used. This resulted in a total of 929 households. Once this was done 59 evenly spaced dates were chosen and from every such date two data sets were determined, one to learn the forecasting model and the and the other to test it. This resulted in a total of 59 pairs of learning and test data sets. During this categorization any missing values were removed. Next, the power consumption values were normalized by the individual household's average consumption. Then the data was also discretized using a quantile-based discretization after which the temporal associations within the data were ascertained using Mutual Information (MI). This MI was in fact used to choose how many past values of power consumption should be used to train the BN model during the training process. The temperature values representing the meteorological data used were also discretized. Apart from the above-mentioned data, the time index and the questionnaire responses were used to train the BN model. The evaluation criteria used in this study are the NRMSE, the MAE, the Median Absolute Error (MedAE) and the Mean Arctangent Absolute Percentage Error

(MAAPE). The model presented was compared in performance with the persistence model, the ANN model and Hidden Markov Model (HMM) and shown to be better.

1.3 Optimization

A crucial economical and technical tool regarding power systems is the Optimal Power Flow (OPF). It is used for the purposes of analysing, operations planning and future expansion planning in power systems [33] . Over the last decade or so this aspect of the power system has been highly researched and numerous different methodologies and approaches have been experimented with. OPF can be used to set control variables of the power system at specific times so that the system under consideration is able to achieve a particular objective set by the OPF user. It also ensures that the power system is able to maintain operational limits with regards to variables such as voltage, frequency, line limits etc.

OPF was first introduced by Carpentier in 1962 [33]. It can be defined as a non-convex and non-linear problem that has an objective function which must be fulfilled to the best of possibilities about a set of equality and inequality functions. In short, OPF is about controlling the power flow in a power network while simultaneously trying to achieve an objective that it is programmed to achieve [34]. It does so while maintaining all electrical parameters of the system within their operational limits.

The OPF problem has numerous extensions some of which are described below:

Static OPF: This OPF problem is about achieving the best objective function value possible while handling all constraints in a single time step or in a particular moment in time [35].

Dynamic OPF: It can be defined as static OPF applied to multiple time steps or periods. It determines the optimal operational state of the power system over a time horizon [36].

Transient stability-constrained OPF: This is the type of an OPF problem that takes into account both the static and dynamic nature of the power system and power system stability concerns as constraints. The system is able to manage severe contingencies in this case [37].

Security-constrained OPF: The security-constrained OPF is an extension wherein the regular OPF (static or dynamic) is solved along with a set of contingencies defined for the power system [38].

Deterministic OPF: Much of the OPF studies carried out are deterministic in nature, as in, they do not consider uncertainties associated with the components and operational variables of the power system [39].

Stochastic OPF: This OPF in contrast to the deterministic OPF considers the uncertainties associated with the components and operational variables of the power system. It is in fact associated with the constraints and the objective functions of the OPF problem defined. The resulting optimized control actions obtained by solving the OPF problem also contain information regarding their uncertainties [40].

Probabilistic OPF: This OPF approach works with PDFs of loads, renewable generators and other uncertain parameters of the power system and generates PDFs of the dependent variables and results. Such PDFs can be generated using the Monte-Carlo method [41], customized Gaussian mixture model etc.

AC OPF: This type of an OPF problem is based on AC power networks and is more representative of the conventional power system in place across the world. It is modelled on the natural characteristics of power flow in the power system [42].

DC OPF: This type of an OPF problem is a simplification of the AC OPF wherein line losses and reactive power flows are not considered [43].

There are many different methodologies by which an OPF problem can be solved. In general, they can be classified in to two different categories namely conventional and meta-heuristic optimization algorithms [44]. Examples from both categories are presented below.

1.3.1 Conventional optimization algorithms

Linear Programming: The power system and it's functioning in terms of power flow and OPF is described by a set of equations that are non-linear in nature. Quite often the problem can be linearized and therefore simplified so that it is easier to obtain a solution. This method is the linear programming method and it is widely used, reliable and has high success at converging to a solution. The disadvantages of the method is that it may result in solutions that are sub-optimal, entrapment in the local minimum solution and the possibility of errors while calculating system constraints. A few examples of the same can be found in [45], [46].

Non-Linear Programming: The non-linear programming does not involve linearization of the problem and it is capable of handling non-linear functions along with the system constraints. These methods use a reduced gradient, Lagrange multiplier methods and it is possible to implement them on a large power system. Moreover, it is expected that the results will be an improvement from the linear programming approach as entrapment in the local minimum and sub-optimal solutions can be avoided. However, It has been observed that certain non-linear linear formulations of the problem do not consider certain system components or constraints which can be seen as it's drawback [47].

Quadratic Programming: In this formulation, the objective function of the problem is presented as a quadratic function whereas the constraints are still maintained as linear functions. This method yields quick results along with improved solutions, an example of the same can be found in [48].

Newton's method: The Newton's method has been used widely, especially for solving the power flow problem. The method utilises gradients and decomposition enabling quick convergence to a solution. The negative aspect of this method could be the entrapment of the solution at a local minima and subsequently sub-optimal solutions. The application of this method for power flow calculations is presented later in the thesis in detail [49].

Interior Point method (IP): This method used for solving OPF problems provides quick convergence and improved solutions. The application of this method to non-linear problems has enabled its application in solving the OPF problem. The drawbacks of this method also include local entrapment of the solution and its suboptimal nature, an example of the same can be found in [50] .

Distributed and Parallel OPF (DPOPF): This OPF approach utilises processing units at every bus of the power system in consideration. This is more applicable to a smart grid with a high penetration of IoT. Utilising the processing units, the approach minimises the optimization sub-problem at every bus. When such an approach is utilised in microgrids that are islanded it is called distributed OPF (DOPF). The method is usually deployed with a control algorithm in

order to carry out synchronized computations when the data arrival in asynchronous. It is quite useful in deregulated power systems when the uncertainty increases. Hence, it is an appropriate approach for a network consisting of renewables that are uncertain in nature such as microgrids, an example of the same can be found in [33].

Alternating Direction Method of Multipliers (ADMM): In this method, sub-problems of the optimization problem are solved by each local controller which are present within the power network. There is exchange of information between the local controller and the main controller and between the local controller and other local controllers. The control action decided by the local controllers is evaluated by the central controller and suggestions are sent back which may or may not be implemented by the local controller. The local controller has the power and the possibility to contradict the main controller until a suitable control action is implemented, an example of the same can be found in [51].

1.3.2 Meta-heuristic optimization algorithms

The meta-heuristic algorithms address some of the concerns mentioned above. They are able to produce results where in there is a higher chance that the solution doesn't get trapped in the local minimum solution compared to the conventional algorithms and that it is as close to the global minimum as possible [35]. It also ensures that there is no need to simply the objective functions and the constraints by means of relaxations and linearization since they can find solutions to complex, non-convex, mixed integer and non-linear problems [35]. Certain disadvantages of this approach is that their results are not always reliable since they are unable to reproduce the same objective function value at every run. Moreover, the algorithms take longer time to converge to a solution compared to the traditional conventional algorithms.

Some of the popular meta-heuristic approaches are described below:

Genetic Algorithm (GA): One of the first meta-heuristic approaches to be developed was the Genetic Algorithm. It can be described as a search technique based on the idea of natural selection and genetics such as mutations, cross-over, selection and inheritance [52]. Like many other meta-heuristics also known as intelligent algorithms, the GA begins the search process by creating a set of possible solutions referred to as the population. The goodness or quality of each solution depends on the objective function value obtained as result of plugging the solution into the objective function, closer the value is to the global optimum solution better is its quality. Henceforth, quality or goodness of a solution will be called as fitness. Every generation or iteration for which the algorithm is run, the GA tries to improve the fitness of the population in its disposal using selection, mutation and crossover. Selection is a process by which moving from one generation to another the GA ensures that the solution candidates with a better fitness have a higher probability to move on to the next generation. Crossover is a process by which there is some exchange of information between the solution candidates with the chance of producing solution candidates with better fitness in the next generation. Mutation is defined as a random act by which certain information within the population (solution candidates) is changed with the hope of obtaining a candidate with a better fitness. In general the advantages of the GA can be summarised as having a higher potential of finding the global optimal solution irrespective of the nature of the objective function, it does not need calculation of derivatives, it can handle both discrete and continuous variables and it is not very sensitive to bad initial solution guesses. The disadvantages include the high computational burden associated in running the algorithm in terms of time and memory, inaccuracy of solutions in certain cases and it is possible for the GA to be stuck in local optimum in certain cases.

Simulated Annealing (SA): This approach can be defined as an iterative process which is based on the annealing process in metallurgy which is based on heating and then gradual cooling of a metal in order to increase crystal size thereby reducing defects. The process begins as usual by creating a candidate solution and setting a positive temperature. Then gradual and controlled cooling is implemented. As one moves from one temperature level to another the probability of a good solution candidate being accepted from the solution space remains high while the probability of a worser solution candidate being accepted from the solution space is low. During every temperature level the algorithm randomly selects a new solution candidate close to the existing best solution, evaluates its fitness and takes a decision to move on it or not based on the probabilities presented to it on that temperature level. The advantages of this method are that it is easy to implement, it is quite robust and is very suitable for combinatorial optimization problems [52]. The disadvantages are that it could have a large computational time, the choice of the final solution could depend on the initial solution that is selected and there is a possibility of entrapment onto a local minimum.

Particle Swarm Optimisation (PSO): This approach to optimization involves mimicking the swarm behaviour of a flock of birds or a school of fishes wherein the individual members (possible solutions) move around their neighbourhood (search space of solutions) searching for the optimal solution. The process begins as in the case of GA wherein an arbitrary solution is first chosen and then it is improved over generations. In every iteration or generation, the fitness of the particles is determined and this is done by means of their location in the search space. The search for a better solution is driven by the location of already known best solutions of the particles themselves and that of the entire population. The iterations carry on in this manner until a suitable solution is arrived at [52]. In an ideal situation this happens with the convergence of all particles onto a solution which would be the global optimum solution but this is not guaranteed. The advantages of this method are that it is simple to implement, it has only a few parameters which determine the manner by which a solution is arrived at, it can be run in parallel computation, it is robust, quick convergence to a solution meaning faster run times, has a higher probability than most algorithms to find the global optimum solution, it is useful in finding results to problems for which formulating mathematical models might be difficult [52]. The disadvantages are that even though the parameters are a few, it could be difficult to find the optimal setting of the parameters for the given problem, it is still possible that the solution may not converge to a global minimum.

Tabu Search (TS): This optimization approach can be described in some manner as a stochastic optimization approach since transitional rules from one solution to a better solution are probabilistic rather than deterministic [53]. The problem begins as usual with a set of current solutions, this is then improved by a process in which trial solutions are generated in the neighbourhood of the current solutions. While generating trial solutions certain conditions are imposed so that regression of solutions is avoided. These conditions are frequently updated in what is called a tabu list. This prevents the algorithm from reverting back to older solutions that could represent local optimal solutions or simply bad solutions. The algorithm also has a mechanism to override the tabu list by something called an aspiration criterion. This ensures that certain moves can be made even if they are in the tabu list if they would yield to better solutions. This is incorporated to make sure that the algorithm is not trapped in a local minimum solution. The advantages of this algorithm are that it can be robust and is easy to implement, also it is known to provide suitable solutions to many combinatorial optimization problems [52]. The disadvantages on the other hand are that, it has a high computational time, possibility of entrapment in a local minimum, the algorithm is sensitive to initial choice of a solution and

that there is no information regarding the upper bounds of how long it could take to arrive at a solution.

Harmony Search (HS): Another algorithm which unlike others based on natural processes is the harmony search algorithm. This approach is based upon the technique employed by musicians to improve the harmony of their instruments. The process begins with an arbitrary set of solutions which is improved over iterations. The process of creating new solutions in this approach involves considering all the solutions from the previous iteration instead of just two as in the case of GA. The HS is relatively new met-heuristic algorithm and has shown to be effective. The advantages of this method are that there is no need to mention the initial solution from which the improvements are made, high possibility of convergence and it can handle both discrete and continuous variables. The disadvantages include the possibility of high number of iterations and may have a lot of iterations without any improvement in the results [52].

Apart from the algorithms mentioned above there are numerous meta-heuristic optimization algorithms used such as the Artificial Bee Colony (ABC), Ant Colony Optimization (ACO) described later in this thesis, Cuckoo Search Algorithm (CSA), Firefly Algorithm (FA), Shuffled Frog Leaping Algorithm (SFLA), Shuffled Bat Algorithm (SBA), Biogeography Based Optimization (BBO) etc.

A few of such approaches applied to OPF and ED problems are shown below.

The application of Gravity Search Algorithm (GSA) to the OPF problem is shown in [54]. The algorithm is used in this case to determine the optimal control settings of the OPF problem. The performance of the GSA is evaluated on two test cases, the IEEE 30 bus and the IEEE 57 bus systems. Also, the performance of the algorithm with regard to different objective functions is made with other popular meta-heuristic optimization algorithms found in the literature. The GSA begins the process by considering numerous agents/objects/candidate solutions, every such agent has a mass of its own which represents the fitness of the solution, higher the fitness, higher the mass. Then the algorithm works based on Newton's laws of physics and the agents slowly move towards the objective having the greatest mass or fitness thus converging towards the global optimum solution. During the course of convergence, the lighter mass agents with lower fitness move faster while the solutions with a better fitness move slowly. The objective functions considered in this case were the quadratic cost function, voltage stability improvement, voltage profile improvement, voltage stability during contingencies, piece wise quadratic function and the quadratic cost function with valve point loadings. The other algorithms with which it was compared with includes the BBO, PSO, Improved GA, Gradient Method (GM), Enhanced GA, Evolving Ant Direction Differential Evolution (EADDE), Evolving Ant Direction with Hybrid Differential Evolution (EADHDE) and Differential Evolution (DE). The GA had outperformed all the above-mentioned algorithms.

A hybrid approach utilising the combination of an Imperialist Competitive Algorithm (ICA) and the teaching learning optimization algorithm for optimal power flow is described in [55]. This approach was adopted due to the nature of ICA to converge to the local optimal solution often despite being a powerful optimization algorithm. Hence, to promote the search for a solution around the global optimal solution it was combined with the Teaching Learning Algorithm (TLA). The performance of the algorithm was evaluated on the IEEE 30 bus and the IEEE 57 bus system with different objective functions. The ICA is based on the imperialist ambitions amongst nations. As usual, the process begins with initialisation of a population of solutions, in this case referred to as countries. The countries with better fitness solutions are

chosen to be the imperialist nations and the rest are colonies of the same. The imperialist nations and their colonies form an empire with each empire fighting for dominance. At the end of the convergence only one empire remains with its imperialist and its colonies having the same fitness which results in achieving the optimal solution. The TLA is based on the teaching-learning process characteristic of a classroom. The process begins with the initialisation of the population which is referred to as bunch of learners and the parameters of the objective function of the problem are given to them as different subjects. The grades of the learner's determine the fitness value of that learner. The learner with the best fitness is then chosen as the teacher of that population and then the exchange of information begins between teacher and the learners and between the learners themselves which eventually over generations leads to the global optimal solution. The different objective functions used in this study include minimization of fuel cost, minimization of fuel cost with valve point effect, minimisation of fuel cost with prohibited zones, minimisation of fuel cost with valve point effect and prohibited zones, piece wise quadratic fuel cost function. The comparison of the results was made with GA, PSO, SA, ICA, TLA, Modified ICA (MICA), MICA-TLA, SFLA and SFLA-SA. The proposed algorithm outperforms other algorithms with which the comparisons were made.

A Glow-worm Swarm Optimization (GSO) algorithm for OPF is described in [56]. This study uses a multi-objective function minimising both costs and emissions. The performance of the algorithm has been tested on two systems which are the IEEE 30 bus and the practical Indian 75 bus system. This algorithm is based on the activities of the glow-worm during the night. These insects interact with each other and are attracted to one another based on the amount of luciferin (responsible for the glow) a worm produces. The process begins with randomly scattering the glow-worms in the workspace and at this point all the worms have the same amount of luciferin, every further iteration results in either the increase or decrease of luciferin based on the fitness of the solution obtained by the moving glow-worm. The way every worm moves from one position to another during one iteration to another is determined in a probabilistic manner. This process is slightly biased as the worms inherently move in the direction of the worm with the strongest glow or luciferin. The final solution represents the point at which all the worms converge. The study has considered several cases which are minimization of cost for the IEEE 30 bus and the Indian practical 75 bus system and multi-objective optimization of the IEEE 30 bus system. The results of this approach have been compared with that of the results from PSO applied to the same systems. It has been shown that the GSO performs better.

References

[1] "Global Energy Review 2019," *Glob. Energy Rev. 2019*, 2020, doi: 10.1787/90c8c125-en.

[2] P. Agreement and U. Nations, "Paris agreement," 2015.

[3] E. Comission, "The European Green Deal," 2019.

[4] U. E. and S. Council, "Progress towards the Sustainable Development Goals," 2019.

[5] D. T. T. and M. A. S. I., "The U.S. Department of Energy's Microgrid Initiative," 2012, doi: 10.1016/j.tej.2012.09.013.

[6] C. Marnay *et al.*, "Microgrid evolution roadmap," *Proc. - 2015 Int. Symp. Smart Electr. Distrib. Syst. Technol. EDST 2015*, pp. 139–144, 2015, doi: 10.1109/SEDST.2015.7315197.

[7] C. Schwaegerl and L. Tao, "The microgrids concept," in *Microgrids: Architectures and Control*, 2013.

[8] M. F. Zia, E. Elbouchikhi, and M. Benbouzid, "Microgrids energy management systems: A critical review on methods, solutions, and prospects," *Appl. Energy*, vol. 222, no. March, pp. 1033–1055, 2018, doi: 10.1016/j.apenergy.2018.04.103.

[9] L. Luukkonen, P.; Bateman, P.; Hiscock, J.; Poissant, Y.; Dignard-Bailey, "Photovoltaic and Solar Forecasting : State of the Art," *Int. Energy Agnecy*, 2013.

[10] C. F. M. C. Yang Dazhi, Jan Kleissl, Christian A. Gueymard, Hugo T.C. Pedro, "History and trends in solar irradiance and PV power forecasting_ A preliminary assessment and review using text mining _ Elsevier Enhanced Reader.pdf." .

[11] P. A. Jimenez *et al.*, "WRF-SOLAR: Description and clear-sky assessment of an augmented NWP model for solar

power prediction," *Bull. Am. Meteorol. Soc.*, vol. 97, no. 7, pp. 1249–1264, 2016, doi: 10.1175/BAMS-D-14-00279.1.
[12] N. Andrew, "Machine Learning Yearning, Technical Strategy for AI Engineers In the Era of Deep Learning," 2018.
[13] Y. Bengio, "Deep Learning of Representations for Unsupervised and Transfer Learning," *JMLR Work. Conf. Proc.*, vol. 7, pp. 1–20, 2011.
[14] S. Jaidee, "Deep Neural Network based on Genetic Algorithm and Ensemble Methods for Regional Solar Power Forecasting in Thailand," pp. 377–380, 2019.
[15] D. A. Snegirev, S. A. Eroshenko, R. T. Valiev, and A. I. Khalyasmaa, "Algorithmic Realization of Short-term Solar Power Plant Output Forecasting," pp. 228–231, 2017.
[16] F. Serttas, "Short Term Solar Power Generation Forecasting : A Novel Approach," pp. 2018–2021, 2018.
[17] X. Zhang *et al.*, "A Solar Time Based Analog Ensemble Method for Regional Solar Power Forecasting," vol. 10, no. 1, pp. 268–279, 2019.
[18] M. Sun, C. Feng, and J. Zhang, "Probabilistic solar power forecasting based on weather scenario generation," *Appl. Energy*, vol. 266, no. March, p. 114823, 2020, doi: 10.1016/j.apenergy.2020.114823.
[19] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, "A review on the forecasting of wind speed and generated power," vol. 13, pp. 915–920, 2009, doi: 10.1016/j.rser.2008.02.002.
[20] X. Zhao, S. Wang, and T. Li, "Energy Procedia Review of Evaluation Criteria and Main Methods of Wind Power Forecasting," *Energy Procedia*, vol. 12, pp. 761–769, 2011, doi: 10.1016/j.egypro.2011.10.102.
[21] Q. Chen, C. Town, C. Town, and C. Town, "Wind Power Forecasting Wind Power Forecasting," *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 414–419, 2018, doi: 10.1016/j.ifacol.2018.11.738.
[22] Aasim, S. N. Singh, and A. Mohapatra, "Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting," *Renew. Energy*, vol. 136, pp. 758–768, 2019, doi: 10.1016/j.renene.2019.01.031.
[23] M. Liu, Z. Cao, J. Zhang, L. Wang, C. Huang, and X. Luo, "Short-term wind speed forecasting based on the Jaya-SVM model," *Int. J. Electr. Power Energy Syst.*, vol. 121, no. March, p. 106056, 2020, doi: 10.1016/j.ijepes.2020.106056.
[24] Z. Liu, P. Jiang, L. Zhang, and X. Niu, "A combined forecasting model for time series: Application to short-term wind speed forecasting," *Appl. Energy*, vol. 259, no. October 2019, p. 114137, 2020, doi: 10.1016/j.apenergy.2019.114137.
[25] G. Memarzadeh and F. Keynia, "A new short-term wind speed forecasting method based on fine-tuned LSTM neural network and optimal input sets," *Energy Convers. Manag.*, vol. 213, no. December 2019, p. 112824, 2020, doi: 10.1016/j.enconman.2020.112824.
[26] A. A. Ezzat, "Turbine-specific short-term wind speed forecasting considering within-farm wind field dependencies and fluctuations," *Appl. Energy*, vol. 269, no. January, p. 115034, 2020, doi: 10.1016/j.apenergy.2020.115034.
[27] C. Kuster, Y. Rezgui, and M. Mourshed, "Electrical load forecasting models: A critical systematic review," *Sustain. Cities Soc.*, vol. 35, no. July, pp. 257–270, 2017, doi: 10.1016/j.scs.2017.08.009.
[28] R. Li, J. H. Li, and H. M. Li, "The Short – Term Electric Load Forecasting Grid Model Based on MDRBR Algorithm," 2006.
[29] G. Chitalia, M. Pipattanasomporn, V. Garg, and S. Rahman, "Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks," *Appl. Energy*, vol. 278, no. January, p. 115410, 2020, doi: 10.1016/j.apenergy.2020.115410.
[30] M. Lekshmi, "Short-Term Load Forecasting of 400kV Grid Substation Using R-Tool and Study of Influence of Ambient Temperature on the Forecasted Load," pp. 1–5, 2019.
[31] M. Barman, N. Behari, and D. Choudhury, "A similarity based hybrid GWO-SVM method of power system load forecasting for regional special event days in anomalous load situations in," *Sustain. Cities Soc.*, vol. 61, no. January, p. 102311, 2020, doi: 10.1016/j.scs.2020.102311.
[32] C. D. Maciel, "Multiple households very short-term load forecasting using bayesian," *Electr. Power Syst. Res.*, vol. 189, no. August, p. 106733, 2020, doi: 10.1016/j.epsr.2020.106733.
[33] H. Abdi, S. D. Beigvand, and M. La Scala, "A review of optimal power flow studies applied to smart grids and microgrids," *Renew. Sustain. Energy Rev.*, vol. 71, no. May 2015, pp. 742–766, 2017, doi: 10.1016/j.rser.2016.12.102.
[34] S. Frank, I. Steponavice, and S. Rebennack, "Optimal power flow: A bibliographic survey I Formulations and deterministic methods," *Energy Syst.*, vol. 3, no. 3, pp. 221–258, 2012, doi: 10.1007/s12667-012-0056-y.
[35] H. H. A. El-Fergany, "Tree-seed algorithm for solving optimal power flow problem in large-scale power systems incorporating validations and comparisons," *Appl. Soft Comput. J.*, vol. 64, no. 5, pp. 307–316, 2018, doi: 10.1016/j.asoc.2017.12.026.
[36] J. Kim and K. K. K. Kim, "Dynamic programming for scalable just-in-time economic dispatch with non-convex constraints and anytime participation," *Int. J. Electr. Power Energy Syst.*, vol. 123, no. May, p. 106217, 2020, doi: 10.1016/j.ijepes.2020.106217.
[37] H. Saberi, T. Amraee, C. Zhang, and Z. Y. Dong, "A heuristic benders-decomposition-based algorithm for transient stability constrained optimal power flow," *Electr. Power Syst. Res.*, vol. 185, no. May, p. 106380, 2020, doi: 10.1016/j.epsr.2020.106380.
[38] S. Rahmani and N. Amjady, "Enhanced goal attainment method for solving multi-objective security-constrained optimal power flow considering dynamic thermal rating of lines," *Appl. Soft Comput. J.*, vol. 77, pp. 41–49, 2019, doi: 10.1016/j.asoc.2019.01.014.
[39] B. V. Rao and G. V. N. Kumar, "Electrical Power and Energy Systems Optimal power flow by BAT search algorithm for generation reallocation with unified power flow controller," *Int. J. Electr. Power Energy Syst.*, vol. 68, pp. 81–88, 2015, doi: 10.1016/j.ijepes.2014.12.057.
[40] H. Sharifzadeh, N. Amjady, and H. Zareipour, "Multi-period stochastic security-constrained OPF considering the uncertainty sources of wind power, load demand and equipment unavailability," *Electr. Power Syst. Res.*, vol. 146, pp. 33–42, 2017, doi: 10.1016/j.epsr.2017.01.011.

[41] D. Fang, M. Zou, G. Coletta, A. Vaccaro, and S. Z. Djokic, “Handling uncertainties with affine arithmetic and probabilistic OPF for increased utilisation of overhead transmission lines,” *Electr. Power Syst. Res.*, vol. 170, no. January, pp. 364–377, 2019, doi: 10.1016/j.epsr.2019.01.027.

[42] K. Vaisakh and L. R. Srinivas, “Evolving ant direction differential evolution for OPF with non-smooth cost functions,” *Eng. Appl. Artif. Intell.*, vol. 24, no. 3, pp. 426–436, 2011, doi: 10.1016/j.engappai.2010.10.019.

[43] T. Ding, C. Li, F. Li, T. Chen, and R. Liu, “A bi-objective DC-optimal power flow model using linear relaxation-based second order cone programming and its Pareto Frontier,” *Int. J. Electr. Power Energy Syst.*, vol. 88, pp. 13–20, 2017, doi: 10.1016/j.ijepes.2016.11.012.

[44] M. R. E. S. W. Zia Ullah, K. S. M. H. Muhammad Azam, and and M. Rizwan, *A Mini-review: Conventional and Metaheuristic Optimization Methods for the Solution of Optimal Power Flow (OPF) Problem*, vol. 1. Springer International Publishing, 2020.

[45] T. A. Al-Muhawesh and I. S. Qamber, “The established mega watt linear programming-based optimal power flow model applied to the real power 56-bus system in eastern province of Saudi Arabia,” *Energy*, vol. 33, no. 1, pp. 12–21, 2008, doi: 10.1016/j.energy.2007.08.004.

[46] L. P. M. I. Sampath, B. V. Patil, H. B. Gooi, J. M. Maciejowski, and K. V. Ling, “A trust-region based sequential linear programming approach for AC optimal power flow problems,” *Electr. Power Syst. Res.*, vol. 165, no. September, pp. 134–143, 2018, doi: 10.1016/j.epsr.2018.09.002.

[47] A. M. Sasson, “Combined Use of the Powell and Fletcher—Powell Nonlinear Programming Methods for Optimal Load Flows,” *IEEE Trans. Power Appar. Syst.*, vol. PAS-88, no. 10, pp. 1530–1537, 1969, doi: 10.1109/TPAS.1969.292281.

[48] P. Fortenbacher and T. Demiray, “Linear/quadratic programming-based optimal power flow using linear power flow and absolute loss approximations,” *Int. J. Electr. Power Energy Syst.*, vol. 107, no. October 2018, pp. 680–689, 2019, doi: 10.1016/j.ijepes.2018.12.008.

[49] V. S. Suresh, “Comparison of Solvers Performance for Load Flow Analysis,” *Trans. Environ. Electr. Eng.*, vol. 3, no. 1, p. 26, 2019, doi: 10.22149/teee.v3i1.131.

[50] F. Capitanescu and L. Wehenkel, “Experiments with the interior-point method for solving large scale Optimal Power Flow problems,” *Electr. Power Syst. Res.*, vol. 95, pp. 276–283, 2013, doi: 10.1016/j.epsr.2012.10.001.

[51] M. Zhang, R. S. Kar, Z. Miao, and L. Fan, “New auxiliary variable-based ADMM for nonconvex AC OPF,” *Electr. Power Syst. Res.*, vol. 174, no. May, p. 105867, 2019, doi: 10.1016/j.epsr.2019.105867.

[52] Z. Abdmouleh, A. Gastli, L. Ben-Brahim, M. Haouari, and N. A. Al-Emadi, “Review of optimization techniques applied for the integration of distributed generation from renewable energy sources,” *Renew. Energy*, vol. 113, pp. 266–280, 2017, doi: 10.1016/j.renene.2017.05.087.

[53] M. A. Abido, “Optimal power flow using tabu search algorithm,” *Electr. Power Components Syst.*, vol. 30, no. 5, pp. 469–483, 2002, doi: 10.1080/15325000252888425.

[54] S. Duman, U. Güvenç, Y. Sönmez, and N. Yörükeren, “Optimal power flow using gravitational search algorithm,” *Energy Convers. Manag.*, 2012, doi: 10.1016/j.enconman.2012.02.024.

[55] M. Ghasemi, S. Ghavidel, S. Rahmani, A. Roosta, and H. Falah, “Engineering Applications of Arti fi cial Intelligence A novel hybrid algorithm of imperialist competitive algorithm and teaching learning algorithm for optimal power fl ow problem with non- smooth cost functions,” *Eng. Appl. Artif. Intell.*, vol. 29, pp. 54–69, 2014, doi: 10.1016/j.engappai.2013.11.003.

[56] K. N. Krishnanand and D. Ghose, “Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions,” *Swarm Intell.*, 2009, doi: 10.1007/s11721-008-0021-5.

2. Solar Forecasting

While the introduction section provided approaches and examples to forecasting of various elements crucial to power system operation, this thesis mainly focuses on solar PV output power forecasting due to the setup at Wroclaw University of Science and Technology. In section 1.2.1 numerous classes of methods to solar forecasting were introduced and, in this thesis, the approaches investigated forms a part of the class of machine learning models for forecasting called deep learning.

Deep learning represented by Deep Neural Networks (DNNs) can be simply defined as ANNs with a deep stack of hidden layers consisting of deep stacks of computation [1]. These class of neural networks are powerful, scalable and versatile making them suitable and applicable to solve highly complicated and large machine learning tasks such as image classification, speech recognition etc.

The purpose of choosing deep learning-based forecasting models are as follows, the traditional forecasting algorithms based on machine learning and also statistical methods flatten quite early in terms of performance. This means that irrespective of the amount of computational power and data available there is a strict and short limit to which the relationship between variables can be ascertained by the algorithm, that is they saturate in performance. In comparison the deep learning networks do not saturate early and improve in performance with greater availability of data and computational power. A comparison of numerous well known machine learning algorithms such as the linear regression, small neural networks and logistic regression with DNNs in terms of performance can be found in [2]. Another argument for choosing this approach is feature learning which enables the model to extract features from raw data and learn numerous unknown representations that might be present within it. This results in forecasting models of improved accuracy [3].

The forecasting is performed for predicting the output power of the solar PV panels located above the building of Wroclaw University of Science and Technology. In total there are 3 different modules which are polycrystalline, monocrystalline and CIGS that consist of 27, 21 and 56 panels. The forecasting carried out in this study is mainly on the data from the polycrystalline module but there is not reason as to why it cannot be extended to other modules. The setup is quite elaborate and numerous variables are measured which are the PV module temperatures (^{o}C), wind speed (m/s), irradiation (W/m^2) and ambient temperature (^{o}C). The peak power output of the module is 5 kW.

2.1 Data pre-processing

While different deep learning algorithms are explored in this thesis. The data pre-processing remains the same with every approach. The features utilised in this study to train models are different, they differ from one another with regards to their units of measurement, range of their values and their respective distributions. During training of neural networks when data of such varied characteristics are used, it leads to big weight values and models tend to perform poorly and can be highly sensitive to any changes in the input data [4]. In order to deal with the challenges mentioned above, all the data used in this study are normalized within the range of [0,1]. The algorithm used to do so is provided below:

$$\frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{1}$$

Where, x represents the vector of the data and x_i represents one particular value in the data.

Another data processing approach used is the sliding window algorithm, while this step is not essential to all the deep learning models, it is imperative for some and in general useful for all as it speeds up model the model training process. Later on depending on the type of DNN used it will explained how this algorithm actually splits the data before training begins. The algorithms is described below.

Algorithm 1

Procedure Variables (X, Z, t)
    i = 0, n = 0; # number of windows = n
    P = []; # P is the set of windows extracted
    While i + Z ≤ length (X) **do** #Z is the length of the sliding window
    P[n] = X [i…. (i + Z – 1)};
    i = i + t; n = n + 1;
    end While
    return F
end Procedure

2.2 Evaluation metrics

The Evaluation metrics utilised in this study to ascertain the performance of a model were selected based on the recommendations of the International Energy Agency (IEA) co-operative programme on photovoltaic systems [5] and a popularly cited review paper on solar forecasting [6]. The metrics are the RMSE, MAE and Mean Bias Error (MBE). RMSE is a parameter which is used to evaluate numerous forecast results, in general it is suitable for solar forecasting as well since it has the characteristic to heavily punish larger errors than smaller ones, on the contrary the MAE punishes all errors equally and MBE provides general information as to whether the model obtained has a tendency to over predict or under predict. The metrics are described below.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_i^2} \tag{2}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |e_i| \tag{3}$$

$$MBE = \frac{1}{N}\sum_{i=1}^{N} e_i \tag{4}$$

$$e_i = y_{i(forecast)} - y_{i(observed)} \tag{5}$$

Here, $y_{i(forecast)}$ and $y_{i(observed)}$ are the forecasted and observed values at the i^{th} time step. e_i is defined as the error at i^{th} time step. $i = 1, \dots \dots, N$ defines all time steps. The evaluation metrics values presented later on are obtained from data that is reverted back from their normalized values by using the inverse of the min-max algorithm mentioned in (1).

2.3 Convolutional Neural Networks (CNNs)

CNNs first introduced in 1998, have been widely adopted for numerous applications such as image recognition, text recognition, translation, speech recognition, sentence classification, face detection etc. [7]. It has the ability to extract spatial and temporal relationships within data

which is crucial for the forecasting process. Since this thesis is working with data that are essentially vectors the CNN is modified and a 1-D CNN is used.

The CNN can be visualized in terms of different layers as shown in Fig. 1 and each layer is defined below:

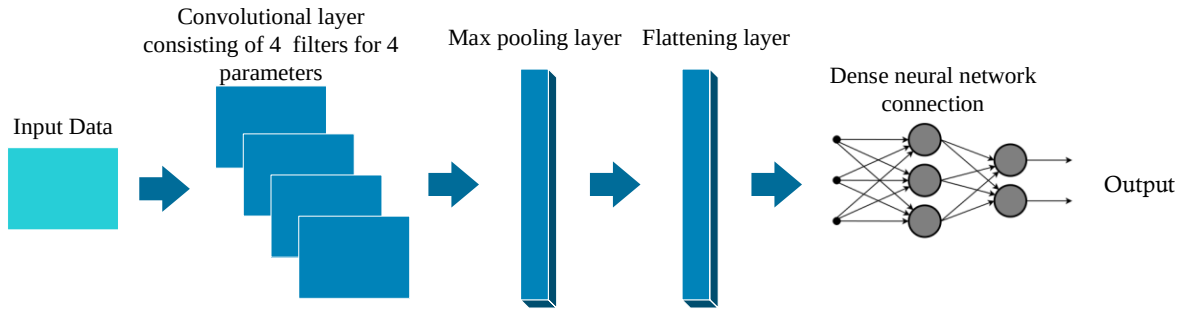

Figure 1 CNN structure visualization

Convolutional layer: This layer is at the heart of a CNN and is responsible for feature extraction [7], [8]. The feature extraction is particularly done by filters also known as kernels. Given the input data these filters can be visualized as rectangles having a width equal to the number of input features in the data and length equal to what is decided by the user. The length in short is the number of time steps that the user would like to take into account. Once this filter is defined it is moved along the entire length of the data one-time step at a time or more as defined by the user. In this study this shifting of the filter also known as the stride is 1 time step [9]. Fig. 2(a) shows such as example where the input data is traversed by the kernel covering 4 possible features in width and 3-time steps in length. The elements of the kernel are multiplied with the elements of the boxes of the input data they cover and are added together. This value is then sent through the activation function of the kernel. Once the kernel has moved along the entire length of the data a new vector of a reduced dimension is created. This dimension reduction helps the CNN to learn features quickly even if the amount of data available is massive.

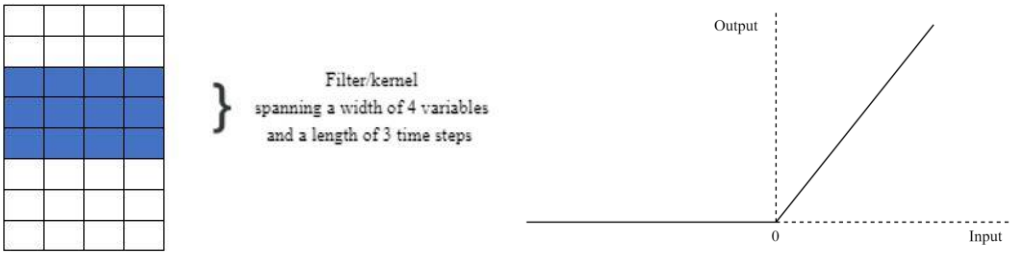

Figure 2 (a) Kernel movement visualization (b) ReLU function

The number of filters or kernels chosen in this study is always kept equal to the number of variables or features that are to be learnt, assuming 4 input variables exist, 4 filters are chosen as shown in Fig. 1. This is not mandatory; any number of filters can be chosen. The activation function chosen in the study is the ReLU (Rectified Linear Unit), this is the one of the most widely used activation functions [7] though it is also possible to use others such as sigmoid, Tanh etc.

Being a non - linear function ReLU actually behaves like a linear function. This feature enables the function to learn complex relationships that exists within the input data. It works in the following manner, when the input value remains above 0.0 the output value remains unchanged and when the input value is less than or equal to 0.0 the output value is 0.0. It is defined in (6).

$$g(z) = \max\{0, z\} \quad (6)$$

Where, z is the input value and g is the ReLU function. The advantages of using ReLU are that it is easy to implement for neural networks due to its linear function like characteristic despite being non-linear, sparsity and ease of computation [10]. The function is visualized in Fig. 2(b).

Pooling Layer: The convolutional layer is followed by the pooling layer which reduces the output of the convolutional layer. In short it summarises the features that have been learnt in the previous step. This reduces the size of the data resulting in higher processing speeds and reduced demand for memory and also prevents overfitting. There are different types of pooling layers such as averaging and max pooling [1], the most widely used is the max pooling layer. The max pooling layer in general is reducing the input data massively, even with a kernel of length 2 and stride 2 for a 1-D CNN the data is be reduced by as much as 50%. Hence, choosing the right parameters during model training are imperative for good accuracy.

Flattening Layer: The flattening layer following the max-pooling layer transforms the output into a vector of one dimension which then serves as the input for the dense or fully connected layer.

The optimization algorithms used to minimize errors and fit a model are of many types. They are RMSprop, Adam, AdaGrad, Nadam etc. The optimization algorithm mainly used for the models in this study is Adam which is a combination of two well-known stochastic gradient descent optimizers known as AdaGrad (Adaptive Gradient) algorithm and RMSProp (Root Mean Square Propagation). The advantage of using Adam is that it is able to adjust its learning rate as the error is minimized during model fitting.

In general, when it comes to CNN based forecasting models, 3 types of architectures are explored in this thesis. They are the simple CNN model as elaborated in Fig. 1, the multi-headed CNN model described in Fig. 3 and a CNN-LSTM model shown in Fig. 4 which is in fact a combination of 2 different types of deep learning architectures.

The approach in a simple CNN model is that feature extraction is done by a single convolutional layer or a single CNN with filters equal to the number of features in the input data. In case of the multi-headed CNN as shown in Fig. 3, a CNN layer is used for every feature in the input data, assuming there are 4 features or variables, 4 different CNNs are used. These are then followed by 4 max-pooling and 4 flattening layers for dimensionality reduction and conversion to a 1-D vector. Since the output from 4 flattening layers have to be tied together into 1, a concatenation layer is used, the output from which is used to train a dense neural network. This approach is shown to have some flexibility but there is not explicit evidence in the literature that it results in accurate forecasting models. An example of such an architecture being used can be found in [11] where a multi-headed CNN approach with 2-D convolutional nets have been used for image classification.

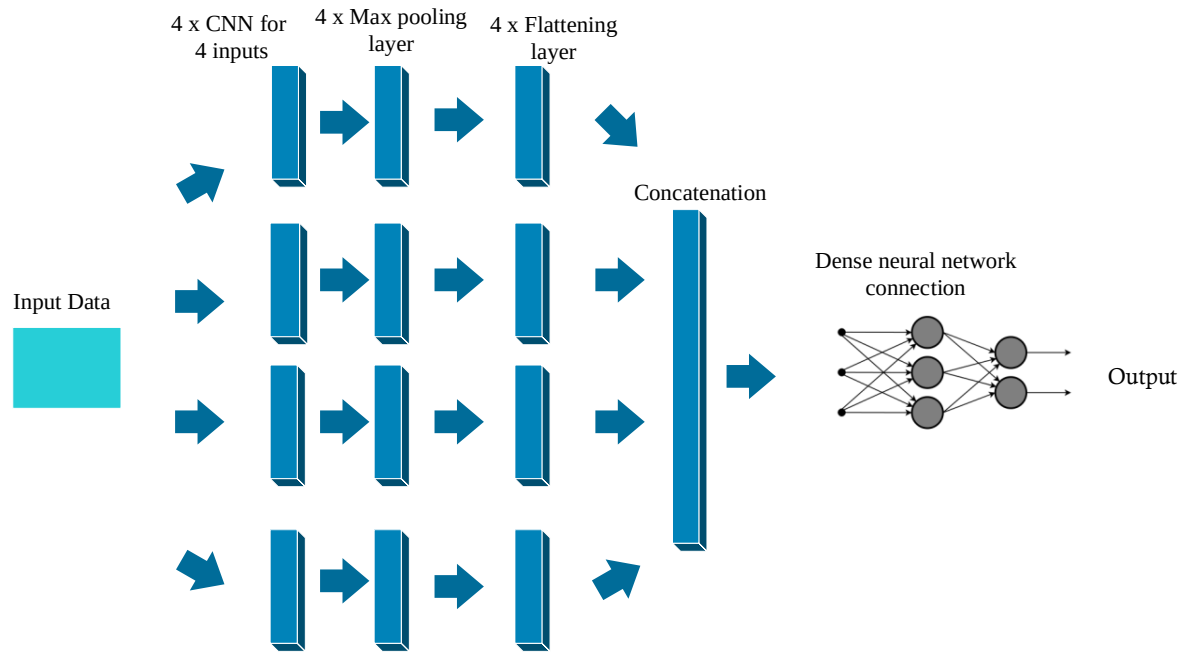

Figure 3 Multi-headed CNN structure.

The architecture for a CNN-LSTM model is shown in Fig. 4 and as mentioned before it is a combination of two different deep learning neural networks where LSTM stands for Long Short-Term Memory. This model has been implemented for numerous time series forecasting problems that have both temporal and spatial relationships to be exploited. The LSTM in particular is very suitable when it comes to time series forecasting and classification. This is because of the presence of a memory cell which in fact behaves as an accumulator of state information. Whenever new information is fed into an LSTM, a part of it is captured by it and is sent back to itself while the rest is forgotten. This ability of the LSTM allows it to remember past data and when a relationship is ascertained between inputs and outputs, the temporal relationship is also captured. This is markedly different from how regular ANNs work. A detailed explanation into the workings of an LSTM will be explained in its respective future section.

The architecture is similar to that of the regular CNN shown in Fig. 1 with the exception of an LSTM layer beyond the flattening layer used to extract the temporal relationships from the 1-D vector output of the flattening layer.

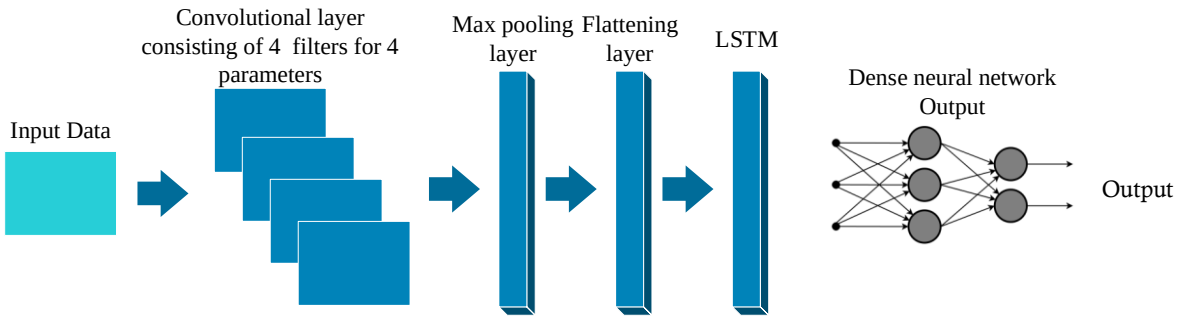

Figure 3. CNN-LSTM structure

It is important to understand how the data is manipulated by different structures as it moves through the numerous layers of different architectures mentioned above. The data manipulation

as it moves through the CNN, multi – headed CNN and the CNN – LSTM model are shown below.

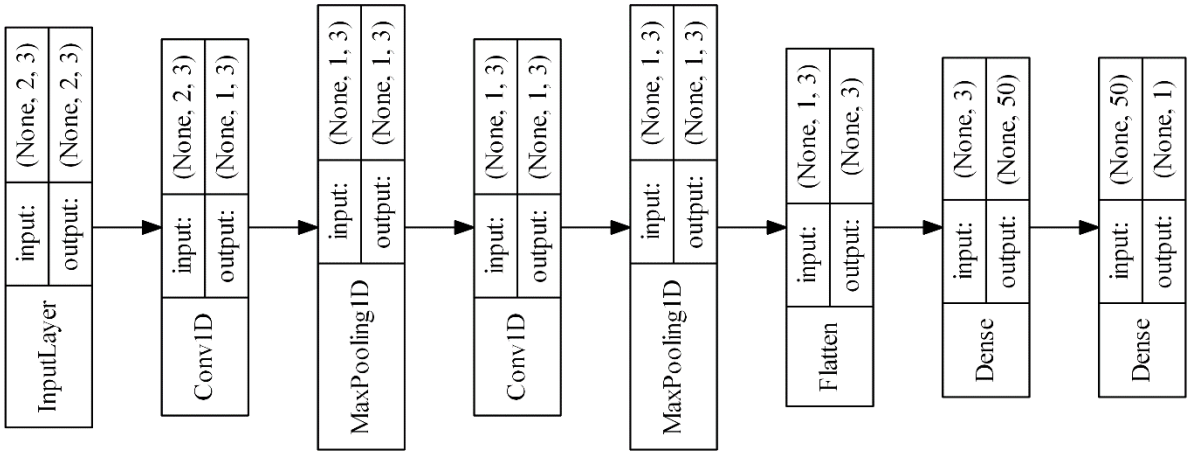

Figure 4. Data flow visualisation for the simple CNN architecture

For the CNN model, it is important to reshape the data into a [samples, timesteps, features] structure which is done by the input layer shown in Fig 4. It can be seen as [None, 2, 3]. None because the model created has the ability to look up the number of samples automatically whenever an array is given to it. 2 represents the past two-time steps that are taken into account and 3 features represent the 3 variables used which are output power, ambient temperature and irradiation.

The convolutional layer reduces the size of the input data and creates a learned representation model of the same followed by a max pooling layer which summarises them. This is then followed by another convolutional layer and a max pooling layer in order to extract further relationships within the input data. This is then followed by the flattening layer which creates a 1-D vector representation of the data which is initially fed to a fully-connected DNN consisting of 50 neurons that is followed by another DNN that takes the input from the previous DNN and produces a 1-D vector of the output which is the forecasted output power.

As described earlier the multi – headed CNN structure is simply a structure where every feature of the input data is handled by its own individual CNN, the learned representations of which are tied together during model training. Fig. 4 presents the data flow during training of such a model. As in the previous case the data is shaped into [samples, timesteps, features]. Instead of having a single [None, 2, 3] input, we have 3 [None, 2, 1] inputs where again 2 represents the number of time steps taken into account and 1 represents the one feature being learnt. The 3 input layers are then followed by 3 convolutional layers for size reduction and 3 max pooling layers for summarising the learning. This is then followed by another set of 3 convolutional layers and max pooling layers for further extraction of relationships. The data is then flattened to a 1-D vector by the flattening layer and there are 3 according to the 3 learned representations from the 3 different features used in this study. The outputs from the 3 layers are then tied together using a concatenation layer before they are given as an input to the DNN consisting of 50 neurons followed by another DNN which takes the input from this DNN and outputs a single 1-D vector of the forecasted power output.

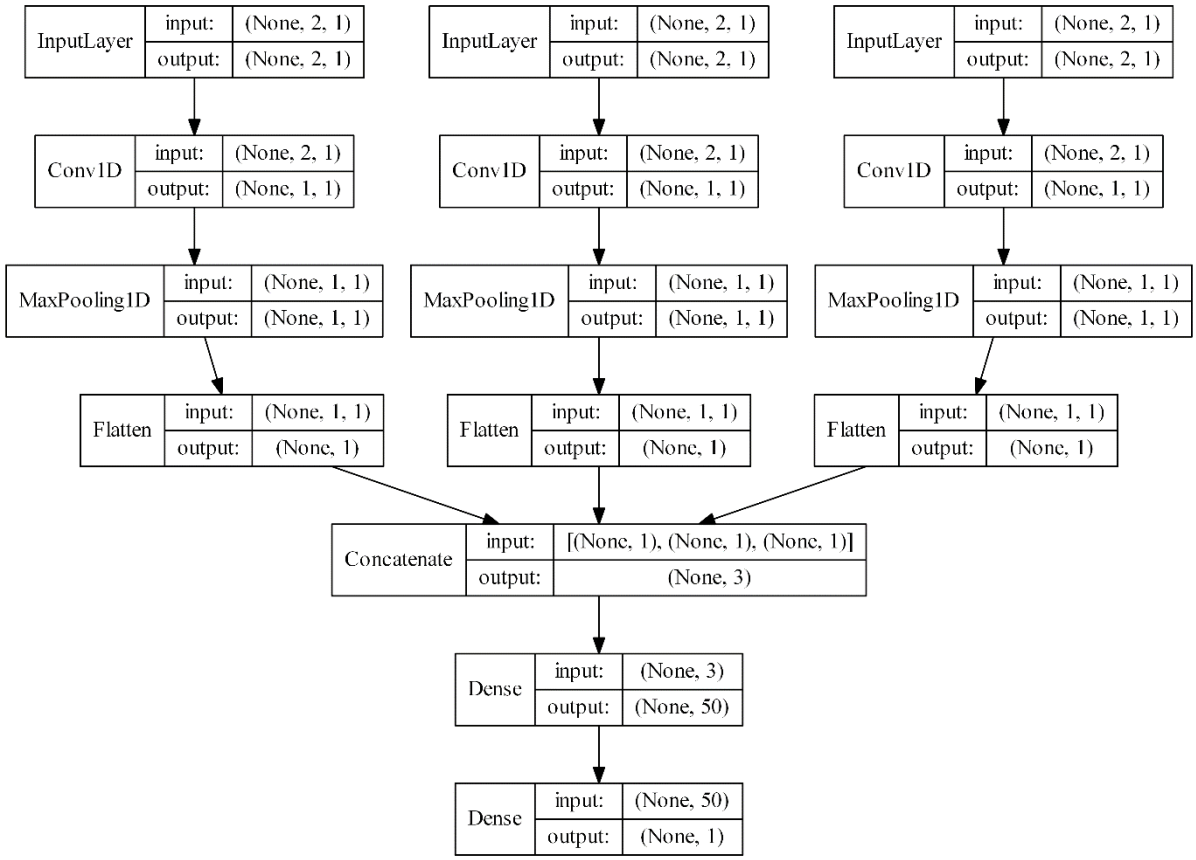

Figure 5. Data flow visualisation for the multi-headed CNN architecture

The CNN-LSTM as described before is a combination of two deep learning models and can be quite effective in feature extraction and learning. In the model used in this study as shown in Fig. 3 the LSTM layer is used at the backend and the input data is analysed by the CNN model. The input data which has been fed until now has always been in the form [samples, timesteps, features] but for this model the input array is further split into two sub sequences which is then analysed by the CNN creating a learned representation which is then given as an input to the LSTM layer.

This study initially feeds an array of 4-time steps which is then split into sub sequences consisting of 2-time steps each which is then processed by the CNN and sent to the LSTM layer. Hence, the data for this model is in the form of [samples, sub sequences, timesteps, features].

It can be seen that the input layer has a shape [?, 2, 2, 3], number of samples has been represented by a "?" because the model is able to judge automatically the total number of samples. The sub sequences are 2 with 2-time steps each and the number of features is 3, the same as for the above two models. Then, as done before the CNN reduces the size of the input data and creates a learned representation which is summarised by the max-pooling layer. This is then further sent for feature extraction by means of another convolutional and max-pooling layer. The dimension is then reduced by the flattening layer and then the output is fed to the LSTM layer which along with the 1 neuron DNN makes the output power forecast. A comprehensive explanation of the LSTM structure will be explained in the coming sections.

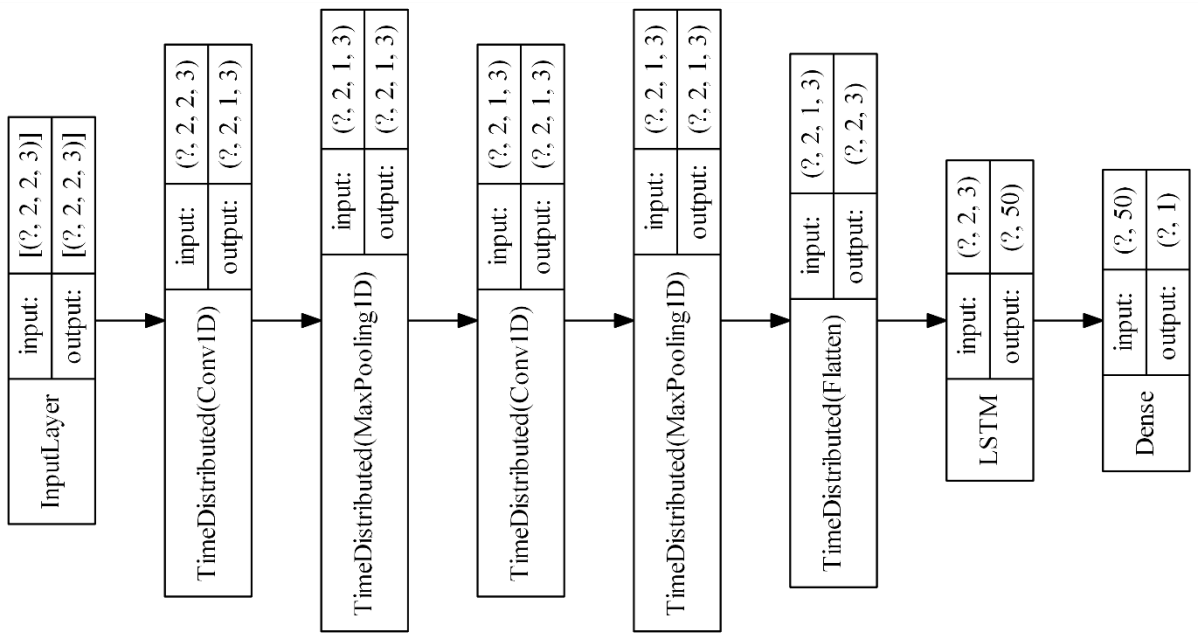

Figure 6. Data flow visualisation for the multi-headed CNN architecture

2.3.1 10 – min ahead forecasting

The forecasting models used in this thesis were all coded in PYTHON using Jupyter Notebooks. The environment for deep learning utilised was TensorFlow and the models were constructed using KERAS. Additionally, numerous PYTHON libraries were put to use for data pre - processing and manipulation. The fitting and evaluation of models and analysis of the results were carried out in a computer with an Intel core i7 9th generation processor with a 16 GB RAM running on a Windows 10 operating system.

The input data used in this study has already been explained previously. Here, only the size of the input vector is defined which is an array of dimensions 201755 x 3. Where, 201755 represents 10-minute time steps taken into account and 3 represents the 3 features taken into account which are the output power, irradiation and ambient temperature from the previous time steps.

The evaluation metrics already explained previously are used to determine the effectiveness of the forecasting models used. The nature of machine learning models to fit different models when they are run every time poses a threat to the reliability of such models. Quite often the accuracy of many such models vary over a range and it is imperative to understand this range in order to make accurate forecasts. For 10-minute forecasts, the results of forecasting utilising all above mentioned models have been analysed by running all models 200 times. In every single run the testing and training data have been selected at random in order to understand how slight changes in training data affects such models and also how the accuracy varies over several runs. While 200 runs may not be enough, the computational burden involved in running them higher than 200 times is significant, even for 200 runs it takes several hours. Later on, while evaluating other deep learning models that are significantly less computationally intensive the results will be analysed over a higher number of runs.

It can be seen from table 1 that except for the CNN-LSTM model, other forecasting models have the tendency to overpredict. This means that on average the predictions made by these models are higher than the actual value. In terms of minimum and maximum values of the RMSE, the spread is the highest for the CNN model and lowest for the CNN-LSTM model

which indicates that the CNN-LSTM model is the most consistent in its performance. The same pattern has been observed in the spread of the MAE values leading to the same conclusion that the CNN-LSTM model is more consistent.

Table 1. 10 – minute ahead forecast evaluation

Models	RMSE (kW)			MAE			BIAS
	Min	Max	Mean	Min	Max	Mean	
CNN	0.009	0.635	0.187	0.005	0.314	0.895	Overprediction
Multi headed–CNN	0.006	0.570	0.166	0.004	0.232	0.060	Overprediction
CNN - LSTM	0.005	0.448	0.109	0.003	0.214	0.052	Underprediction

In order to understand variability of all models, not only is it important to look at the mean, minimum and maximum values of the evaluation criteria over several runs but it is also important to look at the standard deviation and confidence intervals. Table 2 presents

Table 2. 10 – minute ahead forecasts (standard deviation and confidence intervals)

Models	RMSE (kW)		MAE		95% confidence interval	
	Mean	Standard deviation	Mean	Standard deviation	RMSE	MAE
CNN	0.187	0.126	0.089	0.061	[0.170, 0.205]	[0.081, 0.098]
Multi headed-CNN	0.166	0.137	0.060	0.060	[0.147, 0.185]	[0.054, 0.066]
CNN - LSTM	0.109	0.106	0.052	0.050	[0.094, 0.124]	[0.045, 0.059]

In general, the mean of the RMSE for the CNN, multi – headed CNN and CNN - LSTM models over 200 runs as already mentioned before are 0.187 kW, 0.166 kW and 0.109 kW respectively, this is equal to 3.7 %, 3.3% and 2.2% of the peak power output of the solar panel. The standard deviations of the RMSE for the same are 0.126 kW, 0.137 kW and 0.106 kW respectively, which are 2.5 %, 2.7% and 2.2% of the peak power output of the solar panel.

The mean values of the MAE for the 3 models run over 200 times are 0.089 kW, 0.060 kW and 0.052 kW which are 1.7%, 1.2% and 1% of the peak power output of the panel respectively. The standard deviation of the MAE for the same are 0.061 kW, 0.060 kW and 0.050 kW which are 1.2%, 1.2% and 1% respectively.

The 95% confidence intervals for the RMSE and MAE values for all models were calculated and is shown in Table 2. It can be observed that both the spread and the values of the RMSE and MAE are the lowest for the CNN – LSTM model making it the most accurate and reliable model.

Fig. 7 presents the forecasting made every 10-mins for the entire day. 4 random days of the year are considered for this purpose. The days are quite representative with regards to the different seasons in a year seen in Wroclaw. The figure on the upper left and lower left are typical for the summer and days with high availability of sunlight since the peak power during the day reaches 4 kW and above. The figures on the upper and lower right are more typical for the winter and low sunlight days when the power output remains quite low. It can be seen that

the models achieved are able to work reliably despite the change in seasons hence not requiring training different models based on different seasons of the year. The accuracy of all models reduces around the peak values irrespective of the day under consideration.

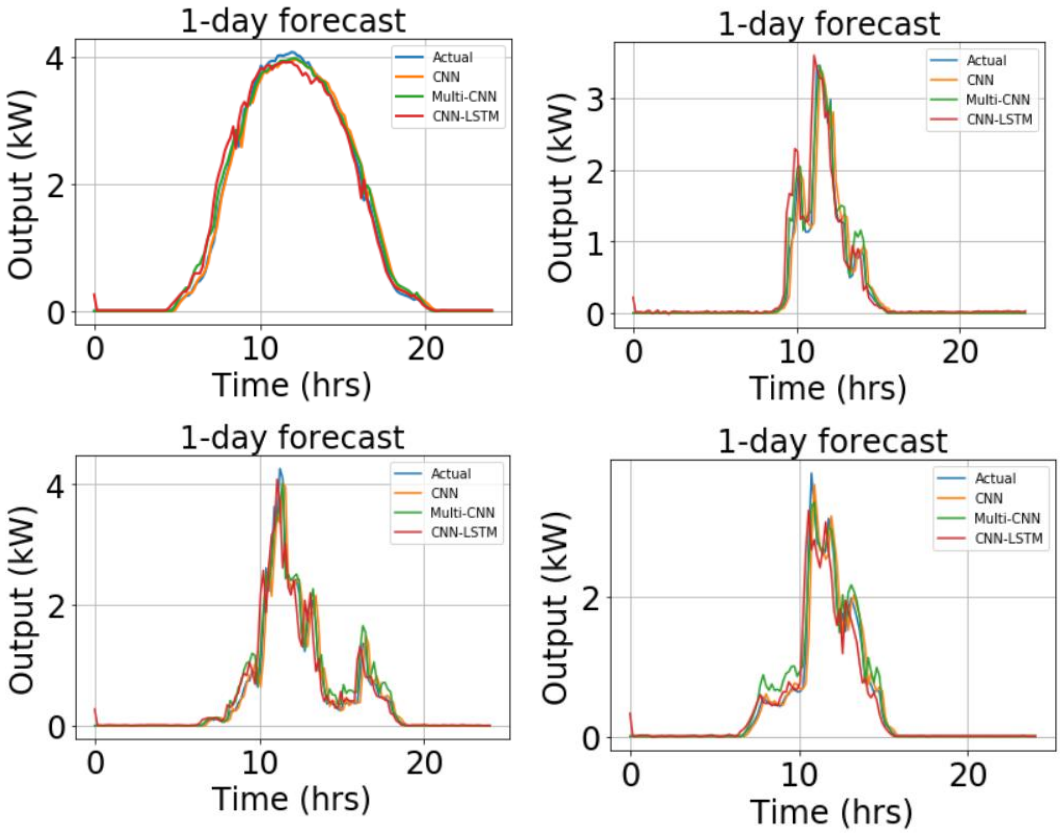

Figure 7. 1-day forecasts for 4 random days of the year with the forecasts made every 10-min

The training time for one run (fitting the model, prediction and evaluation) of the 3 above mentioned models for the input data of the same size are 1min 14 secs, 2 min 50 secs and 2 min 10 secs for the CNN, multi – headed CNN and CNN – LSTM models respectively. From the evaluation metrics analysed above and the run times it can be ascertained that the most accurate model is the CNN – LSTM model and is the second slowest, the multi – headed CNN the slowest but the second most accurate and the CNN is the fastest but least accurate model.

2.3.2 1 – hour ahead forecasting

A similar evaluation for 1- hour ahead forecasts is shown below in Table 3. In this case all models were run 100 times with the training and test data changing in every run. The mean RMSE values for the CNN and CNN-LSTM models are at 0.319 kW and 0.282 kW which are 6.38% and 5.64% of the peak value. For the multi – headed CNN model which is 0.293 kW representing 5.86 % of the peak value. In case of the MAE the CNN-LSTM model has a mean value of 0.152 kW which is 3 % of the peak value and the CNN model has a mean MAE value of 0.171 kW which is 3.42% of the peak value. The multi-headed model has a mean MAE value of 0.200 kW which is 4% of peak value.

Table 3. 1– hour ahead forecast evaluation

Models	RMSE (kW)			MAE			BIAS
	Min	Max	Mean	Min	Max	Mean	
CNN	0.055	0.704	0.319	0.015	0.428	0.171	Underprediction
Multi headed–CNN	0.023	0.418	0.293	0.047	0.331	0.200	Overprediction
CNN - LSTM	0.038	0.605	0.282	0.022	0.331	0.152	Underprediction

Table 4. presents evaluation metrics in order to determine the variability of the models under investigation over the 100 runs performed. All models have varying performance when it comes to variability with the CNN-LSTM model performing better. The CNN-LSTM is less variable and a more reliable model than the CNN and CNN-LSTM since the standard deviation for the CNN-LSTM model is 0.060 (RMSE) and 0.038 (MAE) and the same for the CNN model is 0.090 (RMSE) and 0.070 (MAE) and the same for the multi-headed CNN model is 0.153 (RMSE) and 0.099 (MAE). Like for the 10 – minute ahead forecasts it can be said that the CNN-LSTM model is the most accurate and reliable model for the hour ahead forecasts as well.

When it comes to the 95% confidence intervals of the metrics over all the runs, it can be seen that the spread for the CNN-LSTM model is better.

Table 4. 1 – hour ahead forecasts (standard deviation and confidence intervals)

Models	RMSE (kW)		MAE		95% confidence interval	
	Mean	Standard deviation	Mean	Standard deviation	RMSE	MAE
CNN	0.319	0.090	0.171	0.070	[0.288, 0.424]	[0.168, 0.274]
Multi headed-CNN	0.293	0.153	0.200	0.099	[0.226, 0.373]	[0.179, 0.254]
CNN - LSTM	0.282	0.060	0.152	0.038	[0.213, 0.305]	[0.132, 0.191]

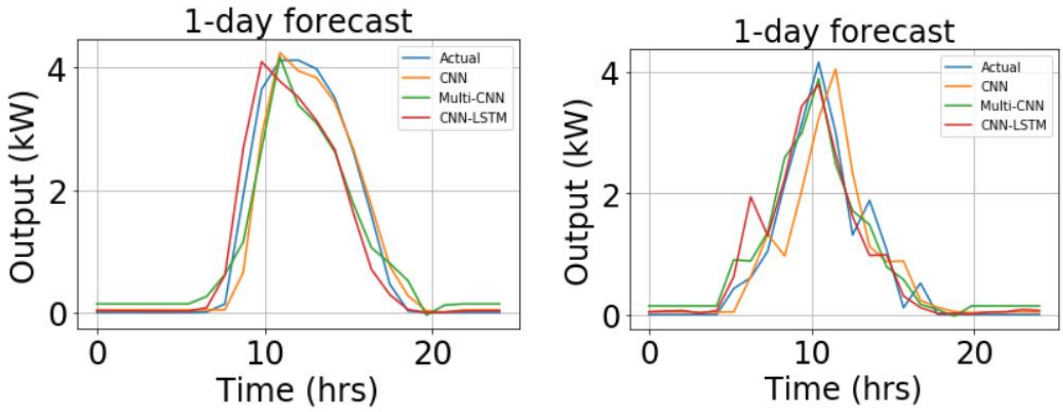

Figure 8. 1-day forecasts for 2 random days of the year with the forecasts made every 1 hour

Fig. 8 presents two different cases. The days were chosen at random and it can be seen that the models are inaccurate at the peaks and in other instances. Overall, when compared to the 10 min ahead forecasts the forecasting models have degraded in performance making this class of deep learning methods suitable only for very short-term forecasting. In the figure on the left

the CNN model follows the actual curve better and in the figure on the right the CNN-LSTM model follows the actual curve better.

2.4 LSTM – Autoencoder (L-AE)

The LSTM category of deep learning models can be defined as specialized Recurrent Neural Networks (RNNs), this category of models has shown significant performance in the case of handling sequential data including time-series sequences [1], [12]. The difference between RNNs and Feed Forward Neural Networks (FFNNs) is the fact that the RNNs posses a memory cell. It is said to have such a feature because unlike in a FFNN where the connections between neurons are just made in one direction (from input to output), in an RNN the connections also point in the backward direction. This results in data being propagated over time, which means that the data received at one time step is stored and then is utilized in the next time step. This propagation of data across time can be visualized as shown in Fig. 9.

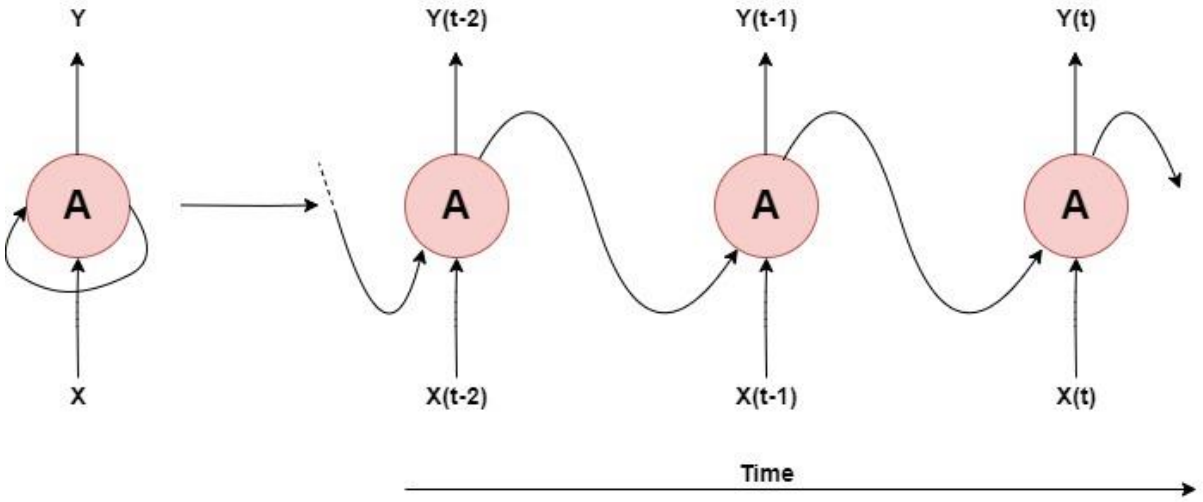

Figure 9. Recurrent neuron (left), unrolled through time (right)

The figure shown above consists of one neuron with an activation function A which takes an input X and produces an output Y. Next, the output is sent back to itself so that it can be used for training the neuron in the next time step along with the new input in the new time step. This way it is made sure that experiences learnt previously are not lost in subsequent time steps. This process is elaborated further on the right-hand side of Fig. 9. in a rolling manner where the process of carrying output from the previous steps to the next time step is visualized. For example, the 3rd neuron from the left receives an input in its time step X(t-1) and also the output from the previous time step Y(t-2) and produces the current output in Y(t-1) which is propagated further. Numerous neurons such as this can be fit into a layer called an RNN layer. In this case the layer would receive a vector of inputs and a vector of outputs from the previous time step and produce a vector of outputs in the current time step. This nature of data handling by RNNs makes them suitable for sequential data. Despite, their suitability they do posses a few disadvantages such as long times for training and vanishing of gradients. This means that over timesteps, the experiences learnt as a result of training at the earlier time steps are lost. This could lead to inaccurate forecasts [1], [12].

In order to overcome the disadvantages of the RNN cell described earlier, numerous models were proposed one of which is the LSTM. The LSTM has quicker training times and is capable of identifying long-term dependencies in data which means it can retain learnt experiences over

numerous time steps [1]. The structure of a memory cell of the RNN and the LSTM are shown in Fig. 10 and the differences are explained below.

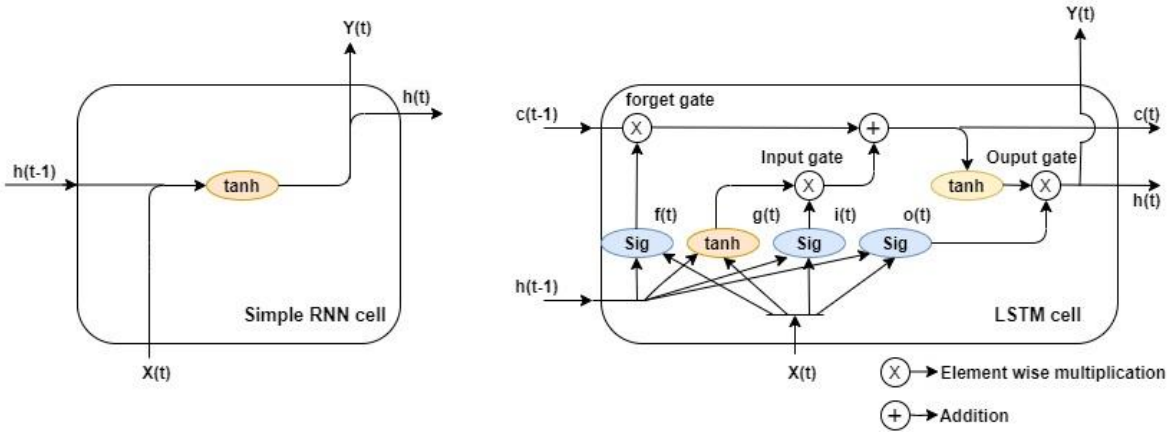

Figure 10. Simple RNN cell (left), LSTM cell (right)

In Fig. 10, *tanh* and *Sig* represent activation functions. X(t) and Y(t) represent inputs and outputs at the time step t. In the simple RNN cell, h(t) and h(t-1) represent the output at t and t-1 respectively. In the LSTM cell, h(t) and h(t-1) represent the short-term state, c(t) and c(t-1) describe the long-term state. f(t), g(t), i(t) and o(t) are gate controllers where f(t) controls the forget gate, g(t) and i(t) control the input gate and o(t) controls the output gate. All the gates receive both the input X(t) and the short-term state vector h(t-1) from the previous time step. Only the forget gate and the output gate receive additionally the long-term state vector c(t-1).

The difference in complexity between the two structures is immediately visible, in the simple RNN structure there is only one output which is then propagated to the next time step and the combined input at any time step includes the input signal at that time step and the output from the previous time step. In contrast the LSTM cell is quite complex and consists of h(t) and c(t). In case of the long-term state vector, information is received from the previous time step c(t-1), this vector contains the representations learnt not only in the previous time step but from a longer time horizon. This vector is manipulated by the forget gate where certain experiences are dropped and new experiences are added to it by the input gate. This modified vector is then sent to the next time step. A copy of the same is made and is sent to the output gate where it is manipulated further by the output gate which also takes an additional input from o(t). The final output from this gate represents the new short-term state h(t) and the output at the current time step.

The above-mentioned structure enables the LSTM to store inputs and manipulate them over both the long-term and short-term which solves the problem of vanishing gradients as seen with the simple RNN structure. This makes the LSTM useful in handling sequential data with long term dependencies.

The entire process mathematically can be explained as follows:

$$\mathrm{i(t)} = \sigma(W_{xi}^{T} x(t) + W_{hi}^{T} h(t-1) + \mathrm{b_i}) \quad (7)$$

$$\mathrm{f(t)} = \sigma(W_{xf}^{T} x(t) + W_{hf}^{T} h(t-1) + \mathrm{b_f}) \quad (8)$$

$$\mathrm{o(t)} = \sigma(W_{xo}^{T} x(t) + W_{ho}^{T} h(t-1) + \mathrm{b_o}) \quad (9)$$

$$\mathrm{g(t)} = \tanh(W_{xg}^{T} x(t) + W_{hg}^{T} h(t-1) + \mathrm{b_g}) \quad (10)$$

$$\mathrm{c(t)} = f(t)c(t-1) + i(t)g(t) \quad (11)$$

$$\mathrm{y(t)} = \mathrm{h(t)} = o(t)\ tanh\ (c(t)) \quad (12)$$

where X(t), Y(t), c(t), h(t), c(t-1), h(t-1), g(t), i(t), o(t), f(t), tanh and σ (sigma activation function) are the same as described previously. Wxi, Wxf, Wxo, Wxg, represent weights of the layers i(t), f(t), o(t) and g(t) respectively for the input signal x(t). Whi, Whf, Who, Whg represent weights for the same layers with for the short-term state h(t-1); bi, bf, bo and bg are biases. The right-hand side of equations 5 and 6 represent element wise multiplications.

For machine learning purposes there are numerous LSTM architectures available. An architecture is recognised by its unique arrangement of LSTM layers, encoding and decoding of data and its combination with other existing deep learning models such as the CNN explained earlier. The issues faced while using CNN based deep learning models for forecasting are long training times and variability of the evaluation metrics when the model is run several times. Keeping this in mind a specific architecture of the LSTM, namely L-AE was chosen since it has the reputation of being the quickest when it comes to fitting a forecasting model which is also verified by the results that will be shown. Moreover, the principle behind the L-AE enables formation of accurate forecasting models. The principle is as follows, the L-AE has two parts which are the encoder and decoder. The encoder manipulates the input vector and creates an encoded vector that contains the complex dynamics within the temporal ordering of the input. While doing so it also reduces the dimensionality of the input vector. This particular vector is then decoded by the decoder in order to recreate the input vector. The encoded vector can also be used to train other machine learning algorithms which provides added flexibility in its usage.

The encoding and decoding process is shown in equations (13) and (14).

$$y = \sigma(W_1 x + \mathrm{b}_1) \tag{13}$$

$$x' = \sigma(W_2 y + \mathrm{b}_2) \tag{14}$$

Here, the weights associated with encoding and decoding are W1 and W2 whereas, their biases are b1 and b2. The activation function is represented by σ and the encoded vector is represented by y, the input vector is x and x' is the recreated input vector after decoding.

The L-AE model used in this thesis is shown in Fig. 11. The encoder part of the model consists of the input layer and an LSTM layer whereas the decoder part of the model includes an LSTM layer and a DNN. An important part of the model which links both the encoder and the decoder is the RepeatVector. It enables the decoder part of the model to read the encoded vector.

As it can be seen from the figure, the input given to this model has to be in the form of [samples, steps, features]. The number of samples is "none" because the model is able to calculate the total number of samples fed to it automatically. The number of steps is 1, since at any time one row of data is fed to the model and the number of features is 6 which are module temperature (^{0}C), output power (W) and irradiation (W/m^2) from the previous two time-steps. The first LSTM layer receives the input data in the above-mentioned form and creates an encoded vector with 1 feature. This is repeated 15 times since there are a total of 15 LSTM cells in the layer. This vector is then manipulated by the RepeatVector to a form that can be utilised by the decoder. The LSTM layer of the decoder takes the output of the RepeatVector and attempts to recreate the input vector by ascertaining the relationship between the input and output data. This layer is then followed by a DNN which prints the output.

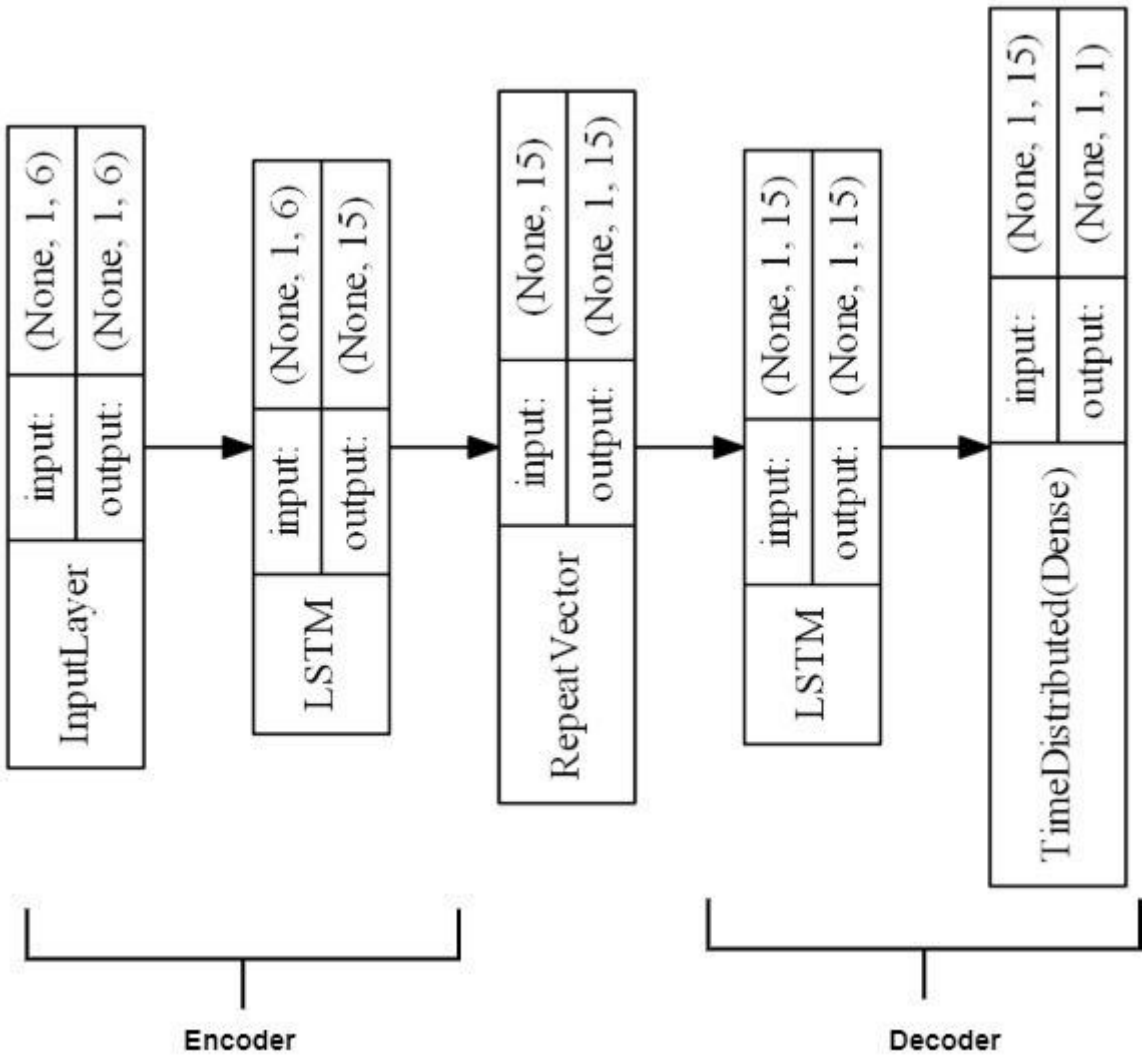

Figure 11. L-AE architecture

The data pre-processing and the evaluation metrics used are the same as described previously.

2.5 ARIMA

A forecasting model that is quite often employed for short term forecasts is the ARIMA model. As explained in the introduction chapter it is utilised not only for the short term but also medium and long term. It also serves as a benchmark in this study to compare the results with.

The ARIMA model can be defined as a regression model characterizing a linear relationship between a variable and its past values[13], [14]. For the ARIMA model to be utilised certain conditions have to be fulfilled such as stationarity of the input signal and determination of the Autoregressive (AR) and Moving Average (MA) coefficients. These determinations for the input signal in this study will be explained later on. The mathematical basis of the model is as follows:

$$y_t'^{AR} = \sum_{i=1}^{m} \emptyset_i x_{t-i} + \omega_t = \emptyset_1 x_{t-1} + \emptyset_2 x_{t-2} + \cdots + \emptyset_m x_{t-m} + \omega_t \quad (11)$$

$$y_t'^{MA} = \sum_{j=0}^{n} \theta_j \omega_{t-j} = \omega_t + \theta_1 \omega_{t-1} + \theta_2 \omega_{t-2} + \cdots + \theta_n \omega_{t-n} \quad (12)$$

$$y_t'^{ARMA} = \sum_{i=1}^{m} \emptyset_i x_{t-i} + \sum_{j=0}^{n} \theta_j \omega_{t-j} \quad (13)$$

where $y_t'^{AR}$, $y_t'^{MA}$, and $y_t'^{ARMA}$ are the autoregression (AR), the Moving average (MA), and the Autoregression moving average (ARMA) time series values respectively. $\emptyset_i$ is the autoregressive coefficient and θ_j is the moving average coefficient. ω_t is the noise.

The autoregressive (AR) value characterizes the current value in a linear relationship with its past values and the noise ω_t. It is represented in Equation (13). The Moving average part is a combination of previous individual noise components, which is used to create a time series, as shown in Equation (4). ARMA is a combination of both AR and MA [15].

The parameters of the model represented by m and n determine the number of past values that should be taken into account for the AR and MA parts of the model. The Partial Autocorrelation Function (PACF) and Autocorrelation Function (ACF) are used to determine the appropriate number of lags for the AR and MA parts of the model. The PACF provides a correlation between a value of the time series and its lag whereas the ACF provides a correlation between an error of the value and lagged error values of the time series. Another parameter to be determined for the ARIMA model is the differencing. This is done to ensure that the signal is stationary. If the input time signal has a trend, it can be differenced one or more times until a stationary signal is obtained. The stationarity can be confirmed in many ways. It can be done visually wherein a clear absence of a trend indicates stationarity, it can be ascertained through the ACF and PACF plots where the subsequent values are low and oscillate around zero and finally it can be determined through stationarity tests such as the Augmented Dickey Fuller (ADF) test and others.

Fig. 12 gives a plot of the entire data set of the output power from the monocrystalline solar panel available at the Wroclaw University of Science and Technology. It also presents the ACF and the PACF plot for 20 lags of the time series data. While the PACF plot reduces to zero after 2 lags, the AFC plot never reaches zero. It can also be seen that the output power in the plot has a clear trend to it. Hence, it can be concluded that the signal is non-stationary.

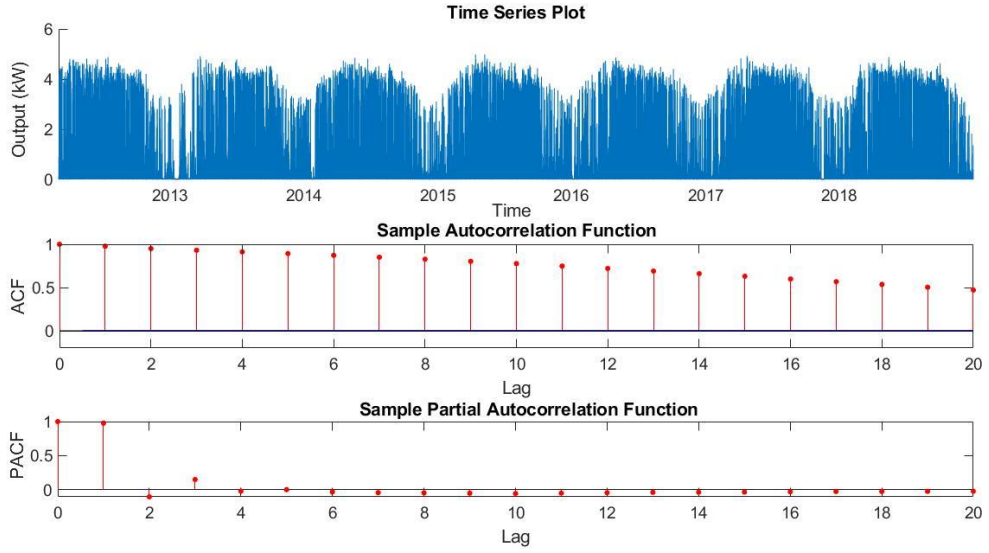

Figure 12. Solar panel output data with an auto correlation function (ACF) and a partial auto correlation function (PACF) analysis.

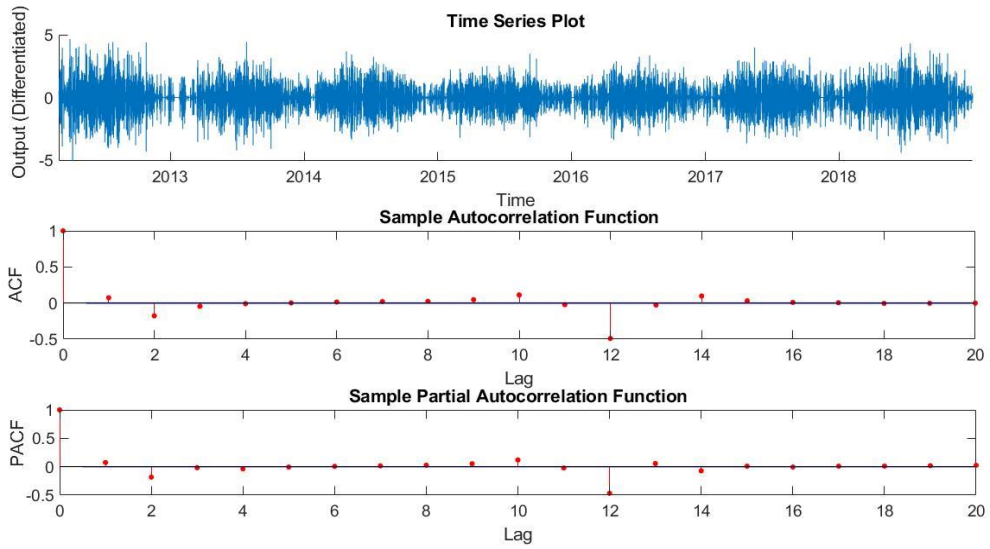

Figure 13. Differentiated output with ACF and PACF analysis

Fig. 13 provides that same analysis as Fig. 12 but with the differentiated time series input. Firstly, visually the output signal now does not have a trend and moreover it can be seen that the ACF and the PACF definitively reduce to zero and stay close to it after 2 lags. Furthermore, the ADF test performed with the differentiated signal resulted in a p-value of 0.001 which confirms stationarity. Hence, an ARIMA model with a maximum AR lag of 2, MA lag of 1 and 1 differencing can be chosen for short-term solar forecasts.

2.6 Results

2.6.1 10 – min ahead forecasting

Previously, the training time for one run (fitting the model, prediction and evaluation) of the 3 above mentioned models for the input data of the same size were 1min 14 secs, 2 min 50 secs and 2 min 10 secs for the CNN, multi – headed CNN and CNN – LSTM models respectively. When the same training time was calculated for the input data of the same size for the L-AE model, it was 37 seconds making is significantly faster than the earlier models explored. The ARIMA model is the quickest of all as expected and the data of the same length as the other models is fit in 14 seconds.

Due to the quick training times of the L-AE and the ARIMA, it becomes possible to conduct extensive analysis on its variability. The model was run 1000 times with the training and validation data changing every time. This was possible by randomly sampling the data set used for validation and keeping the rest for model training. This process of evaluating the reliability of the model is the same as what was performed with the CNN based models.

For the L-AE, mean values for the RMSE and MAE over all the runs are 0.224 kW and 0.089 kW respectively, these represent 4.48% and 1.78% of the nominal power of the panel under consideration. Moreover, it is evident from Fig. 14 (left) and (right) that the frequency distributions of the evaluation parameters are near gaussian enabling the calculation of confidence parameters. The standard deviations of both parameters are 0.053 kW and 0.021 kW representing 1.06% and 0.42% of the nominal power which shows that the variability of the error value is not significant. Hence, the results are consistent. The 95% confidence intervals for the RMSE and MAE parameters are [0.113, 0.33] kW and [0.047, 0.131] kW respectively. This indicates that whenever such a model is run, it can be said with 95% confidence that the best model would have an RMSE of 0.113 kW and an MAE of 0.047 kW

and the worst model would have an RMSE of 0.33 kW and an MAE of 0.131 kW. In general, it can be seen that the RMSE is higher than the MAE values due to the nature of the RMSE to punish bigger errors more severely than the MAE which has the tendency to treat all errors equally.

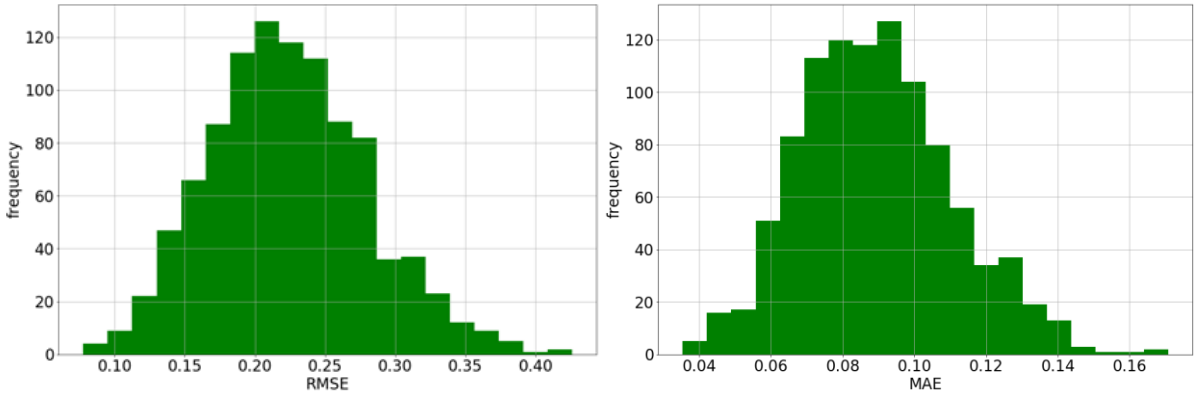

Figure 12. Frequency distributions of RMSE (left) and MAE (right)

For the ARIMA, the same underlying data was fit, and it was run 1000 times with changing test and validation data sets. The parameters of the model are (2,1,1) where 2 is the Autoregressive (AR) coefficient, 1 represents the fact that the time series was differenced once in order to obtain stationarity and 1 is the Moving Average (MA) coefficient. It must be mentioned that the coefficient values of the ARIMA model were changed in every run according to the data, but the order of the model (2,1,1) remained the same. This was kept for the purposes of fair comparison since different deep learning models are not used for different seasons and a similar approach is used with the ARIMA model. The mean RMSE and MAE values are 0.374 kW and 0.181 kW respectively, these are 7.48% and 3.6 % of the nominal power of the panel which is higher than the corresponding values of the LSTM – autoencoder model. The standard deviations for the same are 0.22 kW and 0.105 kW at 4.4% and 2.1 % implying that the spread of the error values is also higher than that of the LSTM – autoencoder model. The 95% confidence intervals for the RMSE and the MAE are [0.36, 0.38] kW and [0.16, 0.20] kW which are higher than the intervals for the LSTM – autoencoder model but with a lesser spread.

Fig. 13 presents the forecasting made every 10-mins for the entire day for 3 models under consideration. These are the L-AE model, the ARIMA and the best performing model from the CNN based models (CSS-LSTM). 4 random days of the year are considered for this purpose. The days are quite representative with regards to the different seasons in a year seen in Wroclaw. The figure on the upper left and lower left are typical for the summer and days with high availability of sunlight since the peak power during the day reaches 4 kW and above. The figures on the upper and lower right are more typical for the winter and low sunlight days when the power output remains quite low. The models achieved are able to work reliably despite the change in seasons hence not requiring training different models based on different seasons of the year. The accuracy of all models reduces around the peak values irrespective of the day under consideration but it can be seen that the L-AE (LSTM_ae) performs slightly better when compared to the ARIMA and the CNN-LSTM models at the peak.

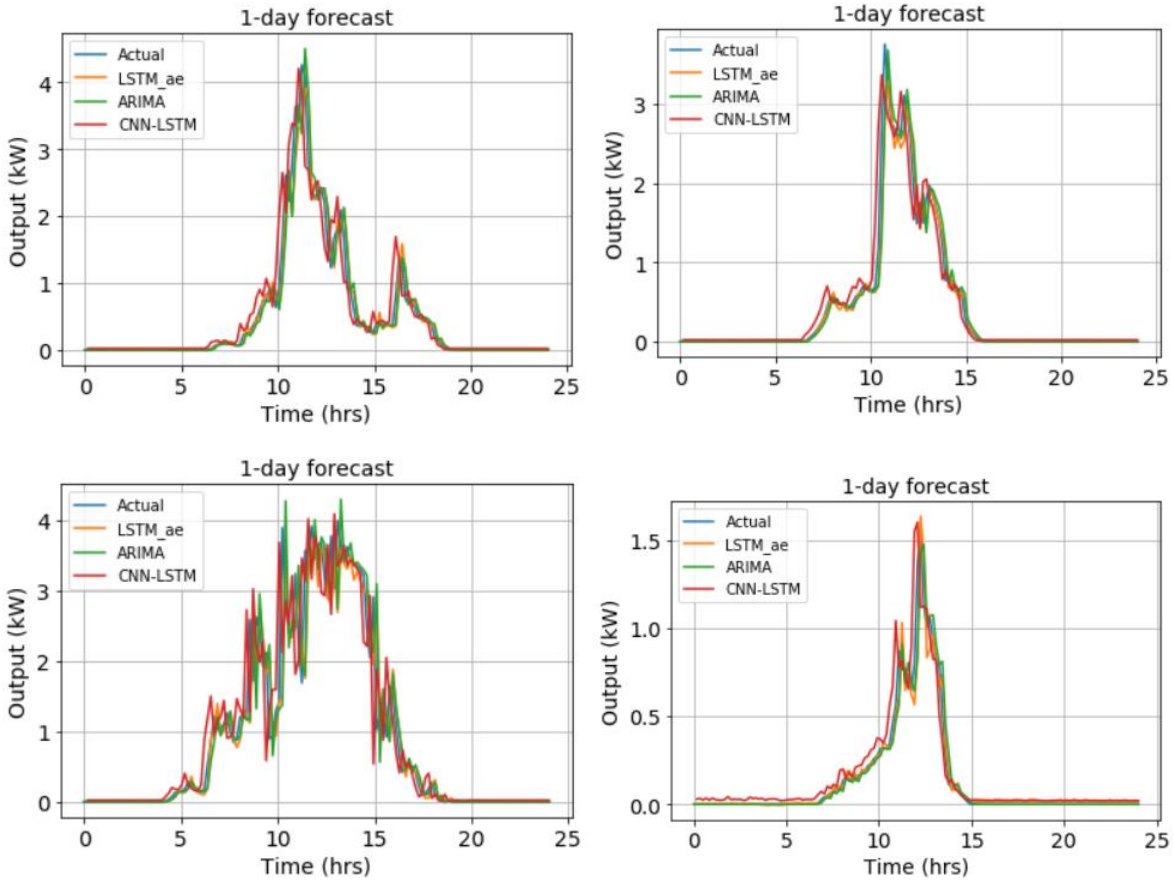

Figure 13. 1-day forecasts for 4 random days of the year with the forecasts made every 10-min

2.6.2 1 – hour ahead forecasting

To obtain an appropriate ARIMA model for the 1-hour averaged output data, analysis must be carried out to obtain the AR, MA and the differencing orders. To do so, the ACF and PACF plots must be obtained and the ADF test performed. Fig. 14 shows the 1-hour averaged data and its associated ACF and PACF plots. It is clearly visible that the output data and the ACF plot have a trend. The PACF plot, while not having a clear trend shows minor oscillation about the x-axis.

To remove the trend and obtain a stationary signal, the output is differentiated, and the results are shown in Fig. 15. In this case the differentiated output now does not posses a clear trend and mainly oscillates about the x-axis. The ACF function now does not possess a trend and the outputs reduce to 0 after 2 lags hence the MA coefficient can be a maximum of 2. The PACF plot reduces to a zero after 1 lag but it is seen that a minor trend does exist like the case when the signal was not differentiated. More differencing did not result in the removal of the mentioned trend hence an order of 1 for the differencing is adopted. The maximum value of the AR order is taken as 1. The final model used to fit the data is (1,1,2).

Like what was done previously with the CNN based models, both the L-AE and the ARIMA models were applied for the 1-hour ahead forecasting. In this case both models were run 100 times with the training and test data changing in every run. The mean RMSE values for the L-AE and ARIMA models are at 0.271 kW and 1.10 kW which are 5.42% and 22 % of the peak value. In case of the MAE the L-AE model has a mean value of 0.146 kW which is 2.92 % of the peak value and the ARIMA model has a mean MAE value of 0.630 kW which is 12.6 % of the peak value. From the previous section the metrics for the CNN-LSTM model are 0.282 kW

and 0.152 kW for the RMSE and MAE, respectively representing 5.64% and 3.04 % of the peak value.

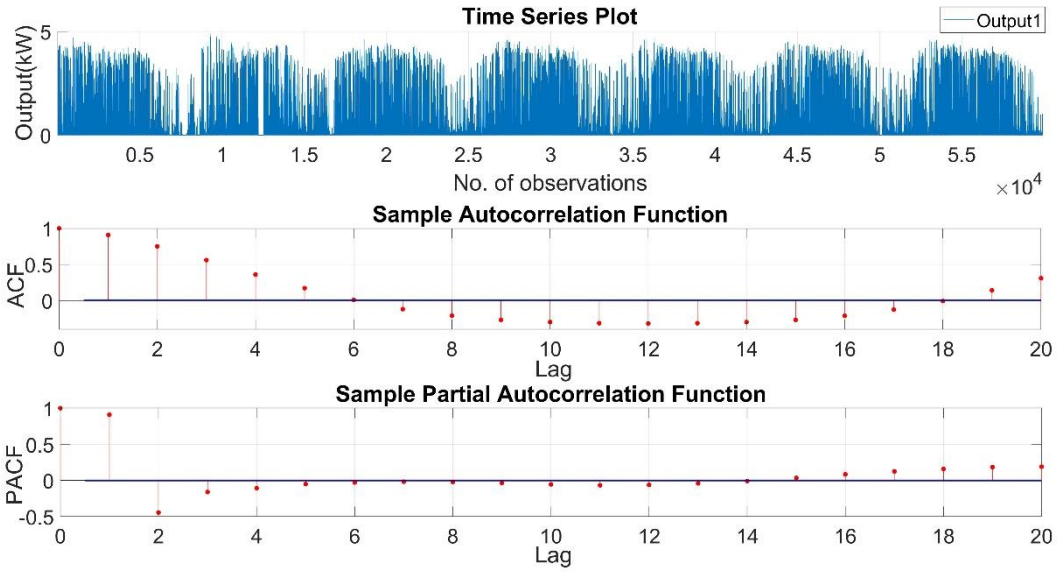

Figure 14. Solar panel output data (1hr) with an ACF and PACF analysis.

It can be said that for all models, compared with their performance for the 10-min ahead forecasting, the error metrics have become worse which is expected since the horizon of forecasting has been increased. The L-AE model performs the best out of all models in terms of evaluation metrics values for both forecasting horizons. It is noticeable that the performance of the ARIMA model has been significantly affected in the 1-hour ahead forecasting. A reason for this could be the use of a single model for all seasonal variations. This was left unchanged since the deep learning models used have been fit with nearly the entire data set in every run and not season specific data for each season. This is a question to be tackled during future work under this research topic.

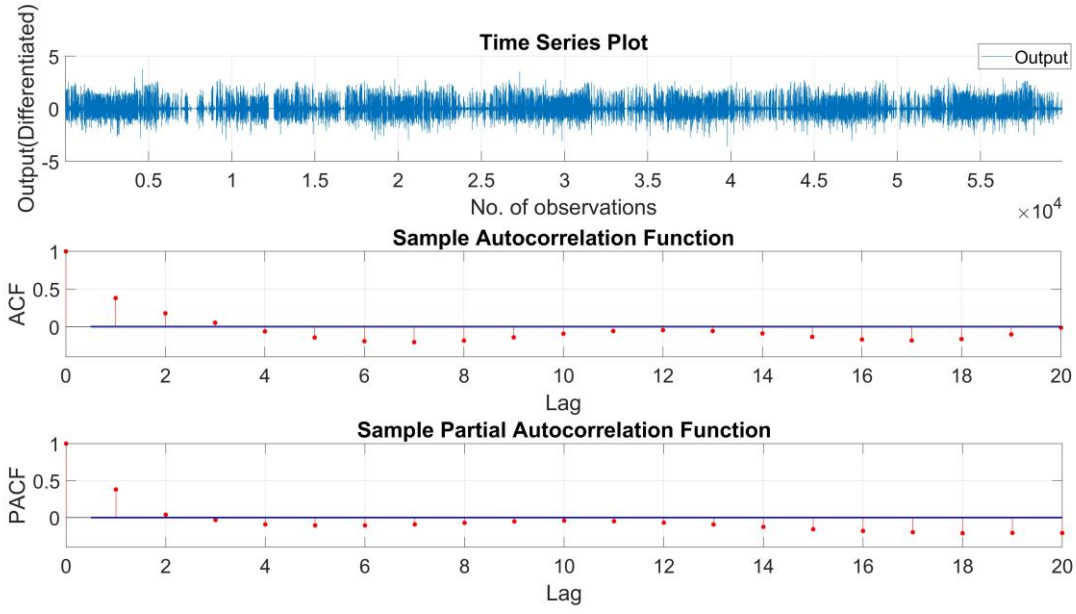

Figure 15. Solar panel output data (differentiated) with an ACF and PACF analysis.

Fig. 16 presents 2 figures from the 1-hr ahead forecasting for the CNN-LSTM, L-AE and the ARIMA model for two randomly selected days of high and low solar availability. It can be seen in general that the performance of all forecasting models has dropped when compared to the 10-min ahead forecasting results. From the figure in the left it is seen that the CNN-LSTM and the L-AE model perform better than the ARIMA model with the L-AE model performing the best. Inaccuracies are observed more when the output power moves towards the peak than away from it for all models used. The ARIMA model overshoots at the peak, the CNN-LSTM underestimates the peak whereas the L-AE model is the closest to the actual peak with some underestimation. For the figure on the right, it represents a day with lower solar availability with considerable changes in irradiation. All methods are inaccurate at the peaks again with the ARIMA overpredicting whereas the others underpredicting. Inaccuracies again are more prominent during the increase of output power than decrease of the same.

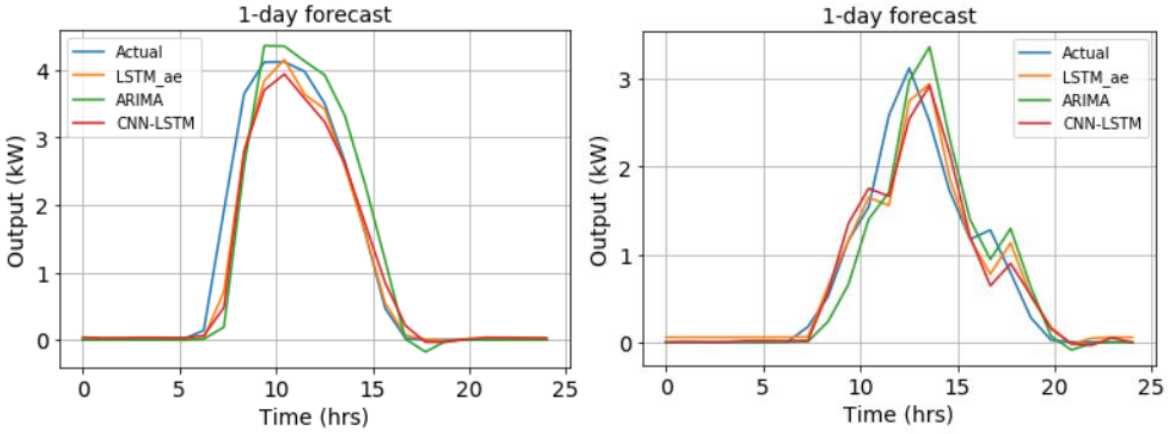

Figure 8. 1-day forecasts for 2 random days of the year with the forecasts made every 1 hour

Conclusions

In this chapter a total of 4 deep learning models and 1 conventional model for forecasting were considered for the 10-min ahead and the 1-hour ahead forecasting. Each deep learning model and the ARIMA, its structure and the associated data processing have been explained comprehensively. It is noticeable that the accuracy for all the models reduce as the time horizon increases from 10 minutes to 1 hr. It is also clear that amongst all the models for both the 10 – minute and 1 – hour ahead forecasts the L-AE model is not only the most accurate but also the fastest except the ARIMA which is faster but less accurate. It is also more reliable since the metrics were obtained as a result of the running the program for 1000 times for the 10-minute ahead forecasts and 100 times for the 1-hour ahead forecasts with changing test and train data in every run. It is noticeable from the figures that the CNN based models may not be suitable for the 1 hour ahead forecasts but the L-AE model is able to show better performance with regard to the 1 hour ahead forecasts. The area of deep learning is ever expanding and intensively researched.

References

[1] A. Geron, *Hands-On Machine Learing With Scikit-Learn & Tensor Flow*. 2017.

[2] N. Andrew, “Machine Learning Yearning, Technical Strategy for AI Engineers In the Era of Deep Learning,” 2018.

[3] Y. Bengio, “Deep Learning of Representations for Unsupervised and Transfer Learning,” *JMLR Work. Conf. Proc.*, vol. 7, pp. 1–20, 2011.

[4] C. M. Bishop, *Pattern recognition and machine learning / Christopher M. Bishop.* 2006.

[5] P. Luukkonen, P. Bateman, J. Hiscock, Y. Poissant, and D. H. and L. Dignard-Bailey, “International Energy Agency Co-Operative Programme on Photovoltaic Power Systems,” p. 2013, 2013, [Online]. Available: http://www.cansia.ca/sites/default/files/201306_cansia_2012_pvps_country_report_long.pdf.

[6] D. Yang, J. Kleissl, C. A. Gueymard, H. T. C. Pedro, and C. F. M. Coimbra, "History and trends in solar irradiance and PV power forecasting: A preliminary assessment and review using text mining," *Solar Energy*, vol. 168. pp. 60–101, 2018, doi: 10.1016/j.solener.2017.11.023.

[7] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Prog. Artif. Intell.*, vol. 9, no. 2, pp. 85–112, 2020, doi: 10.1007/s13748-019-00203-0.

[8] V. Suresh, P. Janik, J. Rezmer, and Z. Leonowicz, "Forecasting solar PV output using convolutional neural networks with a sliding window algorithm," *Energies*, vol. 13, no. 3, 2020, doi: 10.3390/en13030723.

[9] A. Ajit, K. Acharya, and A. Samanta, "A Review of Convolutional Neural Networks," *Int. Conf. Emerg. Trends Inf. Technol. Eng. ic-ETITE 2020*, pp. 1–5, 2020, doi: 10.1109/ic-ETITE47903.2020.049.

[10] J. Heaton, "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning," *Genet. Program. Evolvable Mach.*, 2018, doi: 10.1007/s10710-017-9314-z.

[11] P. Mathur, M. P. Ayyar, R. R. Shah, and S. G. Sharma, "Exploring classification of histological disease biomarkers from renal biopsy images," *Proc. - 2019 IEEE Winter Conf. Appl. Comput. Vision, WACV 2019*, pp. 81–90, 2019, doi: 10.1109/WACV.2019.00016.

[12] Y. Bengio, I. J. Goodfellow, and A. Courville, "Sequence Modeling : Recurrent and Recursive Nets," *Deep Learn.*, 2015.

[13] M. H. Alsharif, M. K. Younes, and J. Kim, "Time series ARIMA model for prediction of daily and monthly average global solar radiation: The case study of Seoul, South Korea," *Symmetry (Basel).*, vol. 11, no. 2, pp. 1–17, 2019, doi: 10.3390/sym11020240.

[14] S. Atique, S. Noureen, V. Roy, V. Subburaj, S. Bayne, and J. MacFie, "Forecasting of total daily solar energy generation using ARIMA: A case study," *2019 IEEE 9th Annu. Comput. Commun. Work. Conf. CCWC 2019*, no. January, pp. 114–119, 2019, doi: 10.1109/CCWC.2019.8666481.

[15] Mahmoud Ghofrani and Musaad Alolayan, "Time Series and Renewable Energy Forecasting," *Intech*, p. 13, 2016, doi: http://dx.doi.org/10.5772/intechopen.70845.

3. Energy Management (Optimal Power Flow)

Energy management is related to the optimization process mentioned in the introduction section. The foundation of modern energy management systems rooted in solving ED and OPF problems can be traced back to an older and traditional power flow in energy systems. Power flow also known as load flow represents a steady state analysis of the power system which helps in both continuously monitoring the system and plan for future activities related to the same such as capacity expansion[1]. It determines various AC power parameters of the network such as voltage magnitudes, voltage angles, active power and reactive powers in every bus or node of the power system. It provides a summary of the system in terms of the one-line diagram of the network, electrical parameters of all elements that are taken into account to determine the steady state solution such as generators, load centres, impedances of transmission and distribution lines, shunt compensation and transformer tap changers amongst others and also provides information about line losses occurring in the system. Overtime, this study has been developed in many ways and has resulted in numerous approaches to not only obtain the steady state solution of the network but also to optimize it. Descriptions of these approaches have been given in the introduction section.

In this chapter, a brief approach to power flow, a detailed explanation of OPF, a comprehensive description of a hybrid meta-heuristic optimizer developed to solve the OPF, a comparative analysis of the performance of the optimizer developed with other approaches and finally application of the same for ED problems of microgrids will be presented.

3.1 Power flow

Power flow is an important tool for power system analysis and provides a plethora of information for the operator of the power system. At present there are numerous commercial and academic software that provide power flow analysis but at the heart of all these tools are common mathematical approaches based on commonly used numerical methods that are used to solve for the steady state of the power network.

The description of the power flow problem along with the commonly used numerical methods and their application to the problem are as follows.

3.1.1 Mathematical description of the problem

Power flow or load flow analysis involves solving for certain AC parameters at every node or bus of the power system. These parameters are the voltage magnitudes (V_i), voltage angles (δ_i), active power (P_i) and reactive power (Q_i) where $i = 1,2,\dots,n$ denotes the number of buses in the power system [1], [2]. It can be noticed immediately that with an increase in the size of the network (number of buses), the number of variables to be solved increases 4folds making the problem computationally intensive with scale.

The problem usually involves categorizing the buses present in the power system which define the electrical parameters are known at the bus and the parameters that are to be determined. This process reduces the total number of variables to be determined while finding a steady state solution to the network [1], [2]. The bus types are as follows:

Load bus (PQ): In this category of buses mainly representing load centres, the total active and reactive power generated and consumed is known. From this information the net active power (P_i) and reactive power (Q_i) are obtained which are the known variables and the variables to be determined are the voltage magnitude (V_i) and voltage angle (δ_i).

Generator bus (PV): In this category of buses mainly representing generating stations, the net active power (P_i) and the voltage magnitude (V_i) are known. The voltage is maintained at a specified level by the generating station by means of reactive power control and other methods. The variables to be determined include the reactive power (Q_i) and the voltage angle (δ_i).

Swing bus/slack bus/reference bus: Known by different names, this bus is usually reserved for the largest generating station in the power system. It is in fact a reference created by a power system professional to aid the process of finding a steady solution to the power network. In theory this bus is deemed capable of supplying active and reactive power according to the needs of the power system to keep it in balance. The variables known are the voltage magnitude (V_i) and voltage angles (δ_i) whereas the ones to be determined are active power (P_i) and reactive power (Q_i).

A table denoting the type of bus and its respective known and unknown variables is shown below.

Table 1: Types of buses and their variables characteristics

Bus type	Active Power (P_i)	Reactive Power (Q_i)	Voltage Magnitude (V_i)	Voltage Angle (δ_i)
Load bus (PQ)	Known	Known	Unknown	Unknown
Generator bus (PV)	Known	Unknown	Known	Unknown
Swing bus (Vδ)	Unknown	Unknown	Known	Known

Once the types of buses are decided, the power flow equations along with the operational and system constraints are defined as shown below

$$P_{gi} = |V_i| \sum_{k=1}^{n} |V_k||Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i) + P_{di} \quad (1)$$

$$Q_{gi} = |V_i| \sum_{k=1}^{n} |V_k||Y_{ik}| \sin(\theta_{ik} - \delta_k - \delta_i) + Q_{di} \quad (2)$$

$$P_{gi}^{min} \leq P_{gi} \leq P_{gi}^{max} \; \forall \; i \; \epsilon \text{ No. of nodes} \quad (3)$$

$$Q_{gi}^{min} \leq Q_{gi} \leq Q_{gi}^{max} \; \forall \; i \; \epsilon \text{ No. of nodes} \quad (4)$$

$$V_i^{min} \leq V_i \leq V_i^{max} \; \forall \; i \; \epsilon \text{ No. of nodes} \quad (5)$$

$$\delta_i^{min} \leq \delta_i \leq \delta_i^{max} \; \forall \; i \; \epsilon \text{ No. of nodes} \quad (6)$$

Equations (1) and (2) are the power balance equations for both active and reactive power which must be always maintained in a power system. In (1), $|V_i| \sum_{k=1}^{n} |V_k||Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$ represents active power losses whereas in (2), $|V_i| \sum_{k=1}^{n} |V_k||Y_{ik}| \sin(\theta_{ik} - \delta_k - \delta_i)$ represents reactive power losses. Equations (3)-(6) maintain system parameters like voltage magnitude (V_i), reactive power (Q_i), voltage angle (δ_i) and active power (P_i) within mandatory operational bounds. V_i and V_k represent the voltages at nodes i and k respectively; Y_{ik} and θ_{ik} are the admittance and admittance angle between buses i and k; P_{gi}, P_{di}, Q_{gi} and Q_{di} represent active power and reactive power supply and demand at nodes i; δ_i and δ_k represent voltage angles at buses *i* and *k*. (3) constraints the active power output of all generators in the system between their minimum and maximum. (4) constraints the reactive power of the same within their limits and (5) and (6) constraints the voltage magnitudes and voltage angles within their minimum and maximum.

3.1.2 Numerical methods for power flow in meshed networks

A. Gauss – Seidel method (GS)

This approach to solving a set of non-linear equations is one of the simplest and earliest. The process begins with an assumption of a solution vector and then it moves towards the actual solution in an iterative manner. The assumption of a solution vector is very significant for this approach and determines both convergence and its speed. In every iteration an updated value of a variable is obtained through substitution of the remaining values in equation (1) and (2). The solution vector is then updated with this new value. This process continues until the final solution is within acceptable limits. This method is not computationally intensive since it does not involve storing of data from one iteration to the next and since there is no computation of a gradient. The application of the method to obtain a solution for the power flow problem is as follows:

- All buses other than the swing bus are assumed as load (PQ) buses. This results in both the net active power (P_i) and reactive power (Q_i) to be known at $(n-1)$ buses where n is the total number of buses. At the slack bus representing the largest generator in the system, the voltage magnitude and angle are known while the active and reactive power are allowed to vary.
- The bus admittance matrix is calculated using the line and shunt admittance data
- The iterative process is initiated with a flat voltage start where the voltage magnitude and angle at all buses are set to $1\angle 0^o$. The voltages at all buses except the slack bus are then recalculated with equations (1) and (2). This process is carried on until the next iterative voltage value is smaller than a pre-determined small value.

$$v_i^{p+1} - v_i^p < \varepsilon \tag{7}$$

- Once the voltage values are determined the active power and reactive power values at the slack bus are determined.
- The process is completed with calculation of line losses with the system line and shunt admittance data together with the new voltage values.

The above described process is one of many approaches to obtaining the steady state solution of the power system. It begins with an assumption that all buses except slack are PQ or load buses. If the buses were assumed to be PV or generator buses instead the process would change as the information that is known at the beginning are P_i and V_i whereas Q_i and δ_i would need to be determined.

B. Newton-Raphson method (NR)

This method represents an approach which is quite popular in the field of mathematics to solve a set of non-linear equations. It is efficient and fast when compared to the earlier described Gauss-Seidel method. This is because the method involves computation of a gradient that directs the solution searching process towards the actual solution. The method is not sensitive to the choice of an initial solution vector and has a higher rate of convergence than the Gauss-Seidel method. Despite its advantages, it poses challenges with regard to the computational burden imposed since values of the previous iteration have to be calculated and Jacobian matrices need to be formed [1], [2]. The Jacobian consisting of partial derivatives of different variables after inversion is the gradient that ensures that the solution is obtained in a few iterative steps. Similar to the previous method the iterations continue until the new voltage values obtained have a change smaller than that of a pre-defined value.

The application of this method for obtaining the steady state solution of the power system is as follows:

- For load or PQ buses, the active and reactive power values are known whereas assumptions of voltage magnitude and voltage angle values are made. Utilising these values and equations (1) and (2) correction values of P_i and Q_i are obtained. The correction values are nothing but the difference between the known values of P_i and Q_i and values of the same obtained from the assumed V_i and δ_i utilising equations (1) and (2).

$$\Delta P_i = P_{specified} - P_{calculated} \quad (8)$$

$$\Delta Q_i = Q_{specified} - Q_{calculated} \quad (9)$$

  Utilising the corrected values and the Jacobian, updated V_i and δ_i values can be obtained.
- At the swing bus the V_i and δ_i values are set to $1\angle 0^o$. Hence, equations pertaining to the same are not included in the Jacobian.
- Once the delta V_i and δ_i values are obtained using the Jacobian, they are added to the assumed V_i and δ_i values and the above-mentioned steps are repeated. This iterative process is repeated until the improvement in voltage magnitude values is smaller than a pre-defined small value.
- For PV buses, the process is similar except for the fact that the known values for them are P_i and $V_i$whereas the Q_i and δ_i values are assumed.

C. Fast Decoupled Load Flow (FDLF)

3.1.3 Power flow (meshed networks) – example and comparison

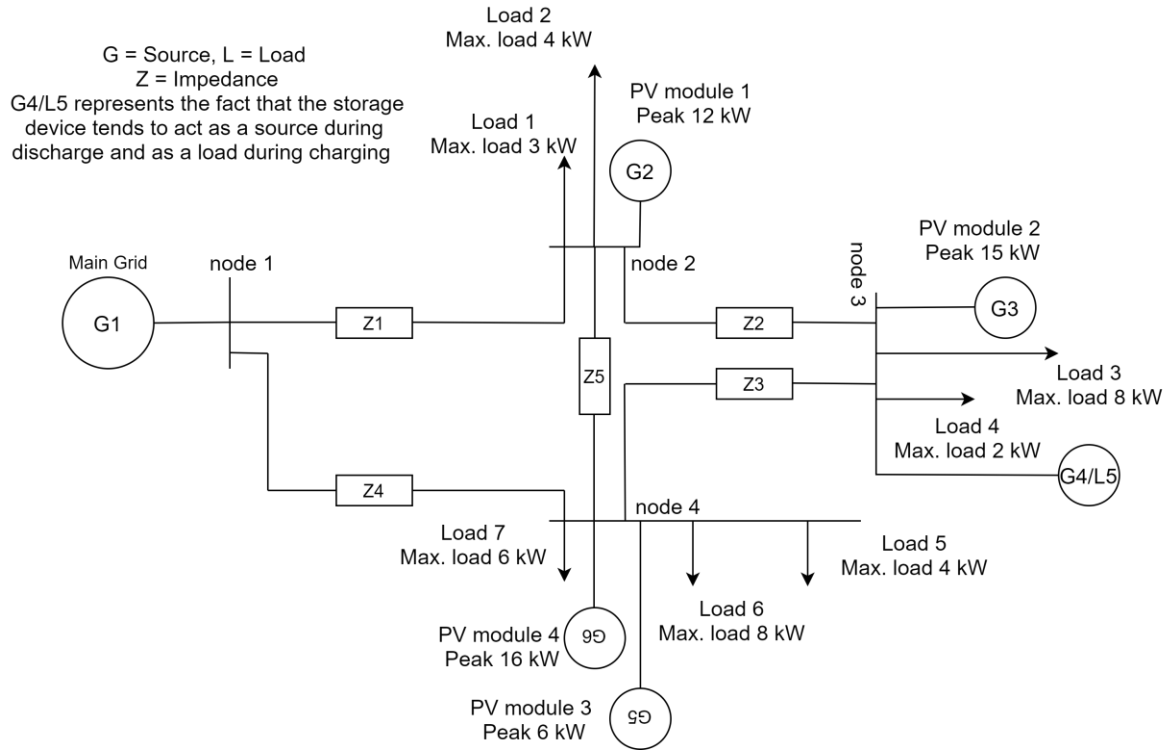

Figure 1: Meshed microgrid network

In power systems, there is an interdependence between active power and voltage angle ($P - \delta$) and reactive power and voltage magnitude ($Q - V$). This results in a weak coupling between

the pairs of variables that enables the power system analyst to solve for the pairs of variables independently thereby simplifying the solution finding process. It reduces the computational burden significantly. This approach utilizes the same procedure as the Newton-Raphson method with the exception of the Jacobian which is now simplified with the elimination of elements with weak coupling. This results in the matrix being sparse. This approach could affect convergence to the true solution but represents a trade-off between accuracy and computation burden which seems to make it acceptable [1], [2].

It has to be mentioned that the above-mentioned approaches to load flow are applicable mainly to meshed networks. For radial networks another set of approaches exist which are explained later on and why they are to be used is also explained. Below an example of the different methods explained above for energy management in a simple microgrid is presented. More complex microgrid models will be utilised later on while describing meta-heuristics and combined optimizers. The power flow in this cases is run every 10 minutes corresponding to the sampling of the load and generator data.

The above microgrid model consisting of numerous sources and loads was chosen to carry out power flow simulations. It consists of 6 generators, 5 of which are solar panels and 1 Battery Storage System. There is a total of 7 loads. The underlying data for all are taken from the real world such as from the PV system at Wroclaw University of Science and Technology and loads from a nearby village. Other loads and PV panels in the model were obtained by using different data manipulation techniques. The battery storage system was modelled according to the recommendations of [3]. While a simple model for the BSS is chosen at this moment, a far more updated model of the same is used later on. The algorithm of battery management in the microgrid is shown in Fig. 2.

The functioning can be split into two cases, when generation within the microgrid is greater than the load within the same and when it is lesser. The BSS is programmed to charge when additional power exists within the microgrid. This usually happens during mid-day when availability of solar irradiation is high. The BSS is programmed to discharge when there is a deficit of power within the microgrid. This usually happens when availability of solar irradiation is low. It must be mentioned that the BSS is programmed to maintain a minimum State of Charge (SOC) of 20% and a maximum of 95%. Discharging below the lower limit and charging higher than the upper limit is prohibited.

The impedances represented by Z of the lines between all nodes of the microgrid have parameters as shown in Table 2. They are aluminium cables with PVC outer sheath with XLPE insulation having 1 core with 120mm^2 cross-section.

Table 2: Cable data

From	To	Distance (m)	r + jx (Ω) (10^{-1})
node 1	node 2	200	0.5060 + 0.1630j
node 2	node 3	110	0.2780 + 0.0900j
node 3	node 4	135	0.3420 + 0.1100j
node 4	node 1	215	0.5440 + 0.1760j
node 2	node 4	150	0.3790 + 0.1230j

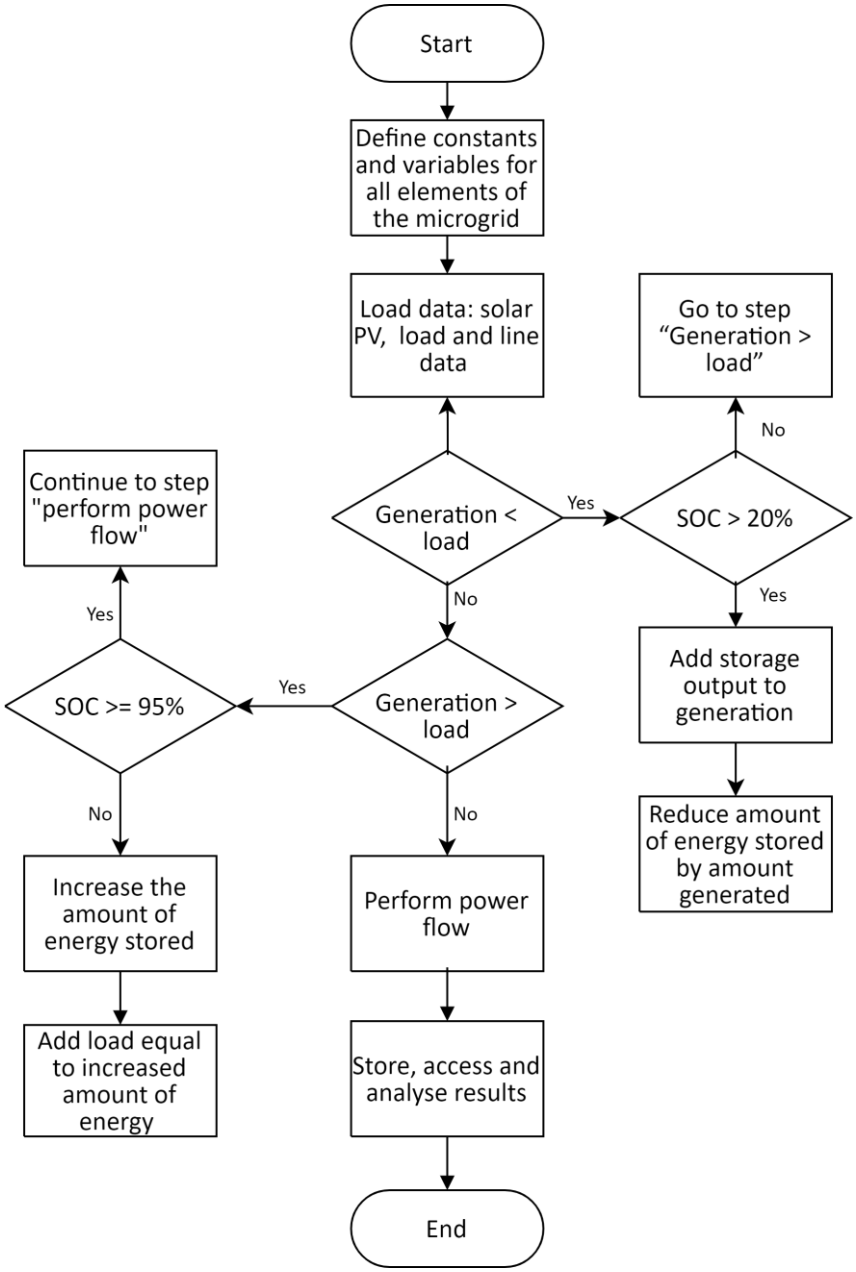

Figure 2: Battery management algorithm

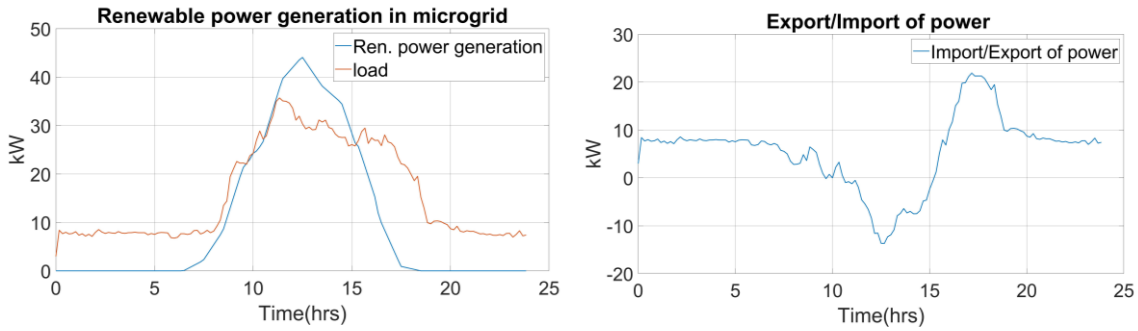

Figure 3: Total renewable power generation and load in the microgrid (left) and exchange of power with the main grid (right)

The total load within the microgrid and the total renewable power generation within the same are shown in Fig. 3 (left) whereas Fig. 3 (right) shows the total amount of power imported through node 1 or PCC of the microgrid. During mid-day, the total renewable power produced

exceeds the load demand of the microgrid and correspondingly it is seen in Fig. 3 (right) that during the same time period power is exported from the microgrid through the PCC. During other times the excess load is satisfied by importing power.

Fig. 4 shows the storage device characteristics during the day of analysis. It can be seen that the device charges when there is excess power available in the microgrid and discharges when there is a deficit.

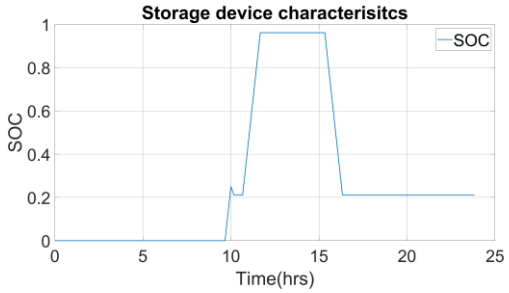

Figure 4: Storage device characteristics

All the above figures were obtained by running power flow at every time step using the NR algorithm. Below, information is provided regarding the run-time when all GS, NR and FDLF are used. Since the network under consideration is small, the 3 algorithms ‘GS’, ‘NR’ and FDLF were run on a week’s data with data sampled every 10 minutes. This resulted in the following performance: the GS method ran the entire schedule in 3.82 s, the NR method ran the same in 2.818 s and the FDLF in 3.145 s. It can be seen that the NR method is the quickest of all for the system presented in Fig. 1 whereas the slowest is the GS. This is mainly because of the gradient that is used for finding the solution in the former and not in the latter. The FDLF is closer in its performance to the NR method but it has to be remembered that the memory needed to run the FDLF is lesser when compared to the NR. Given advances in computer hardware and processing power and the fact that the system under consideration is small, this improvement that the FDLF offers is not significant.

3.1.4 Methods for power flow in radial networks

While the methods mentioned above are suitable for meshed networks, they are not quite suited for radial networks. Radial networks are weakly meshed and have (R/X) ratios that are high, this causes divergence during the iterative process and can result in not reaching the solution. The methods described earlier are also not suitable when there are numerous unbalanced loads within the networks and distributed energy sources [4]. Hence, to overcome the above-mentioned problems another set of methods called the forward/backward methods are utilised. They are of the following types:

- Current summation method [4]–[6]
- Power summation method [5], [6]
- Admittance summation method [5]

The methods discussed are in reference to figure shown below.

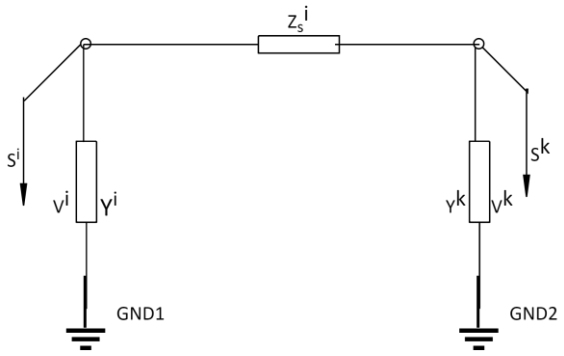

Figure 5: Branch representation: sending bus and branch identified by *i*, receiving branch identified by *k*

A. Current Summation (CS) method

- The method is initiated with a flat voltage start wherein the voltage value at iteration 1 at all the nodes is set to $1\angle 0^o$.
- The current flow in branch *i* is calculated as a sum of the currents due to the load demand at the receiving end and the admittance of the elements at the same. It is described mathematically in (10)

$$I^i = \frac{S^k}{V^k} + Y^k.V^k \tag{10}$$

- Backward sweep: Current summation is now carried out across all branches starting from the back end and moving towards branch 1 of the network. Taking example of the figure shown above, if the current in branch *i* is I^i and the current in branch k after node k can is I^k then the summation of currents across the two branches are described by the equation shown below.

$$I^{sum} = I^i + I^k \tag{11}$$

- Forward sweep: Now that the branch current values, impedance values and the sending node voltage at node 1 are known. The receiving node voltages across all nodes in the network can be calculated as shown in equation (12).

$$V^k = V^i - Z_s^i.I^i \tag{12}$$

- Once the voltage values are determined throughout the network an iteration is complete and the entire is process is repeated again and the difference in voltage values between successive iterations is calculated. If the difference between the voltage values in successive iterations is less than a pre-determined small value, the process is stopped, and the values obtained are taken to be the final values. Equation (13) presents the stopping criteria where *p* is the number of iteration and n represents branch number and the maximum difference is calculated amongst the differences of the voltages between successive iterations which is then compared to the pre-determined small value ε.

$$Max\ (V_{p+1}^n - V_p^n) < \varepsilon \tag{13}$$

B. Power Summation

- The method as in the previous approach begins with a flat voltage start with values set at $1\angle 0^o$.

- In this approach the branch power is calculated instead of the branch current as in the previous approach and it is written as a sum of the load demand at the receiving node and the power drawn by the admittances connected to the same. This process is described in (14)

$$S^{k1} = S^{k} + \frac{Y^{K}}{(V^{k})^{2}} \tag{14}$$

- Backward sweep: Power summation is now carried out across all branches starting from the back end and moving towards branch 1 of the network. Taking example of the Fig. 5, if the total power in branch *i* is S^{it} and the total power in branch k after node k can is S^{kt} then the summation of currents across the two branches are described by the equation shown below.

$$S^{sum} = S^{it} + S^{kt} \tag{15}$$

  The branch total powers are calculated as a sum of the power shown in (14) plus the line losses due to impedance in the same branch. It is described in (16) for total power in branch *i*.

$$S^{it} = S^{k1} + Z_{s}^{i} . \left(\frac{S^{k}}{V^{k}}\right)^{2} \tag{16}$$

- Forward sweep: Now that the branch impedance values, and the sending node voltages are known. The receiving node voltages across all nodes in the network can be calculated as shown in equation (17).

$$V^{k} = V^{i} - Z_{s}^{i} . \left(\frac{S^{k}}{V^{i}}\right) \tag{17}$$

- Once the voltage values are determined throughout the network an iteration is complete and the entire is process is repeated again and the difference in voltage values between successive iterations is calculated. If the difference between the voltage values in successive iterations is less than a pre-determined small value, the process is stopped, and the values obtained are taken to be the final values. Equation (18) presents the stopping criteria where *p* is the number of iteration and n represents branch number and the maximum difference is calculated amongst the differences of the voltages between successive iterations which is then compared to the pre-determined small value ε.

$$Max\ (V_{p+1}^{n} - V_{p}^{n}) < \varepsilon \tag{18}$$

C. Admittance Summation

- The process is initiated by defining a driving point admittance (Y_{e}^{k}) in addition to the existing shunt admittance (Y^{k}). Apart from which an equivalent current generator (I_{e}^{i}) is also introduced in order to represent the current at the node. This current is a sum of all the load currents at the node.
- The above mention step is followed by a flat voltage start with values set at $1\angle 0^{o}$.
- Once all admittances are defined in the network including the driving point, branch and shunt admittances, a summation is carried out of the admittances from the last node to the first. The process, keeping Fig. 5 as a reference, is shown in (19) and (20)

$$D_{b}^{k} = \frac{1}{1+Z_{s}^{i} . Y^{K}} \tag{19}$$

$$Y_{sum} = Y^i + D_b^k \,.\, Y^k \tag{20}$$

- Backward sweep: The equivalent current generator (I_e^i) is calculated at all nodes after which current summation is carried out. This is shown in (21)

$$I_{sum} = I^i + D_b^k \,.\, I^k \tag{21}$$

- Forward sweep: The voltages at the receiving nodes can now be calculated with the available information on voltages at the sending nodes and the current generators.
- Once the voltage values are determined throughout the network an iteration is complete and the entire is process is repeated again and the difference in voltage values between successive iterations is calculated. If the difference between the voltage values in successive iterations is less than a pre-determined small value, the process is stopped, and the values obtained are taken to be the final values. Equation (22) presents the stopping criteria where *p* is the number of iteration and n represents branch number and the maximum difference is calculated amongst the differences of the voltages between successive iterations which is then compared to the pre-determined small value ε.

$$Max\ (V_{p+1}^n - V_p^n) < \varepsilon \tag{22}$$

Much of the studies and analysis carried out so far have been done so utilising MATPOWER. The same academic power analysis tool will also be used for carrying out radial load flow studies. While numerous methods have been described so far, the current summation method will be used to carry out analysis in the examples shown below. In order to successfully employ the radial power flow, oriented branch ordering is used. It imperative to utilise the nomenclature described in the ordering to carry out analysis. The ordering requires that all the branches of the system are oriented from the sending to the receiving bus and that no receiving bus can have an index less than that of a sending bus. It is clearly explained in [5].

In this study the generator at node 1 also representing the PCC of the radial microgrid will be chosen as the swing/slack/reference bus.

The impedances represented by Z of the lines between all nodes of the microgrid have parameters as shown in Table 3. They are aluminium cables with PVC outer sheath with XLPE insulation having 1 core with 120mm^2 cross-section.

Table 3: Cable data

From	To	Distance (m)	r + jx (Ω) (10^{-1})
Bus 1	Bus 2	155	0.392 + 0.127j
Bus 2	Bus 3	150	0.379 + 0.123j
Bus 2	Bus 4	210	0.531 + 0.172j
Bus 4	Bus 5	190	0.481 + 0.155j
Bus 5	Bus 6	260	0.658 + 0.212j
Bus 5	Bus 7	170	0.430 + 0.139j

Figure 6: Radial microgrid network

The above microgrid model consisting of numerous sources and loads was chosen to carry out power flow simulations. It consists of 9 generators, 7 of which are solar panels and 1 Battery Storage System. There is a total of 11 loads. The underlying data for all are taken from the real world such as from the PV system at Wroclaw University of Science and Technology and loads from a nearby village. Other loads and PV panels in the model were obtained by using different data manipulation techniques. The battery storage system was modelled according to the recommendations of [3]. While a simple model for the BSS is chosen at this moment, a far more updated model of the same is used later on. The algorithm of battery management in the microgrid is shown in Fig. 2 and is the same as used previously.

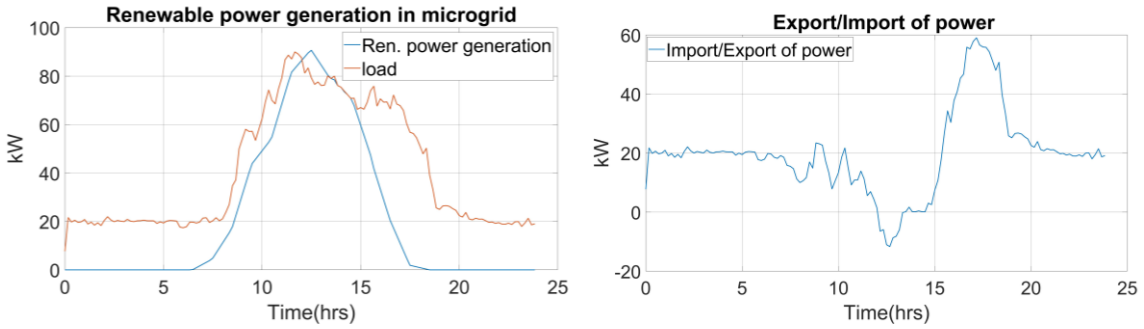

Figure 7: Total renewable power generation and load in the microgrid (left) and exchange of power with the main grid (right)

The load of the microgrid and the total renewable power generation of the same are shown in Fig. 7 (left) whereas Fig. 7 (right) shows the total amount of power imported through the PCC of the microgrid. During mid-day, the total renewable power produced exceeds the load demand of the microgrid just for a moment and correspondingly it is seen in Fig. 3 (right) that during the same moment power is exported from the microgrid through the PCC. During other times, the excess load is satisfied by importing power.

Fig. 8 depicts the storage device characteristic during operation for the day under consideration. It can be seen that the storage charges when there is more renewable power produced in the microgrid than the load and discharges when the load is greater than the renewable energy produced in the microgrid.

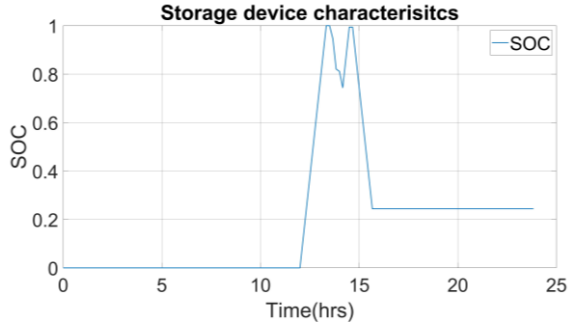

Figure 8: Storage device characteristics

3.2 Optimal Power flow

In the previous sections, power flow for both meshed and radial networks was introduced briefly with examples and in this section OPF will be introduced. Firstly, conventional methods of OPF will be introduced with a few examples followed by the meta-heuristic and hybrid algorithms to solve the OPF problem. A brief introduction to both approaches was provided in the introduction section.

3.2.1 OPF problem formulation

While the main task in power flow is to balance active and reactive power while maintaining operational and system technical parameters there is no optimization. While OPF was introduced over half a century ago for economic dispatch, it has evolved since then in numerous ways due to introduction of renewable energy sources, electricity markets, prosumer concept, changing objectives and others. The existing OPF problem is highly complex, nonlinear, nonconvex and is large scale problem which necessitates continuous improvement of solvers used to solve the OPF problem [7], [8]. It is important because with increased complexity comes increased time of computation to find an optimal solution and the quality of the solution itself which is further away from the true solution.

In general, it can be said that OPF attempts to achieve an objective set by the user such as operational cost minimisation by controlling power flow within an electrical network while at the same time maintaining different technical and operational constraints of the network. The similarity with power flow lies in the fact that OPF also solves for active power and reactive power injections at all nodes along with voltage magnitudes and angles in the same. The improvement over power flow is the fact that it recognizes the existence of numerous steady state conditions of the network where every above-mentioned value can exist within a certain defined range. [7], [8]

A review of numerous objectives functions used are as follows:

- Operational cost minimisation [9]–[12]
- Cost minimisation considering capital cost of RES [13], [14]
- Minimisation of losses (Active Power) [15]–[17]
- Minimisation of losses (Reactive Power) [8], [18]
- Cost minimisation considering VAR devices [19]
- Optimal voltage profile [20]–[22]
- Emissions minimisation [23], [24]
- Optimal rescheduling and shifting of controls [25]
- System loadability [25]
- Load shedding [15], [26]

Apart from the objective functions shown above, it is also quite regularly observed that combinations of objective functions are used in a multi objective manner. [27] presents a study in which a multi-objective function considering minimisation of operational costs and network losses are considered. Whereas, [28] presents a study where a multi-objective function considering operational costs and dynamic thermal rating of the line is considered.

For any optimization problem decision variables are chosen carefully. This is done so in-order to improve the performance of the optimization process in terms of speed so that premature solving for all variables can be avoided. The selection of such variables quite often depends on domain knowledge. The OPF problem is no different, it can be split into two groups of state and control variables [8], [29]. The control variables also known as independent variables are the set which is controlled by the optimization algorithm. These values are varied continuously while the state variables are solved once the values of the control variables are decided. The state variables are also known as dependent variables. Different OPF formulations utilise different control and state variables [8], [29], [30].

Quite often for the OPF problem the control variables are active power generation of all buses except the slack bus, voltage magnitudes of all generating buses, transformer tap setting and reactive power injections into the network [8], [29]. The other variables constitute state variables some of which are power outputs of the reference/slack bus, line loadings, voltages at the load buses etc.

The variables mentioned above usually are allowed to exist within a range and are not fixed unchangeable values. This is especially true in case of control variables which are varied by the optimization algorithms. The range of the variables is usually defined with the help of inequality constraints and a few examples are as follows:

- Active/reactive power limits of generators and VAR compensators
- Load demand constraints (controllable loads)
- Voltage magnitude limits
- Line current/power limits
- Reserve power limits
- Transient stability/voltage angle limits

The equality constraints are typically power balance equations for both active and reactive power in the system.

3.2.1 Mathematical model of the OPF problem

The objective function, equality constraints and inequality constraints make up the optimization problem. When applied to OPF with adequate knowledge of the types of objective functions, control variables, state variables, equality, and inequality constraints it becomes an effective tool for control of power flow in the system and it mathematically described as follows:

3.2.1.1 General optimization problem

$$Min\ f(u,x) \tag{23}$$

$$g(u,x) = 0 \tag{24}$$

$$h(u,x) \leq 0 \tag{25}$$

Where, $f(u,x)$ represents the goal of the problem to be minimised or in some cases maximised where it is simply written as negative of the minimisation. It is usually a continuous function but can be a vector too depending upon whether there is a single objective or there are multiple objectives. $g(u,x)$ and $h(u,x)$ represent the problem's equality and inequality constraints which cannot be in violation while obtaining the final optimized solution. (23), (24) and (25) together describe the complexity of the problem and give information about the type of optimization problem that has to be solved whether it is a linear programming, non-linear programming, mixed integer non-linear programming etc.

3.2.1.2 Objective functions

The objective function in OPF problems can be of many types as described earlier. A few of them are described here.

- **Quadratic cost (operational cost)**

$$Min(F_{cost}) = \sum_{i=1}^{n} \alpha_i P_{gi}^2 + \beta_i P_{gi} + \gamma_i \quad (26)$$

Where, i represents the number of the generator. P_g represents the real output power of the generator and $\alpha_i, \beta_i, \gamma_i$ represent the cost coefficients of the generators. F_{cost} representing the total cost of producing power across the network is the function to be minimised.

- **Linear cost (operational cost)**

$$Min(F_{cost}) = \sum_{i=1}^{n} \beta_i P_{gi} + \gamma_i \quad (27)$$

Certain approaches simplify the cost function making it linear as shown in (27), here i represents the number of the generator. P_g represents the real output power of the generator and β_i, γ_i represent the cost coefficients of the generators. F_{cost} representing the total cost of producing power across the network is the function to be minimised.

- **Active power loss**

$$Min\ (P_l) = \sum_{i=1}^{nl} g_k\ (V_i^2 + V_j^2 - 2V_i V_j cos\delta_{ij}) \quad (28)$$

Where, P_l is the power loss of the system to be minimised, g_k is the conductance of the k^{th}branch, nl is the number of lines in the system, V_i and V_j are the voltage magnitudes at bus/nodes i and j and δ_{ij} represents the voltage phase angle difference between buses i and j.

- **Voltage profile improvement**

$$f = \sum_{i=1}^{N_{PQ}} |V_M(i) - 1| \quad (29)$$

Where, V_M is the voltage magnitude at every node of the power system and one represents the fact that the voltage magnitude in p.u. should be maintained ideally at 1. N_{PQ} is the number of load/PQ buses in the network.

- **Emissions**

$$Min(E) = \sum_{i=1}^{N_g} \alpha_i + \beta_i P_{gi} + \gamma_i P_{gi}^2 + \delta_i \exp(\epsilon_i P_{gi}) \quad (30)$$

Where, N_g is the number of generators, $\alpha_i, \beta_i, \gamma_i, \delta_i\ and\ \epsilon_i$ are the emission coefficients for the generating unit i. P_{gi} is the power generated by every unit i and E is the total emission of the system that is to be reduced.

3.2.1.2 Equality constraints

The equality constraints of the OPF problem are usually power balance equations for both active and reactive power. There are two major approaches to writing the power balance equations when it comes to AC OPF. This includes writing the equations either in polar form or rectangular form.

- **Polar form**

$$P_{gi} - P_{di} - V_i \sum_{j=1}^{n} V_j [G_{ij} \cos(\delta_i - \delta_j) + B_{ij} \sin(\delta_i - \delta_j) = 0 \quad (31)$$

$$Q_{gi} - Q_{di} - V_i \sum_{j=1}^{n} V_j [G_{ij} \sin(\delta_i - \delta_j) + B_{ij} \cos(\delta_i - \delta_j) = 0 \quad (32)$$

V_i and V_j represent voltage magnitudes at buses i and j respectively. P_{gi}, Q_{gi} represent active and reactive power generations at bus i, P_{di} and Q_{di} represent active and reactive powers at bus i, δ_i and δ_j represents voltage angles at buses i and j respectively, $G_{ij}\ and\ B_{ij}$ are the conductance and susceptance between buses *i* and *j*. *n* is the total number of buses.

- **Rectangular form**
  In the rectangular form, the voltages are represented in terms of their real and imaginary parts instead of the magnitude and angle form ($V\angle\delta$). This approach offers an advantage with regard to computation as trigonometric functions are eliminated from the constraints and instead constant second partial derivatives are used [8]. The equations are shown below:

$$P_{gi} - P_{di} - \sum_{j=1}^{N} G_{ij}\left(E_iE_j + F_iF_j\right) + B_{ij}\left(F_iE_j - E_iF_j\right) = 0 \quad (33)$$

$$Q_{gi} - Q_{di} - \sum_{j=1}^{N} G_{ij}\left(F_iE_j - E_iF_j\right) - B_{ij}\left(E_iE_j + F_iF_j\right) = 0 \quad (34)$$

  Where, P_{gi}, Q_{gi}, P_{di} and Q_{di} have the same definitions as earlier. E and F represent the real and imaginary components of the bus voltages at their respective buses. $G_{ij}\ and\ B_{ij}$ are the conductance and susceptance between buses *i* and *j* A few studies utilizing the rectangular form for OPF can be found in [31], [32].

3.2.1.3 Inequality constraints

The inequality constraints have been explained earlier. They can be defined as operational parameters which are constrained within a range. It can be due to regulations defined by regulatory authorities or can be due to the physical limits of numerous elements that exist in a power system. Mathematically, they are defined as shown below.

$$P_{gi}^{min} \leq P_{gi} \leq P_{gi}^{max}\ \forall\ i\ \epsilon\ \text{No. of generators} \quad (35)$$

$$Q_{gi}^{min} \leq Q_{gi} \leq Q_{gi}^{max}\ \forall\ i\ \epsilon\ \text{No. of generators} \quad (36)$$

$$V_i^{min} \leq V_i \leq V_i^{max}\ \forall\ i\ \epsilon\ \text{No. of buses} \quad (37)$$

$$\delta_i^{min} \leq \delta_i\ \leq \delta_i^{max}\ \forall\ i\ \epsilon\ \text{No. of buses} \quad (38)$$

$$Q_{Ci}^{min} \leq Q_{Ci} \leq Q_{Ci}^{max}\ \forall\ i\ \epsilon\ \text{No. of Var compensators} \quad (39)$$

$$S_{li}\ \leq S_{il}^{max}\ \forall\ i\ \epsilon\ \text{No. of lines} \quad (40)$$

Where, P_{gi} and Q_{gi} represent the active powers and reactive powers generated at the generating stations and are constrained within the generators' minimum and maximum capacity. V_i and δ_i represent the voltage magnitude and angles at all buses, they are constrained within their minimum and maximum values defined by regulations of power quality and system stability. Q_{ci} represents the reactive power compensation provided by VAR elements and they are also constrained within their maximum capacities. The line loading defined by S_i is constrained within its maximum rated power carrying capacity.

3.3 Hybrid meta-heuristic optimization approaches

While both conventional and meta-heuristic optimization approaches to OPF were explained in the introduction. It was done only briefly, and a more detailed analysis will be dealt with in

this section. Conventional methods of OPF such as interior point method, Newton's method etc will not be discussed in this section since their results are widely available but will be used for comparison. All approaches adopted in this study are described as hybrid in nature since OPF solutions are obtained by algorithms that are meta-heuristic but have MATPOWER embedded in their solution searching process. This particular academic suite was chosen since it has been long established as a reliable power system simulation tool, is open source and is developed regularly [1], [5].

3.4 Genetic – Algorithm (GA)

GA was briefly explained in the introduction section but is explained in a more detailed manner in this section. The algorithm was developed in 1975 and then popularized by Goldberg in the late 80's. GA was developed in parts with numerous features being added over time which improved both its solution searching capabilities and speed [33]. As explained earlier the algorithm is based on the Darwinian principal of the survival of the fittest. Where, individuals with a higher level of fitness due to a variety of reasons are able to survive when compared to the weaker individuals when faced with adversity. These traits that result in a higher fitness are then passed on to future generations making them fitter. During this evolutionary process random mutations also occur which could possibly increase the fitness of an individual that is also passed on to the next generation. In terms of finding a solution to an optimization problem, each individual also called a chromosome represents a potential solution. The chromosome consists of numerous genes which can be equated to the decision variables of the optimization problem. The values of these decision variables ultimately determine the fitness of the individual which can be equated to the quality of the solution. Higher the fitness, better is the solution. Every generation has a population of a fixed size and it consists of individuals from the previous generation with the highest levels of fitness and their children. A new generation is usually created in this manner using operations called selection, cross-over and mutation. The above-mentioned process continues and new generations are created until the improvement from one generation to the next is negligible or the max. time for running the algorithm is reached.

Selection is the process of choosing individuals in a given population at a certain generation for producing children. They are called parents and there are different methods of selecting the right parents. Commonly used methods are the proportionate selection method, the ranking method and tournament selection method.

In the proportionate selection method parents are chosen based on probabilities. The probability of each individual being selected is directly proportional to the fitness of the individual. Higher the fitness, higher the probability of being selected as a parent. Once probabilities of all individuals are decided they receive a place in a roulette, the area of which is proportional to their probabilities. The roulette is then spun as many times as decided by the user which equals the number of individuals being selected as parents [33]. An example roulette is shown in Fig. 9 consisting of 5 individuals. Amongst all candidates, candidate 1 has the highest fitness hence largest portion of the roulette which gives it the highest probability to be selected when the roulette is spun. Conversely, the 5^{th} candidate has the lowest fitness hence the smallest portion of the pie and therefore the least probability of being selected.

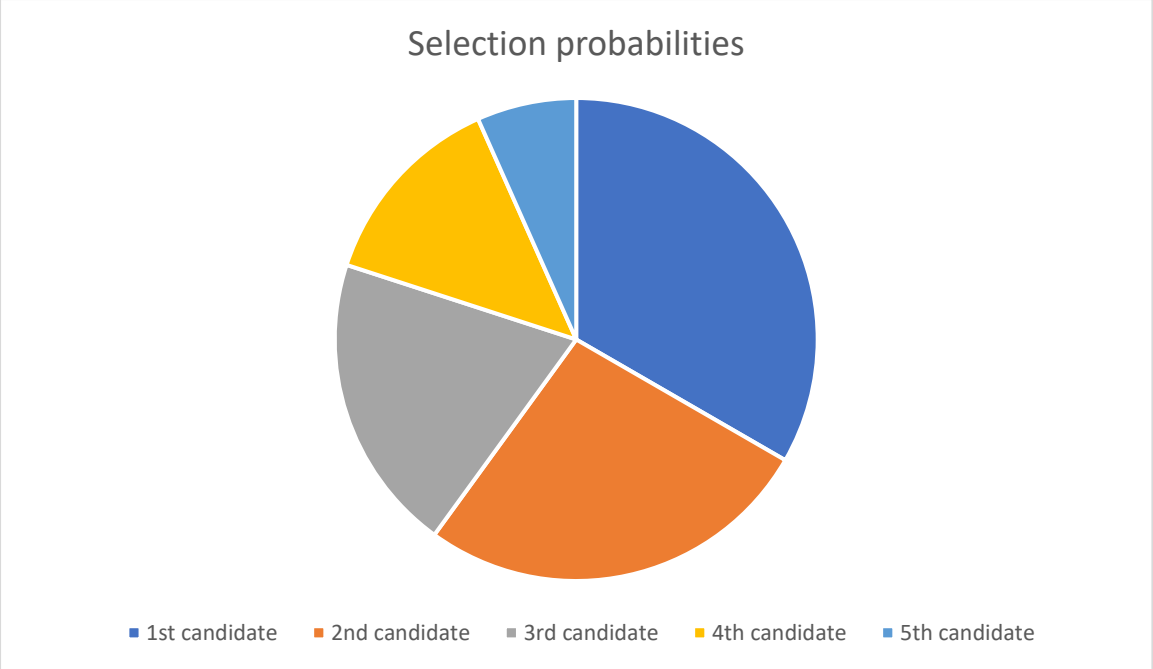

Figure 9: Underlying probability for proportionate selection method

The ranking solution method is quite straightforward and easy to implement. The candidate solutions are ranked according to their fitness values with the best solution getting rank 1 and the worst solution getting the last rank. Once the ranks are determined the candidate solutions' probabilities are determined with the help of a linear function directly proportionate to their ranks.

The tournament selection method involves randomly selecting a few candidates from the pool of all possible candidates then the solution with the highest fitness is selected. The number of randomly selected candidates is called as the selection pressure. Higher the number, higher is the selection pressure.

Apart from the classical selection approaches mentioned above other approaches exist, such as stochastic universal sampling, truncation selection etc.

Crossover can be described as a process of obtaining new chromosomes in a future generation from two chromosomes in the previous generation. It involves the exchange of genes between two chromosomes of the previous generation (parents). In short, the decision variables between two possible solutions are exchanged. There are different approaches to crossover such as one-point, two point and uniform crossover. These approaches are shown in Fig. 10.

In the one-point crossover approach a point is randomly selected on the chromosome representing the parents and then a part of the child is taken from one side of the point to the end and the other part of the child is taken from the other side of the point to the end. It is represented by the first crossover operation in Fig. 10. In the two-point crossover, two points are selected at random on the chromosomes representing both parents and the child is obtained by swapping out the genes apart from the ones located within the boundary of both the points. This is represented by the second crossover operation in the figure[33]. In case of the uniform crossover, points are selected at random where the genes are exchanges amongst the parents to create children. This is represented by the 3rd crossover operation in the figure.

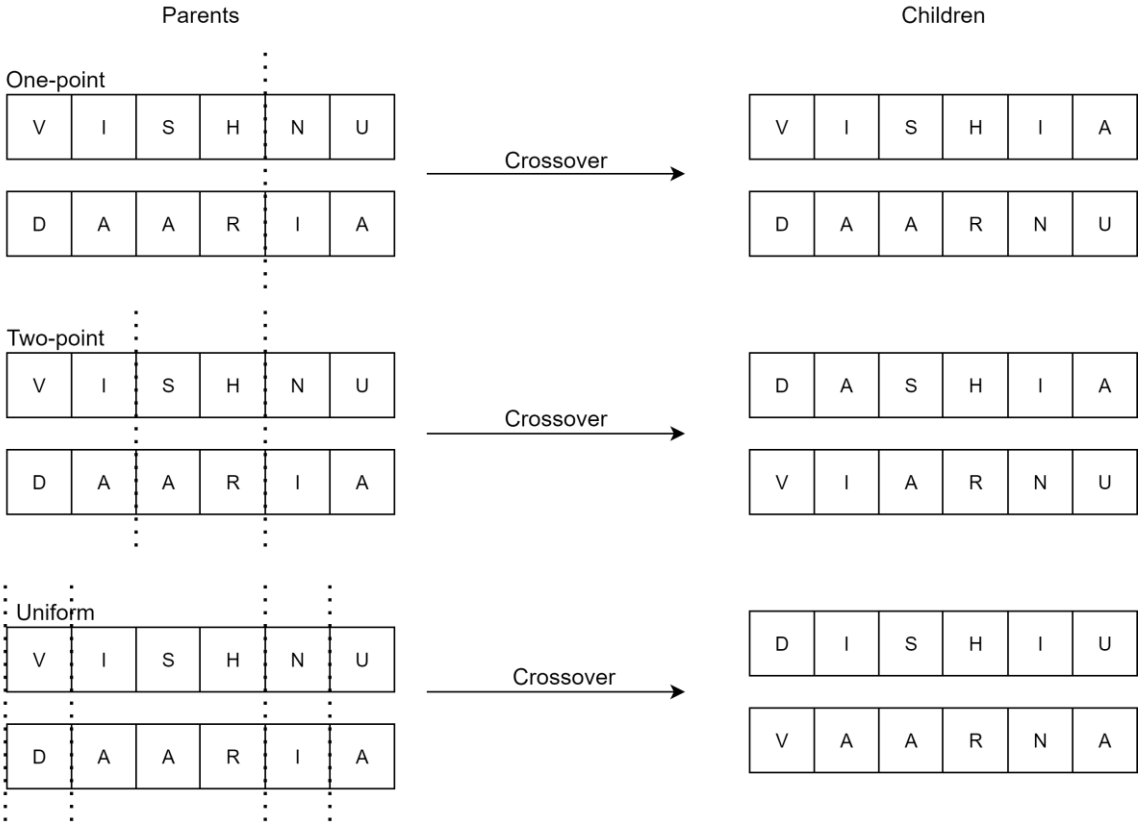

Figure 10: Different crossover approaches

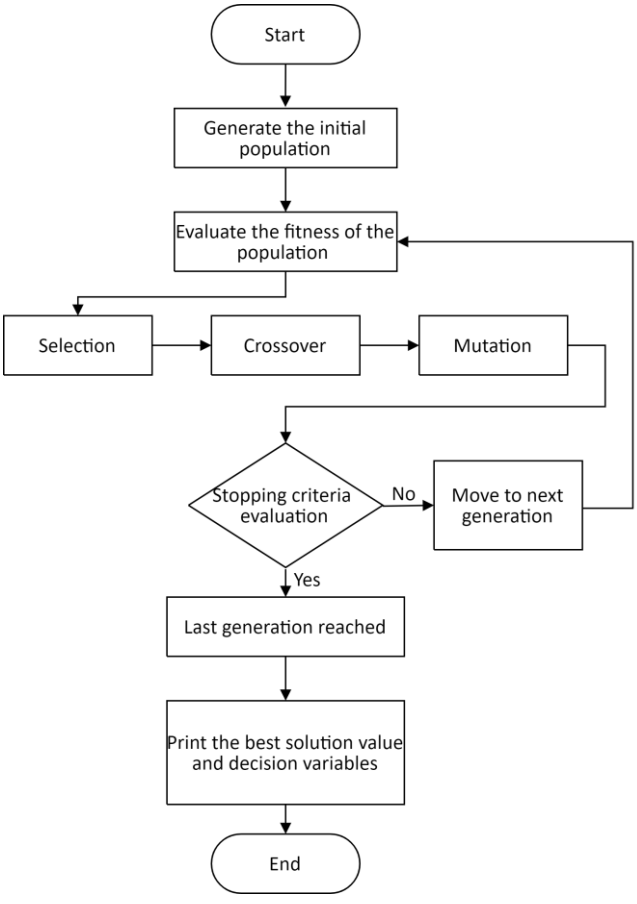

Figure 11: Genetic algorithm

The 3rd operation while creating a new generation is the mutation. As already explained before it is a random act (introduction of a random new genetic material) which might improve the fitness of the new solution obtained. In this case it involves swapping out a decision variable from the solution for a random unknown value. There are two approaches for mutation called uniform and non-uniform mutation. In the uniform approach a decision variable is replaced with a value from the feasible space of solutions whereas in the non-uniform approach a localised value is chosen to replace the gene and this local search space for a random value keeps reducing at the number of generations increase[33]. The process described above can be visualised in terms of an algorithm as shown in Fig.11.

For application to the OPF problem, GA is embedded with MATPOWER and the algorithm of the search solution process for OPF is shown in Fig. 12. The process is initiated with the declaration of all variables describing the characteristics of all elements in a power network such data pertaining to generators, loads, transmission lines etc. This is followed by declaration of the objective function, the constraints and the control variables. The control variables used in this study will be discussed in the results section of this chapter while evaluating the performance of this algorithm in the IEEE 30 bus system. Once the control variables are initialized, the state variables are solved for using MATPOWER after which the value of the objective function is computed. In the next iteration these control and state variables are updated once again and a new objective function value is calculated. This iterative process keeps repeating until an improvement in the final objective function value is negligible or until a max time for running the program is met.

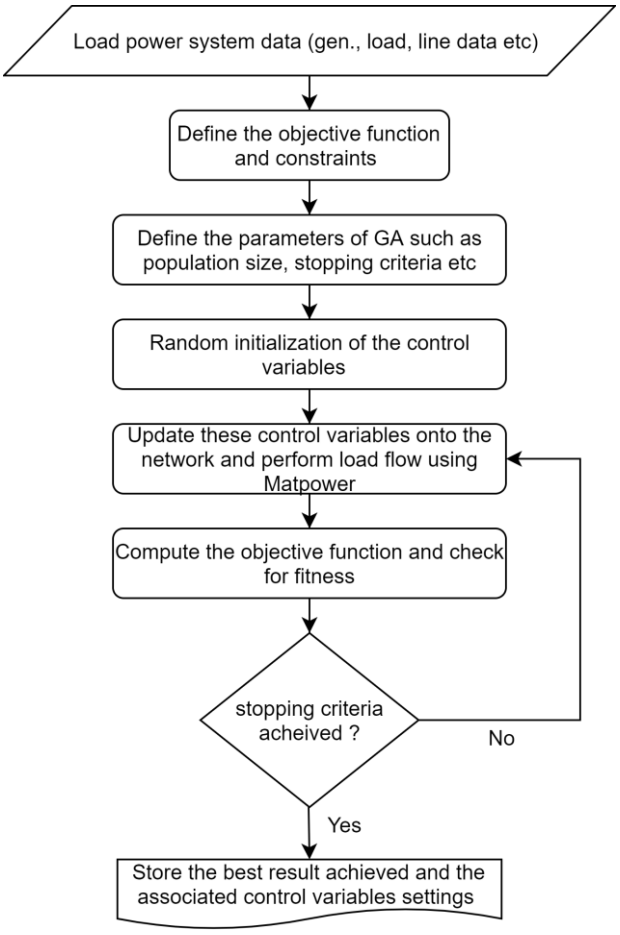

Figure 12: Genetic algorithm embedded with MATPPOWER for OPF

3.5 Particle Swarm Optimization (PSO)

Swarm intelligence is a concept modelling the behaviour of animals in large groups. It models their way of living and interaction with one another. It characterizes for example the manner in which fishes interact in a school of fishes or how birds behave in a flock of birds or even the behaviour of humans in large groups [34]. The advantages of using PSO are that, it can be implemented easily and also can be applied to numerous engineering problems with relative ease. It is one of the most powerful meta-heuristic optimization algorithms when it comes to speed of convergence to a solution [34]. The reason for this is the manner in which the solution searching process is carried out. Unlike GA which has operators for evolution such as mutation and cross-over, the PSO adopts a rather simple approach of simply moving particles (potential solutions) across a search space based on the location of the previously known best solutions. This approach significantly reduces memory requirements for running the optimization algorithm and increases the speed of convergence but the trade-off comes in the form of reduced accuracy and increased unreliability [34].

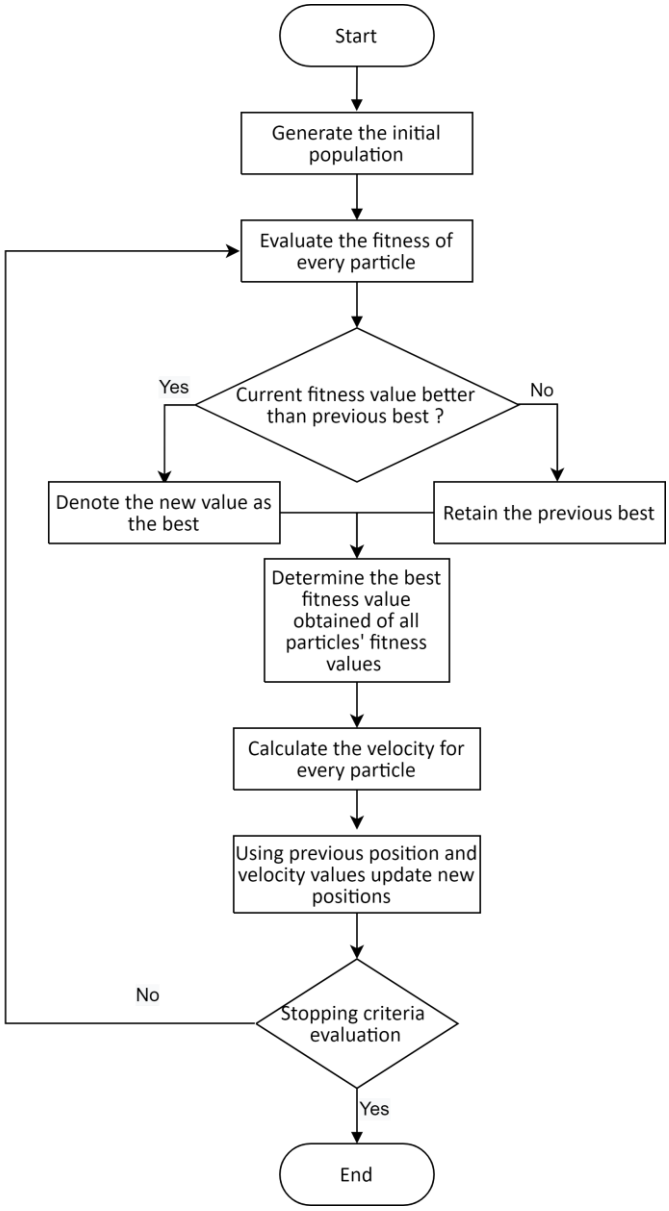

Figure 13: Particle Swarm optimisation

While there are numerous variants and upgrades made to the PSO [35]. The classical algorithm works in the following manner. The process is initiated by deploying a swarm of particles representing potential solutions randomly in a solution search space. The particles then move around the specified search space looking for new improved solutions. The movement of particles is determined by their position and velocity. It should be mentioned that the movement is not only influenced by the previously known best position of the particular particle but also the best positions found by all the particles within the search space. In this manner all particles move towards the best global solution [35]. Mathematically, the process is described by the equations shown below.

$$V = c1 \times rand \times (pbest - X_{present}) + c2 \times rand \times (gbest - X_{present}) \quad (41)$$

$$X\_new = V + X_{present} \quad (42)$$

Where, the velocity of each particle is V, the learning rates are $c1$ and $c2$ and is usually a number between 1 and 4, $rand$ represents a random number, the best position of a particle is represented by $pbest$, $gbest$ represents the best fitness of all particles in an iteration, $X_{present}$ is the current position of a particle and X_{new} is the new calculated position of a particle. Algorithmically, the process is described in Fig. 13.

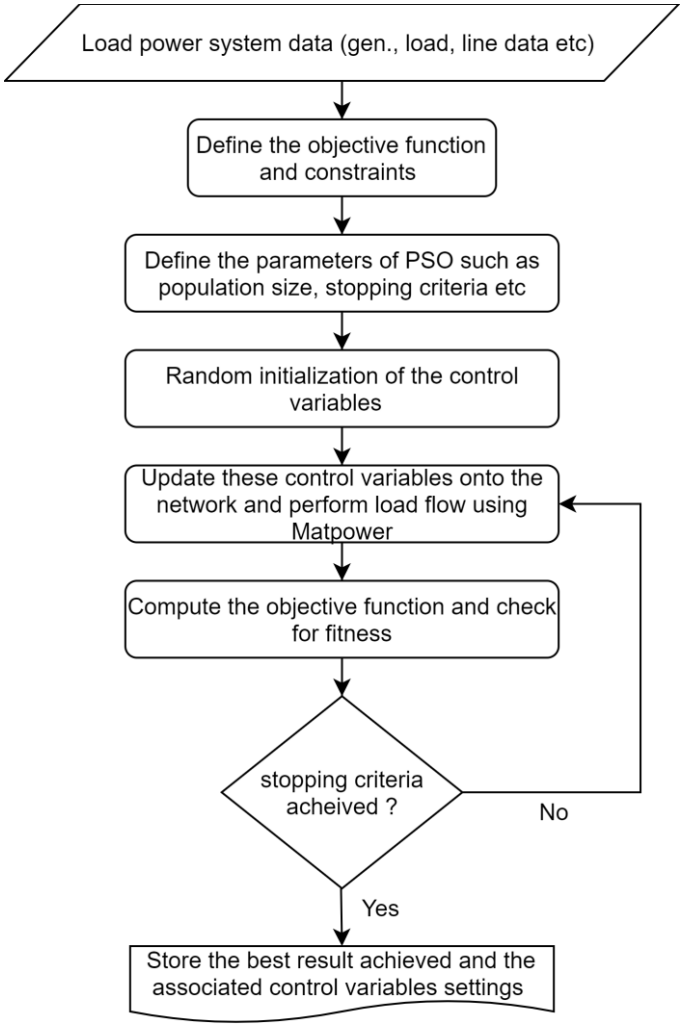

Figure 14: PSO embedded with MATPPOWER for OPF

For application to the OPF problem, PSO is embedded with MATPOWER and the algorithm of the search solution process for OPF is shown in Fig. 14. The process is initiated with the

declaration of all variables describing the characteristics of all elements in a power network such data pertaining to generators, loads, transmission lines etc. This is followed by declaration of the objective function, the constraints and the control variables. Once the control variables are initialized, the state variables are solved for using MATPOWER after which the value of the objective function is computed. In the next iteration these control and state variables are updated once again and a new objective function value is calculated. This iterative process keeps repeating until an improvement in the final objective function value is negligible or until a max time for running the program is met.

3.6 Mixed Integer Distributed Ant Colony Algorithm (MIDACO)

Ant colony optimization (ACO) was introduced initially by Marco Dorigo [36]. It is a robust and highly researched meta-heuristic optimization algorithm based on the foraging behaviour of ants. In nature, initially the ants move randomly about their colony searching for food. Once a food source is located, food is brought back and while doing so a pheromone trail down for identification of the food source by all ants. This pheromone trail evaporates over time. The food source that is located closest to the colony is associated with the strongest pheromone trail since it is frequented by the greatest number of ants all of which lay down a pheromone trail to the same source. In this manner all the ants converge to a single food source. The food sources can be equated to possible solutions and the associated pheromone trails determine the fitness of the solution. Hence, stronger the pheromone trail better is the fitness of the associated solution.

Traditional implementation of the ACO is done through a construction graph $G_c(V, E)$ where, V and E are the vertices and the edges of the graph respectively. V represents the food sources (solutions) and E represents the pheromone trails (fitness). Artificial ants move across the graph from one vertex to another and update the pheromone values. This way they create an optimal path to the best solution [36], [37]. During this process information about the vertices (solutions) and their associated pheromones (fitness) is shared with the other ants so that eventual convergence to a food source can be achieved. While the vertices and pheromones ensure search for a global optimal solution, the optimization problem under consideration might contain certain local constraints to be fulfilled which is achieved with the help of daemon actions. The daemon actions appropriately adjust the pheromone values (fitness) associated with a food source based on whether local constraints of the optimization problem are satisfied or not. The stopping criteria for ACO can either be the maximum run time or amount of improvement seen over subsequent best solutions. This process is described in Fig. 15.

MIDACO represents an extension over the traditional ACO because unlike the traditional ACO, MIDACO is able to handle both integer and continuous variables [37]. Moreover, it builds incremental solutions using probabilistic choices based on Probability Distribution Functions (PDFs). A PDF is defined as follows:

$$\int_{-\infty}^{\infty} P(x) = 1 \tag{43}$$

MIDACO uses a weighted sum of individual PDFs since a single PDF is centred around one mean only which implies that it cannot represent different points in the solution search space. This approach of using numerous PDFs in a weighted manner to search for a solution can be represented as shown in the equation (44).

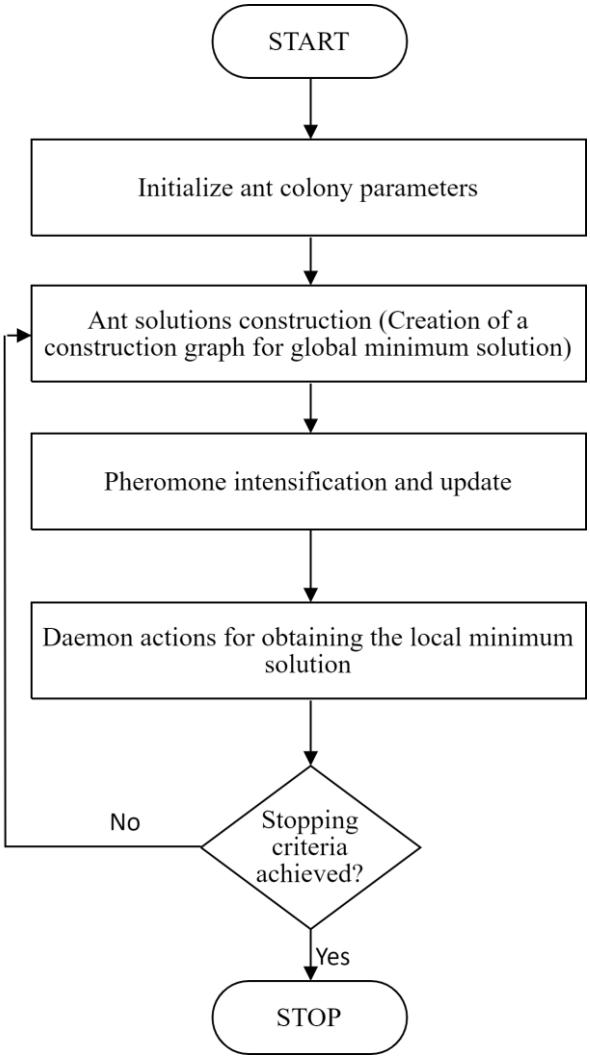

Figure 15: Traditional Ant Colony Optimization

$$G^i(x) = \sum_{m=1}^{k} w_m^i . g_m^i(x) = \sum_{m=1}^{k} w_m^i \frac{1}{\sigma_m^i \sqrt{2\pi}} e^{-\left(x-\mu_l^i\right)^2 / 2\sigma_l^{i2}} \quad (44)$$

Where, $G^i(x)$ is the weighted sum of individual PDFs ($g^i(x)$) and w_i, σ_i, μ_i represent weights, standard deviations, and mean of the individual Gaussian functions of the weighted cumulative representation in i dimensions of the search domain. l is the kernel of the individual function in the PDF.

The parameters described above characterizing a PDF can be equated to the pheromones of the ACO. They are responsible for guiding the ants all across the search space. The best solutions that are arrived at are usually stored in a Solution Archive (SA). It has k kernels specified by the user hence, if k has a value of 50 then the 50 best solutions obtained in an iteration are stored. Along with the best solutions in the kernel, the SA also has information regarding penalties that should be added to the objective function in case of violations such as inequality constraints violations. This ensures that only those objective functions having the best solution and following all constraints remain in the SA while others are discarded.

Mathematically, these weights representing pheromones are calculated as shown in (45), the weights are stored in the SA and a priority is established amongst all the possible weight values ascertained by the algorithm in its solution searching process.

$$w_m^i = \frac{(k-m+1)}{\sum_{j=1}^{k} j} \tag{45}$$

In (45), w_m^i represents one of the weights used. Prioritization of other weights follows the same approach. k represents the kernel size hence, the number of solutions ranked from best to worst and are stored in the SA. m is the index according the rank of each solution hence the solutions with a small m value are preferred by the algorithm. j is simply used to denote the solution under consideration out of all solutions available. As an example, the best solution is s_1 with index 1 whereas the worst solution is s_k with the last index k. While moving from iteration-to-iteration new solutions are explored and the SA is updated. For every new solution, a comparison is made between its penalty function value and the penalty function values that exist in the SA from the previous iteration. If there is a solution in the SA at the j^{th} position whose penalty function value is much higher than that of the current solution, the current solution replaces the j^{th} solution and the j^{th} solution is moved up by one index and so are all the other solutions moved up by one position until the k^{th} solution which is simply kicked out.

The way a new solution is explored in the search space is as follows. Based on (45) a mean μ_i is calculated for all decision variables. Then a mean μ_i corresponding to a particular decision variable is selected at random. During this selection the probability that μ_1, corresponding to the mean of the decision variable of the best solution in the SA, is selected is the highest whereas μ_k corresponding to the mean of the decision variable of worst solution in the SA is the lowest. Once a mean is selected randomly, utilizing a standard deviation σ calculated using the SA a new part of the search space from which a new solution can be obtained is created. The process of determining the standard deviation is shown below.

$$\sigma_m^i = \frac{dist\max(i) - dist\min(i)}{number\ of\ generations} \tag{46}$$

$$dist\max(i) = \{\left|s_g^i - s_h^i\right| : g, h \in \{1..,,k\}, g \neq h \tag{47}$$

$$dist\min(i) = \{\left|s_g^i - s_h^i\right| : g, h \in \{1..,,k\}, g \neq h \tag{48}$$

In this manner, new spaces in the search space are explored, then best possible solutions are chosen and over numerous iterations these solutions are improved until a stopping criterion of the maximum number of iterations is reached. The process is also stopped if a solution obtained cannot be improved more than a pre-determined negligible value.

A robust penalty method called Oracle is utilised for handling constraints in MIDACO [37], [38]. With regard to penalty methods in optimization algorithms certain approaches such as death or static are easier to implement whereas others such as annealing and adaptive are more difficult to implement but it should be mentioned that the latter has a significantly higher performance than the former. The latter is also associated with a higher computation burden. Hence, MIDACO was embedded with a penalty approach that has both a high performance and less computational burden in comparison with other penalty approaches mentioned. Oracle utilises a single parameter and is unaffected by the choice of the initial solution. A more comprehensive analysis of the penalty function is described in [38]. Further explanation about the penalty method and MIDACO can be obtained from [37], [38].

For application to the OPF problem, MIDACO is embedded with MATPOWER and the algorithm of the search solution process for OPF is shown in Fig. 16. The process is initiated with the declaration of all variables describing the characteristics of all elements in a power network such data pertaining to generators, loads, transmission lines etc. This is followed by

declaration of the objective function, the constraints and the control variables. Once the control variables are initialized, the state variables are solved for using MATPOWER after which the value of the objective function is computed. In the next iteration these control and state variables are updated once again and a new objective function value is calculated. This iterative process keeps repeating until an improvement in the final objective function value is negligible or until a max time for running the program is met.

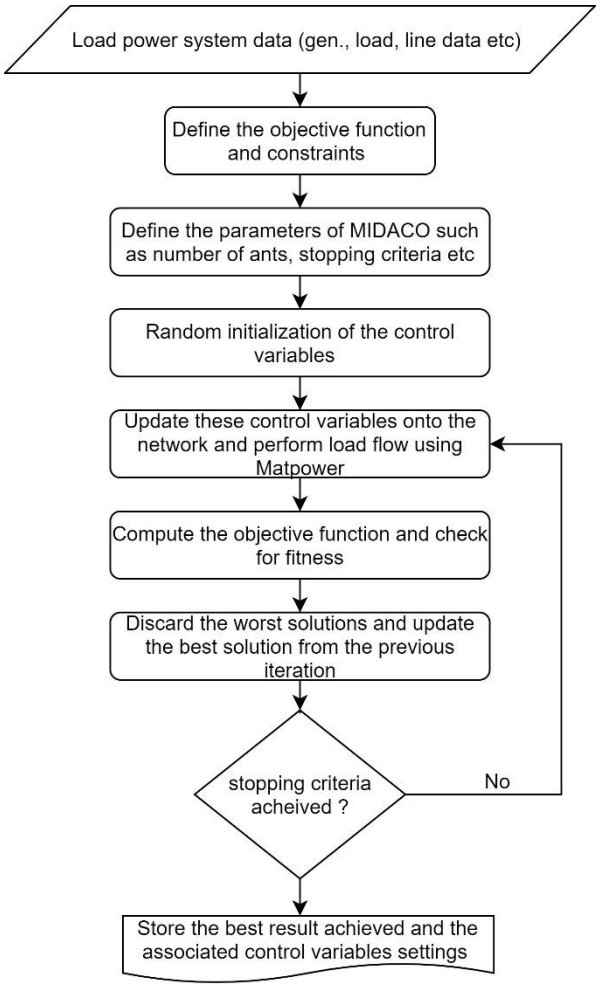

Figure 15: MIDACO embedded with MATPPOWER for OPF

3.7 Political Optimizer (PO)

While the above-mentioned approaches are quite popular, they can be classified as traditional in the world of meta-heuristic approaches to optimization. GA, PSO and MIDACO though improved in many ways were still developed over a long time ago. In recent times a gamut of new meta-heuristic optimization approaches is being proposed. This section discusses one such approach called the Political Optimizer developed in 2020.

PO is an approach that mimics the working of various political institutions and actors. It considers numerous procedural activities carried out in order to form and run a government such as election campaigns, intra-party and inter-party elections, defection to opposition parties, constituency elections and parliamentary affairs [39]. It comes under the category of meta-heuristic algorithms that are inspired by human behaviour and interaction.

The idea behind PO is thought to be appropriate for a global optimization algorithm because it incorporates two perspectives. One representing the candidates (potential solutions) who try to win the election (objective function) and another where political parties try to gain a majority in the parliament.

A comprehensive description to locating the global minimum solution is as follows, initially a population of candidates are generated. The population size p is decided by the user. This population is then further divided into n political parties consisting of n members. It has to be mentioned that every candidate representing a possible solution is a vector consisting of as many elements as the decision variables in the problem. An assumption is made at this point with regard to the number of constituencies and for PO it is always n, equal to the number of political parties. The purpose of a constituency is to select the best candidate of all available candidates with the same index in a political party. An example is shown in Fig. 16.

	C1	C2	C3
P1	P11	P12	P13
P2	P21	P22	P23
P3	P31	P32	P33

Figure 16: Constituency winner allocation

Assuming a population of 9 solution candidates. 3 parties $P1$, $P2$ and $P3$ each consisting of 3 members is shown in Fig. 16. Firstly, an assumption is made with regard to the number of constituencies which is the same number as that of the number of political parties, in this case, 3. Then for constituency C1, the best solution candidate is chosen from amongst the first member of each party in this case from either P11, P21 and P31. Similarly, the best candidates are chosen for C2 and C3. At this point the party leader is also chosen who is the best candidate in each party. The best candidate for each party and constituency is chosen based on its fitness (objective function value). The constituency winners together form the parliamentarians. This concludes the initialization of the PO following which are 4 phases that improve all the candidate solutions. They are described in brief as follows.

The exploratory and exploitation phase of the algorithm is called as the election campaign. It involves improving the solution candidates position in the search space so that they move towards the global optimal solution. Every candidate has a new proposed position based on 3 considerations which are their recent past position in the previous election, the position of the constituency winner and the position of the party leader [39].

In parallel to this phase is another phase called the party switching phase. The user of PO is allowed to change the parameter regulating this phase freely. The parameter fades towards zero as the number of iterations of the algorithm keeps increasing. The process involves switching a member of a political party to another party where this member replaces the member of the lowest fitness in the new party. The probability of choosing a member of a political party for switching depends on the value of the probability parameter at that iteration [39].

The election phase is run again, wherein new fitness functions of all candidates are calculated and again party leaders and constituency winners are determined.

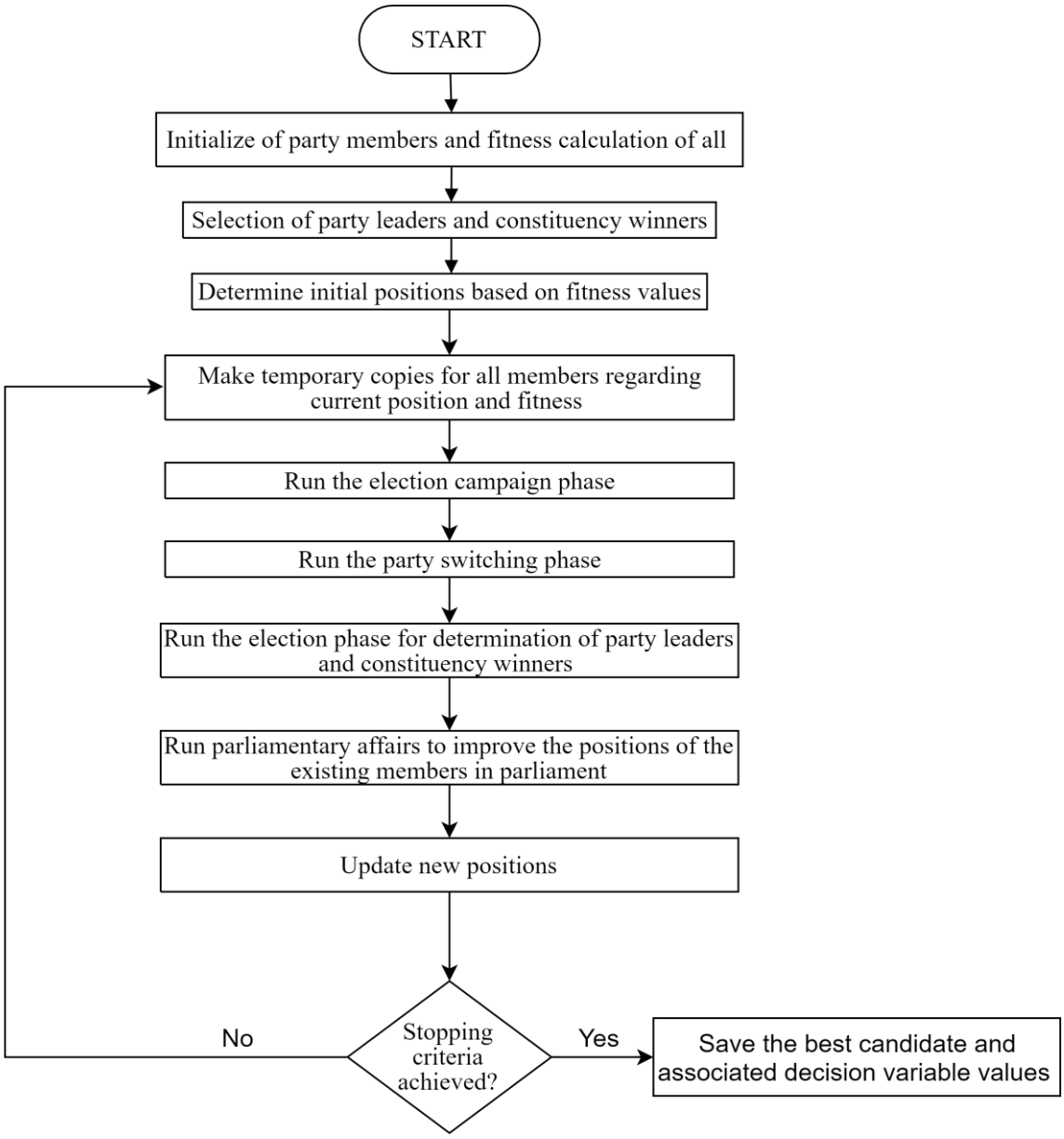

Figure 17: Political optimizer

Once the government is formed, in every iteration the next phase is called parliamentary affairs. In this phase every member of the parliament updates his/her position based on a randomly chosen parliamentarian. If there is an improvement in fitness the new position is recorded. The flow from one phase to another and selection of the best solution is shown in Fig. 17. The integration of PO to MATPOWER to solve OPF is similar to what has been done with the 3 traditional algorithms before with the exception being that the PO is used to solve for control variables instead of the traditional meta-heuristics.

3.8 Lichtenberg Algorithm (LA)

This algorithm is inspired by Lichtenberg Figures (LF). Based on the natural phenomena of lightning formation amongst clouds, it exploits fractal power and is significantly different from numerous other meta-heuristic optimization algorithms found in the literature [40]. This

approach was introduced in 2020 and represents another recent introduction to already existing pool of modern meta-heuristic algorithms.

The natural phenomena the algorithm is based on in particular is intra-cloud lightning formation. Lightnings arise due to electric discharge from electrical fields that are created due to the friction that arises because of cloud movement with respect to one another. Different types of lightning include inter cloud, intra cloud, cloud to ground, ground to cloud and from cloud to atmosphere. Amongst all the possible options, the intra-cloud lighting discharge is the most suitable to model an optimization algorithm since it is radial in nature and scans a wide area. While the intra-cloud lighting is a natural phenomenon that produces an LF. It can be created in dielectric materials by observing the propagation of an electric discharge on the material. This was demonstrated by Lichtenberg in 1777 [40].

Recently, it was proposed that LFs can be created via cluster forming random growth processes. Two popular approaches in the literature include Dielectric Breakdown Model (DBM) [41] and the Diffusion Limited Aggregation (DLA) [42]. The LA used in this study utilises the DLA model. It is a numerical approach which results in a random growth structure that is used to study the breakdown of resistance in dielectric materials. The principal of solid particles is used as the basis for the growth of clusters. The algorithm for DLA has two inputs namely the number of particles within the cluster and its size. Once the algorithm is run the output is completely random and complex and can only be defined by fractal geometry. More recently, a stick coefficient was introduced to this algorithmic approach which determines the speed of creation of the cluster and its density.

Mathematically, a binary matrix with entries one and zero is created in a map like manner. At the centre of the matrix is a particle with index 1 and is considered to be the centre of matrix. The cluster or the LF is built incrementally by the spaces of the matrix that have the entry 1 and the empty spaces have a value of zero. It must be mentioned that every entry 1 represents a particle and the total number of particles for creating the LF is defined by the user at the beginning. The overall space within which the LF is to be created is also defined by the user at the beginning in the form of a creation radius R_c. This process continues until the defined number of particles have found their position in the cluster or the limits of construction of the LF has been reached [40].

The particles begin from the centre and move across the search space constrained by the creation radius. If they reach a reach a position that is greater in distance from the centre than R_c that particle is eliminated and another particle starts moving in a different direction in a random manner. It is important to remember that every LF generated this way in different runs can be completely different. This kind of random stochastic exploration of a given search space makes it a suitable candidate for optimization problems with non – convexity. Two Lichtenberg figures are shown in Fig. 18. Both figures are created with a stick parameter of 1 and 1×10^8 particles. It can be seen that even though both figures were created with the same parameter settings they are significantly different.

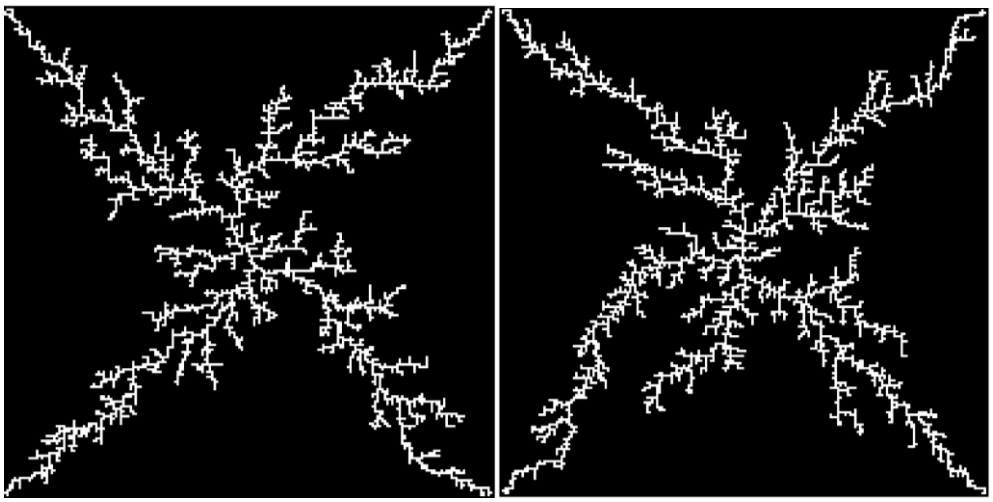

Figure 18: Lichtenberg optimizer

It can also be seen from the above figures that there is a significant difference between points and branches of the figure which could compromise the accuracy of optimization algorithms during solution searching. In order to combat this problem, the developers took 4 different measures as explained below.

The first measure was the introduction of a random value between 1 and 0 at every iteration. This randomly generated value is then multiplied with the LF's size. This is done to ensure that the distance between different points and branches does not remain large throughout the iterative process and more of the search space can be covered. It is allowed to vary between its max. allowed size to less than 1% of the maximum. It has shown to improve the efficiency of LA [40].

The second measure includes introducing a random angle at every iteration. Then with regard to this angle the entire LF is rotated. This process is especially useful for optimization problems with a high number of decision variables. It results in efficient exploration of the search space and ensures that repetition of branches in the same direction does not take place.

The third measure was the introduction of a refinement parameter which is a value between 0 and 1. This parameter creates another LF with a size equal to the original size of the LF multiplied with the refinement parameter. The main purpose of this new LF is to enable search for local optimization problems that need to be solved.

The fourth measure introduced resulted in another parameter called M in order to keep the search space continuous and not discretize it. This parameter takes on 3 possible values [0,1,2] where zero indicates that one figure is generated during the iterative process and it remains the same throughout, 1 indicates that a new figure is created at every iteration and 2 is a case where no figure is created but a previously saved figure from the archives is utilised.

At the beginning of the optimization process, the population and the number of particles is defined. The two values remain separate since not all particles are used for fitness evaluation but only the number of particles that are determined by the size of the population. Usually, the population size is 10 times the number od decision variables for this algorithm. Apart from this the value for the reference parameter, M parameter, R_c value and the number of iterations are defined. There is also a Sticky parameter S that determines the speed at which the LF is generated along with its density.

The LA during its search process denotes different parts of the search space with a number of points equal to the population size. While other particles do exist and randomly generate the LF, it is these particles for which the fitness function is evaluated and the point with the highest fitness serves as a trigger point for further exploration. The entire process is algorithmically is shown in Fig. 19.

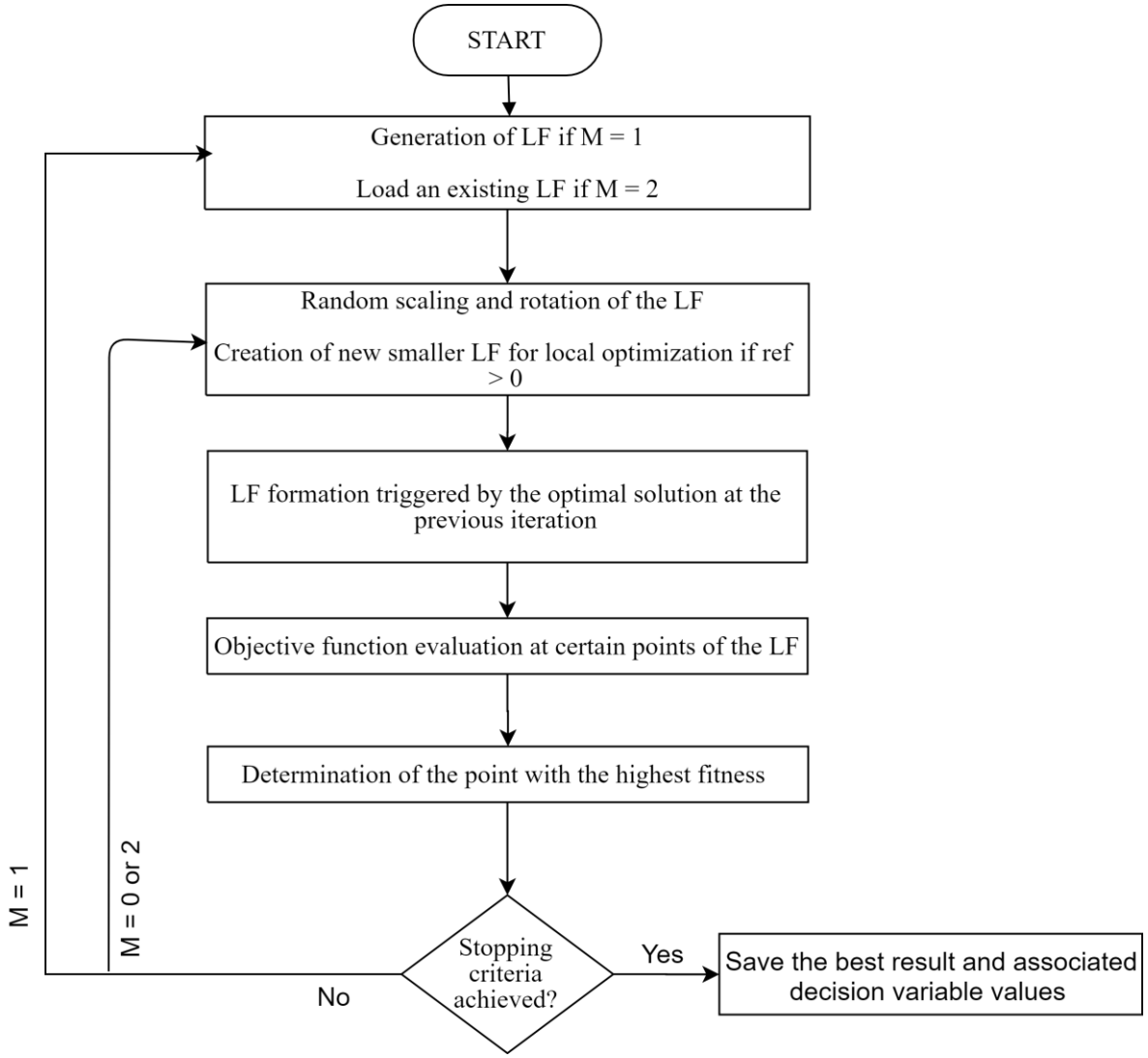

Figure 19: Lichtenberg optimizer

The integration of LA to MATPOWER to solve OPF is similar to what has been done with the 3 traditional algorithms before with the exception being that the PO is used to solve for control variables instead of the traditional meta-heuristics.

3.8 Performance evaluation of meta-heuristic approaches to OPF

3 traditional and 2 recent algorithms were explained in detail in the preceding sections. It was also explained how every approach was modified and incorporated with MATPOWER for solving OPF problems related to power networks. In all the literature explored in this chapter and the introduction chapter a valid and popular approach is to test any proposed optimization

algorithm in a standard IEEE 30 bus system. The objective function is minimisation of cost and the same approach is adopted in this work to compare all above-mentioned algorithms.

First the IEEE 30 bus system is explained. It is an open-source model representing an actual power system which is used to test the performance of optimization algorithms and carry out steady state analysis. As the name suggests it has in total 30 bus bars. It also has a total of 6 generators at bus numbers 1, 2, 5, 8, 11 and 13 respectively, 4 transformers whose tap settings are allowed to change within a specified range between buses 6 and 9, 6 and 10, 4 and 12 and 28 and 27 respectively. Shunt var compensation is provided at buses 10, 12, 15, 17, 20, 21, 23, 24 and 29 [12].

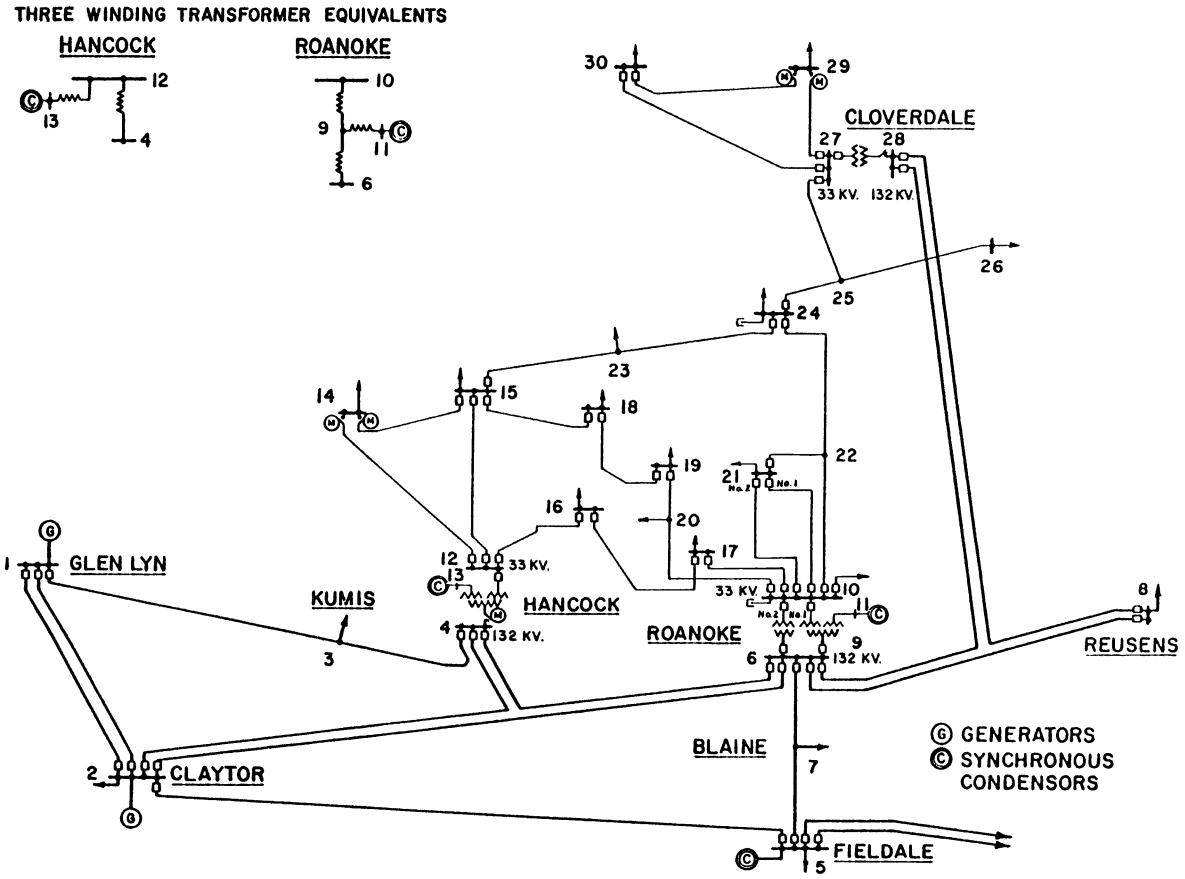

Figure 20: IEEE 30 bus system

Source: https://labs.ece.uw.edu/pstca/pf30/pg_tca30fig.htm (open source)

The control variables chosen for this work is similar to many in the literature already described. They are the active power output of the generators except slack bus, voltage magnitude values, transformer tap settings and reactive power compensation. These control variables form the decision variables of the optimization algorithms used. They are the variables that are allowed to vary in order to minimize the cost for the IEEE 30 bus system. Rest of the variables are solved with MATPOWER. It is also important to mention that while the generators maximum and minimum active and reactive power output limits along with VAR compensation limits are a part of the data that would be provided later.

Table 4: Control variables, cost and computation time comparison of meta-heuristics applied to OPF

Control variable values	GA	PSO	MIDACO	PO	LA
*P_{G1} (MW)	176.45	176.76	177.09	177.39	180.45
P_{G2} (MW)	48.75	49.36	48.83	48.84	47.59
P_{G5} (MW)	21.09	21.76	21.49	21.40	22.99
P_{G8} (MW)	23.20	25.73	21.73	21.69	18.30
P_{G11} (MW)	12.21	11.12	12.24	12.19	12.86
P_{G13} (MW)	10.95	13.81	11.31	11.20	10.98
V_1 (p.u.)	1.06	1.06	1.06	1.06	1.06
V_2 (p.u.)	1.04	1.04	1.05	1.04	1.05
V_5 (p.u.)	1.01	1.01	1.01	1.01	1.01
V_8 (p.u.)	1.01	1.01	1.01	1.01	1.01
V_{11} (p.u.)	1.08	1.08	1.08	1.08	1.08
V_{13} (p.u.)	1.07	1.07	1.07	1.07	1.07
T_{11}	0.94	1.03	0.94	1.01	0.95
T_{12}	1.07	0.96	1.05	0.93	1.01
T_{15}	0.97	0.96	0.95	0.94	1.09
T_{36}	0.94	0.95	0.93	0.93	1.00
Q_{c10} (MVAr)	3.26	4.77	4.98	4.61	0.23
Q_{c12} (MVAr)	4.30	4.10	4.99	5.00	3.39
Q_{c15} (MVAr)	4.00	0.13	4.99	4.32	2.97
Q_{c17} (MVAr)	4.85	0.04	4.86	4.92	3.55
Q_{c20} (MVAr)	4.58	3.23	4.96	4.56	4.56
Q_{c21} (MVAr)	4.65	4.26	4.97	5.00	2.56
Q_{c23} (MVAr)	3.27	0.28	3.19	2.71	0.81
Q_{c24} (MVAr)	1.39	4.27	3.41	3.65	3.75
Q_{c29} (MVAr)	3.06	1.37	2.10	2.72	3.89
Run time (s)	34.20	7.89	3.45	4.48	54.15
Cost ($/h)	801.6	801.7	801.5	801.6	802.8

*not a control variable but included in the table for reference since it is the power output of the biggest generator

The voltage limits are maintained between 0.95 – 1.05 p.u. and the transformer tap settings are also allowed to vary between 0.9 and 1.1. A single line diagram of the IEEE 30 bus system in shown in Fig. 20.

A detailed comparison of the control variables decided by different meta-heuristic approaches is presented in table 4. This comparison involves only the 5 meta-heuristic algorithms that were investigated in this study but in the literature, there are numerous other approaches and also a comparison with conventional algorithms has to be made. A table with conventional algorithms for OPF and some found in literature are presented in table 5 after which wholesome

conclusions are drawn regarding the performance of all the algorithms being discussed. A part of these results are published and a similar comparison made in [43]. The thesis adds to that by introducing more algorithms.

Table 5: Comparison of performance of different optimization algorithms for the ieee 30 bus system

Algorithm	Computational Time (seconds)	Generation Cost ($/hr)
MIDACO	3.45	801.5
NOPF*	< 1	805.45
PSO	7.89	801.70
GA	34.20	801.60
PO	4.48	801.60
LA	54.15	802.80
EDOPF*	< 1	813.74
MDE*	23.25	802.37
GSA*	10.75	798.67
MVPA*	8.97	799.12

Taken from the literature include conventional approaches such as the Newton OPF (NOPF) and Extended Dommel – Tinney OPF (EDOPF). Meta-heuristic approaches such as the Modified Differential Evolution (MDE) algorithm, Gravity Search Algorithm (GSA) and Most Valuable Player Algorithm (MVPA).

It can be seen that in the literature there are results that show a minimized cost lower than that of all the algorithms explored in this thesis but it is important to keep in mind that for ED problems of microgrids not only is the final solution important but also the computational time since optimisation will be made numerous times in every time step to prepare a schedule. Hence, approaches adopted in this thesis focus on finding a balance between the quality of the global minimum solution achieved and the run time.

It can be seen that the conventional algorithms have the least run time of all with less than a second to solve the problem but their final solution remains quite high, north of 805 ($/hr). In case of conventional algorithms, the GSA has the best solution at 798.67 ($/hr) but a high run time of 10.75 seconds. A similar performance is provided by the MVPA. Amongst the algorithms explored in this thesis the worst performance is by LA which has high run time and high cost whereas MIDACO has the best run time and the lowest cost. The others are in between.

PO and LA were the algorithms that were introduced in 2020 and it can be seen that while LA is not a suitable candidate for OPF, PO is very much suitable. Its performance is very close to that of MIDACO and it is important to keep in mind that PO is new and not developed to the extent of MIDACO which is a traditional approach to which numerous elements have been added over time. Hence, a development of PO could provide improved results when compared to all algorithm. Despite this fact this thesis will adopt MIDACO for ED analysis in the next chapter since it has the best performance of all algorithms explored in this thesis and it represents a good balance between run time and cost. It is higher in speed compared to GSA and MVPA and is cost-effective when compared to conventional optimization algorithms. The data of the IEEE 30 bus system needed to obtain the results are provided in the appendix to the book.

Conclusions

This chapter began with describing the power flow problem in order to obtain the steady state solution of the power system. The focus was mainly geared towards microgrids wherein two

different architectures were considered. One is the meshed network and other is the radial. The steady state solution for each architecture was obtained using different methodologies and numerical approaches. For the meshed network the GS, NR and FDLF methods were utilised whereas for the radial networks the current summation, power summation and admittance summation methods were used.

This was succeeded by definition of the OPF problem structure and exploration of numerous objective functions available and the constraints. Appropriate control variables of the problem was chosen based on literature. Once the problem was defined 5 meta-heuristic algorithms were used to solve the same 3 of which were traditional meta-heuristic approaches (GA, PSO and MIDACO) and 2 were novel optimization algorithms introduced in 2020. These approaches were used to optimize the IEEE – 30 bus system and a comparison was made with the subsequent results. It was found that with regard to both the quality of the final solution and the speed of finding the solution MIDACO presented the most balanced solution of all and it was selected for further solving the ED problems of modern microgrids described in next chapter.

References

[1] V. S. Suresh, "Comparison of Solvers Performance for Load Flow Analysis," *Trans. Environ. Electr. Eng.*, vol. 3, no. 1, p. 26, 2019, doi: 10.22149/teee.v3i1.131.

[2] S. Picco *et al.*, "We are IntechOpen , the world ' s leading publisher of Open Access books Built by scientists , for scientists TOP 1 %," *Intech*, no. tourism, p. 13, 2016, [Online]. Available: https://www.intechopen.com/books/advanced-biometric-technologies/liveness-detection-in-biometrics.

[3] Y. Levron, J. M. Guerrero, and Y. Beck, "Optimal power flow in microgrids with energy storage," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 3226–3234, 2013, doi: 10.1109/TPWRS.2013.2245925.

[4] V. Suresh and S. Rotondo, "Radial Power Flow in a Single Bus Microgrid with Dispatchable Load," no. i, pp. 1–6, 2019, doi: 10.1109/eeeic.2019.8783578.

[5] C. E. Murillo-Sánchez, R. D. Zimmerman, C. L. Anderson, and R. J. Thomas, "IEEE TRANSACTIONS ON SMART GRID 1 Secure Planning and Operations of Systems with Stochastic Sources, Energy Storage and Active Demand." Accessed: Dec. 19, 2018. [Online]. Available: http://www.pserc.cornell.edu/matpower/MOST-paper.pdf.

[6] R. A. Aguirre, "Improved Power Flow Program for Unbalanced Radial Distribution Systems Including Voltage Dependent Loads," vol. 4, no. 1, 2016.

[7] U. Zia *et al., A Mini-review: Conventional and Metaheuristic Optimization Methods for the Solution of Optimal Power Flow (OPF) Problem*. Springer International Publishing, 2020.

[8] S. Frank, I. Steponavice, and S. Rebennack, "Optimal power flow: A bibliographic survey I Formulations and deterministic methods," *Energy Syst.*, vol. 3, no. 3, pp. 221–258, 2012, doi: 10.1007/s12667-012-0056-y.

[9] B. V. Rao and G. V. N. Kumar, "Electrical Power and Energy Systems Optimal power flow by BAT search algorithm for generation reallocation with unified power flow controller," *Int. J. Electr. Power Energy Syst.*, vol. 68, pp. 81–88, 2015, doi: 10.1016/j.ijepes.2014.12.057.

[10] K. Vaisakh and L. R. Srinivas, "Evolving ant direction differential evolution for OPF with non-smooth cost functions," *Eng. Appl. Artif. Intell.*, vol. 24, no. 3, pp. 426–436, 2011, doi: 10.1016/j.engappai.2010.10.019.

[11] H. H. A. El-Fergany, "Tree-seed algorithm for solving optimal power flow problem in large-scale power systems incorporating validations and comparisons," *Appl. Soft Comput. J.*, vol. 64, no. 5, pp. 307–316, 2018, doi: 10.1016/j.asoc.2017.12.026.

[12] Y. Sönmez and N. Yörükeren, "Optimal power flow using gravitational search algorithm," vol. 59, pp. 86–95, 2012, doi: 10.1016/j.enconman.2012.02.024.

[13] P. P. Kumar and R. P. Saini, "Environmental Effects Optimization of an off-grid integrated hybrid renewable energy system with various energy storage technologies using different dispatch strategies ARTICLE HISTORY," *Energy Sources, Part A Recover. Util. Environ. Eff.*, vol. 00, no. 00, pp. 1–30, 2020, doi: 10.1080/15567036.2020.1824035.

[14] P. P. Kumar and R. Prasad, "Optimization of an off-grid integrated hybrid renewable energy system with different battery technologies for rural electrification in India," *J. Energy Storage*, vol. 32, no. March, p. 101912, 2020, doi: 10.1016/j.est.2020.101912.

[15] T. Das, R. Roy, and K. K. Mandal, “Impact of the penetration of distributed generation on optimal reactive power dispatch,” vol. 5, 2020.

[16] A. Pal, A. K. Chakraborty, and A. R. Bhowmik, “Optimal Placement and Sizing of DG considering Power and Energy Loss Minimization in Distribution System,” vol. 12, no. 3, pp. 624–653, 2020, doi: 10.15676/ijeei.2020.12.3.12.

[17] H. Tehzeeb-ul-hassan, M. Faizan, K. Mehmood, K. Mehmood, A. H. Milyani, and Q. Rasool, “Optimization of power flow by using Hamiltonian technique,” *Energy Reports*, vol. 6, pp. 2267–2275, 2020, doi: 10.1016/j.egyr.2020.08.017.

[18] K. Girigoudar and L. A. Roald, “On the impact of different voltage unbalance metrics in distribution system,” *Electr. Power Syst. Res.*, vol. 189, no. October 2019, p. 106656, 2020, doi: 10.1016/j.epsr.2020.106656.

[19] A. A. Gli, “Optimal power flow with SVC devices by using the artificial bee colony algorithm,” pp. 341–353, 2016, doi: 10.3906/elk-1305-55.

[20] B. Bentouati *et al.*, “Optimizing performance attributes of electric power systems using chaotic salp swarm optimizer,” *Int. J. Manag. Sci. Eng. Manag.*, vol. 15, no. 3, pp. 165–175, 2020, doi: 10.1080/17509653.2019.1677197.

[21] C. Zhong and G. J. Cokkinides, “Object-Oriented Voltage Control for AC-DC Hybrid Distribution Systems,” 2018.

[22] T. T. Nguyen, T. T. Nguyen, M. Q. Duong, and A. T. Doan, *Optimal operation of transmission power networks by using improved stochastic fractal search algorithm Fuel cost*, vol. 32, no. 13. Springer London, 2020.

[23] K. Mahender, M. Vyshnavi, S. Patange, and H. Pulluri, “Solution of optimal power flow problem using colliding bodies optimization Solution of Optimal Power Flow Problem Using Colliding Bodies Optimization,” vol. 030036, no. October, 2020.

[24] A. Abdollahi, A. A. Ghadimi, M. R. Miveh, and F. Mohammadi, “Optimal Power Flow Incorporating FACTS Devices and Stochastic Wind Power Generation Using Krill,” no. 1.

[25] Y. Zhang and Z. Ren, “Optimal Reactive Power Dispatch Considering Costs of Adjusting the Control Devices,” vol. 20, no. 3, pp. 1349–1356, 2005.

[26] N. Sadati, T. Amraee, and A. M. Ranjbar, “A global Particle Swarm-Based-Simulated Annealing Optimization technique for under-voltage load shedding problem,” *Appl. Soft Comput. J.*, vol. 9, no. 2, pp. 652–657, 2009, doi: 10.1016/j.asoc.2008.09.005.

[27] T. Ding, C. Li, F. Li, T. Chen, and R. Liu, “A bi-objective DC-optimal power flow model using linear relaxation-based second order cone programming and its Pareto Frontier,” *Int. J. Electr. Power Energy Syst.*, vol. 88, pp. 13–20, 2017, doi: 10.1016/j.ijepes.2016.11.012.

[28] M. Ghasemi, S. Ghavidel, S. Rahmani, A. Roosta, and H. Falah, “Engineering Applications of Arti fi cial Intelligence A novel hybrid algorithm of imperialist competitive algorithm and teaching learning algorithm for optimal power fl ow problem with non- smooth cost functions,” *Eng. Appl. Artif. Intell.*, vol. 29, pp. 54–69, 2014, doi: 10.1016/j.engappai.2013.11.003.

[29] C. J. Hatziadoniu, “Selection of Most Effective Control Variables for Solving Optimal Power Flow Using Sensitivity Analysis in Particle Swarm Algorithm.”

[30] H. Abdi, S. D. Beigvand, and M. La Scala, “A review of optimal power flow studies applied to smart grids and microgrids,” *Renew. Sustain. Energy Rev.*, vol. 71, no. May 2015, pp. 742–766, 2017, doi: 10.1016/j.rser.2016.12.102.

[31] Q. Y. Jiang, H. D. Chiang, C. X. Guo, and Y. J. Cao, “Power-current hybrid rectangular formulation for interior-point optimal power flow,” *IET Gener. Transm. Distrib.*, vol. 3, no. 8, pp. 748–756, 2009, doi: 10.1049/iet-gtd.2008.0509.

[32] G. TORRES and V. QUINTANA, “AN INTERIOR-POINT METHOD FOR NONLINEAR OPTIMAL POWER FLOW USING VOLTAGE RECTANGULAR COORDINATE,” *IEEE Trans. Power Syst.*, vol. 13, no. 4, pp. 1211–1218, 1998.

[33] O. Bozorg-haddad, M. Solgi, and H. A. Loáiciga, “Genetic Algorithm Mapping the Genetic Algorithm (GA) to Natural Evolution,” no. 2005, pp. 53–67.

[34] H. Lu, J. Chen, and L. Guo, “Energy Quality Management,” *Compr. Energy Syst.*, vol. 5, 2018, doi: 10.1016/B978-0-12-809597-3.00521-6.

[35] S. Sun and H. Liu, *6 Particle Swarm Algorithm : Convergence and Applications*. Elsevier Inc., 2013.

[36] M. Dorigo, M. Birattari, and T. Stutzle, *Ant Colony Optimatization*. 2006.

[37] M. Schlüter, M. Gerdts, and J. J. Rückmann, “A numerical study of MIDACO on 100 MINLP benchmarks,” *Optimization*, vol. 61, no. 7, pp. 873–900, 2012, doi: 10.1080/02331934.2012.668545.

[38] M. Schlüter, J. A. Egea, and J. R. Banga, “Extended ant colony optimization for non-convex mixed integer nonlinear programming,” *Comput. Oper. Res.*, vol. 36, no. 7, pp. 2217–2229, 2009, doi: 10.1016/j.cor.2008.08.015.

[39] Q. Askari, I. Younas, and M. Saeed, “Knowledge-Based Systems Political Optimizer : A novel socio-inspired meta-heuristic for global,” *Knowledge-Based Syst.*, vol. 195, p. 105709, 2020, doi: 10.1016/j.knosys.2020.105709.

[40] J. Luiz *et al.*, “Lichtenberg Algorithm: A Novel Hybrid PHYSICS-Based Meta-Heuristic For Global Optimization,” *Expert Syst. Appl.*, p. 114522, 2020, doi: 10.1016/j.eswa.2020.114522.

[41] L. Niemeyer, L. Pietronero, and H. J. Wiesmann, “Fractal Dimension of Dielectric Breakdown,” *Phys. Rev. Lett.*, vol. 52, no. 12, 1984.

[42] T. A. Witten and L. M. Sander, “Diffusion-limited aggregation,” vol. 27, no. 9, 1983.

[43] V. Suresh, P. Janik, J. M. Guerrero, Z. Leonowicz, and T. Sikorski, “Microgrid Energy Management System with Embedded Deep Learning Forecaster and Combined Optimizer,” *IEEE Access*, vol. 8, pp. 1–1, 2020, doi: 10.1109/access.2020.3036131.

[44] J. Soares, T. Sousa, Z. A. Vale, H. Morais, and P. Faria, “Ant colony search algorithm for the optimal power flow problem,” *IEEE Power Energy Soc. Gen. Meet.*, pp. 1–8, 2011, doi: 10.1109/PES.2011.6039840.

[45] S. Duman, U. Güvenç, Y. Sönmez, and N. Yörükeren, “Optimal power flow using gravitational search algorithm,” *Energy Convers. Manag.*, 2012, doi: 10.1016/j.enconman.2012.02.024.

[46] K. Srilakshmi, P. R. Babu, and P. Aravindhababu, “An enhanced most valuable player algorithm based optimal power flow using Broyden ’ s method,” *Sustain. Energy Technol. Assessments*, vol. 42, no. May, p. 100801, 2020, doi: 10.1016/j.seta.2020.100801.

4. Energy Management (Economic Dispatch)

It was shown in the previous chapter that the optimization algorithm chosen to carry out energy management activities for the economic dispatch problems in microgrids will be MIDACO integrated with MATPOWER. The reason for this as already explained before is its performance with regard to both the quality of the global minimum solution and speed of convergence to a solution.

The modern microgrid, its typical components and complete architecture was presented in the introduction section. This chapter will begin with mathematical modelling of the various components. This is necessary since data pertaining to all components are not available and the study enables analysis before future expansion of the microgrid at Wroclaw University of Science and Technology.

4.1 Generators

4.1.1 Solar PV outputs

In the microgrid model the data from the solar PV panels remain as they are and are not varied by the optimization algorithm. This data is obtained every 10 minutes from the installation at the Faculty of Electrical Engineering in the University. The description of the installation already mentioned is also repeated here. In total there are 3 different modules which are polycrystalline, monocrystalline and CIGS that consist of 27, 21 and 56 panels. The setup is quite elaborate and numerous variables are measured which are the PV module temperatures (^oC), wind speed (m/s), irradiation (W/m^2) and ambient temperature (^oC).

4.1.2 Battery storage system (BSS)

The algorithm by which the battery storage system operates in the microgrid is the same as explained in the microgrid in the previous chapter. It is modelled according to the recommendations of [1], [2]. The algorithm is shown in Fig. 1. The equations representing its mathematical model are as follows.

Battery charging:

$$E_s(t) = (1-\sigma) \times E_s(t-1) + \left(E_g(t) - \frac{E_l(t)}{\eta_{\text{conv}}}\right) \times \eta_{\text{cc}} \times \eta_{\text{rbat}} \qquad (1)$$

Where, $E_s(t)$ and $E_s(t-1)$ represent the storage system energy levels at time steps t and $t-1$ respectively. The self-discharge rate of the battery is represented by σ. The total energy generated within the microgrid is represented by E_g whereas the total energy demand withing the microgrid is represented by E_l. The efficiencies of the converter and charge controller are represented by η_{conv} and η_{cc}. The round-trip efficiency of the battery is represented by η_{rbat}.

The charging of the battery usually takes place when there is excess energy in microgrid whereas discharging of the battery takes place when there is excess demand in the microgrid.

Battery discharging:

$$E_s(t) = (1-\sigma) \times E_s(t-1) + \left(\frac{E_l(t)}{\eta_{\text{conv}}} - E_g(t)\right) / \eta_{\text{rbat}} \qquad (2)$$

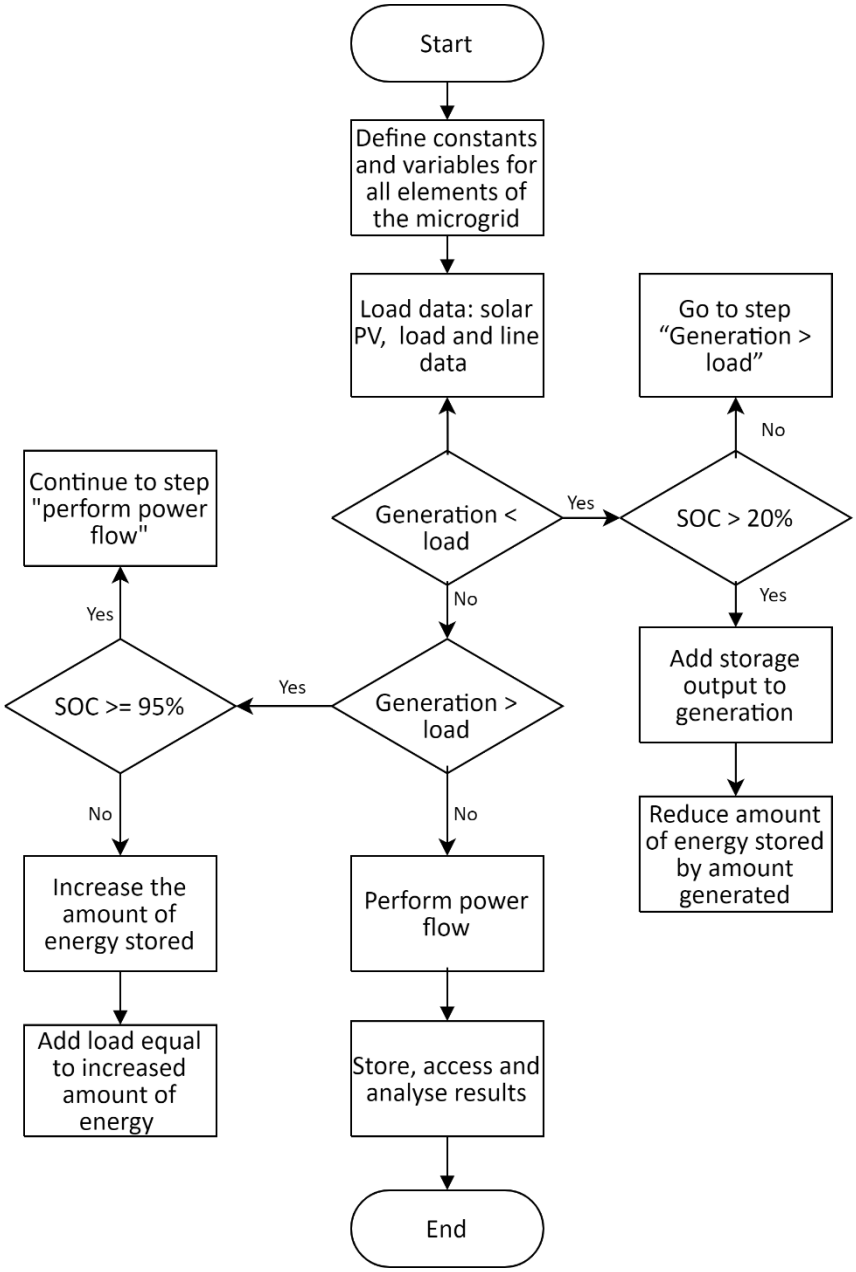

Fig. 1 Battery management system

4.1.3 Fuel Cell (FC) & Hydrogen storage system (HSS)

The HSS consists of numerous elements such as a fuel cell, storage tanks for hydrogen and an electrolyser. The trigger for this system remains similar to that of the storage system. When there is excess energy within the microgrid the electrolyser operates utilising the excess energy and converting water into oxygen and hydrogen. This hydrogen is then compressed via compressors and is then stored in hydrogen tanks. Mathematically, the stored hydrogen can be expressed in terms of energy equivalents by the equation shown below [2].

$$E_{hs}(t) = E_{hs}(t-1) + \left(E_g(t) - \frac{E_l(t)}{\eta_{\text{conv}}}\right) \times \eta_{\text{EL}} \quad (3)$$

Where, the amount of energy in the hydrogen storage tanks at time steps t and $t-1$ is represented by $E_{hs}(t)$ and $E_{hs}(t-1)$. The efficiency of the electrolyser is represented by η_{EL}.

When there is excess demand in the microgrid, similar to the BSS the HSS produces power. In this case this operation is performed by the fuel cell. The fuel cell consists of two electrodes

situated on either side of the electrolyte. During discharge a potential difference is created between the anode and the cathode when hydrogen is supplied to the anode and Oxygen is supplied to the cathode. During this process electricity, heat and water are produced which can be utilised. The chemical equation is quite simple and is shown as follows: $2H_2 + O_2 \rightarrow 2H_2O + electrical\ energy + heat$. A simple illustration of the process is shown in Fig. 2.

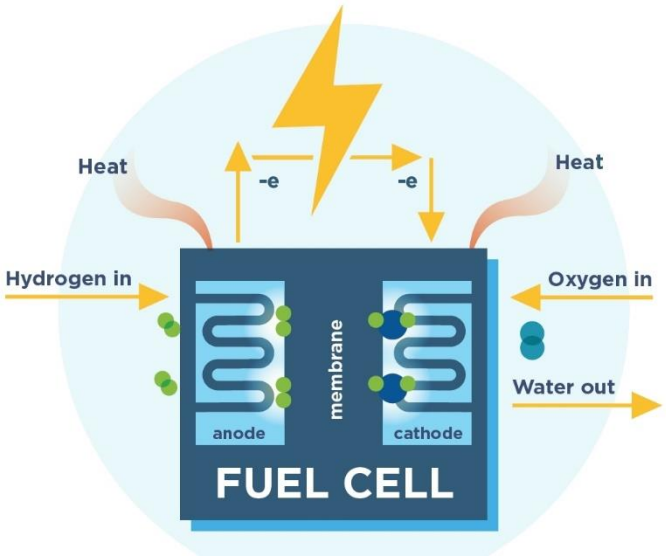

Fig. 1 Fuel cell mechanism

Source: http://www.fchea.org/h2-day-2019-events-activities/2019/8/1/fuel-cell-amp-hydrogen-energy-basics

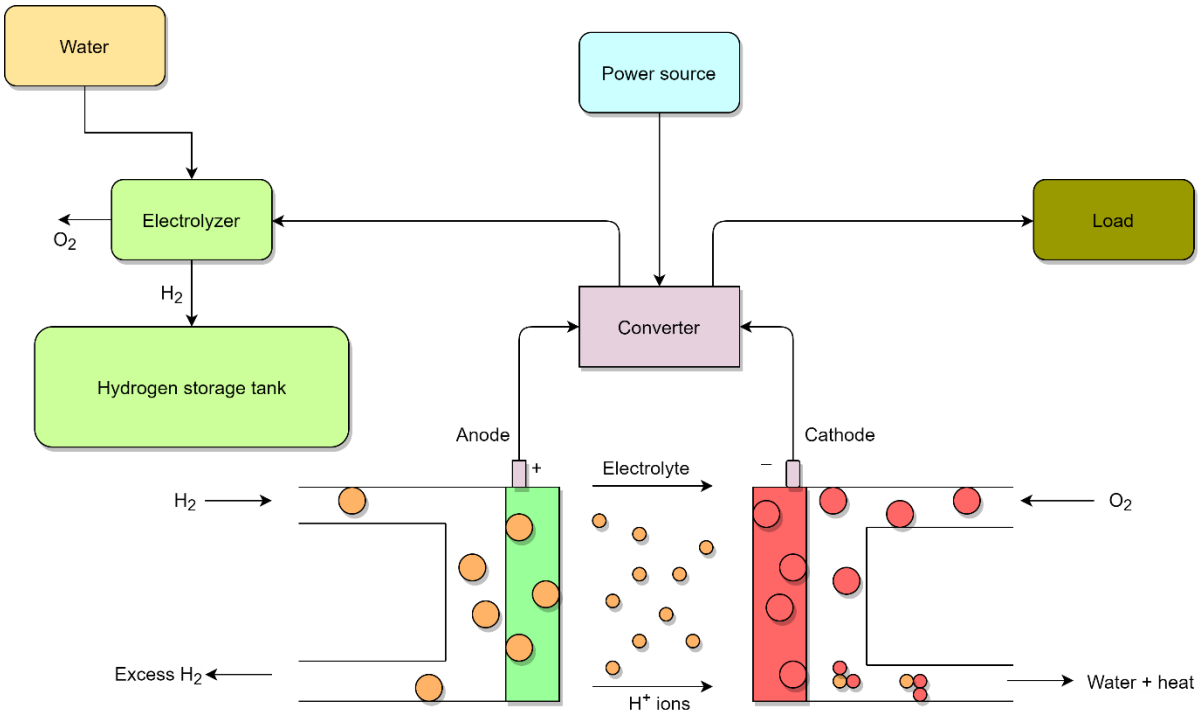

Fig. 2 Hydrogen storage system with fuel cell

The discharging process is described by equation (4):

$$E_s(t) = \ (1-\sigma) \times E_s(t-1) + \left(\frac{E_l(t)}{\eta_{\text{conv}}} - \ E_g(t)\right)/\eta_{\text{FC}} \quad (4)$$

Where the efficiency of the fuel cell is represented by η_{FC} .

The entire process of charging the hydrogen storage tanks, functioning of the electrolyser and the fuel cell is shown in a workflow in Fig. 2.

4.1.4 Diesel Generator (DG)

DGs represent a crucial component in microgrids especially stand-alone microgrids. When there is no connection with the main grid the stability of the microgrid becomes precarious as it will be subject to all the uncertainties concerned with renewable energy sources such as solar, wind etc. While batteries can provide a steady stream of power their integration into the microgrid is often expensive and cannot provide stable reliable power for extended time periods. Hence, the DG comes into the picture to improve the stability of the grid as it is also able to handle quick changes in power demand within its capacity.

The DG fuel consumption can be calculated using a linear function as shown below [2].

$$F_{DG}(t) \ = (a_{DG} \ \times \ P_{DGgen}(t) \ + \ b_{DG} \ \times \ P_{DGrat}) \, l/h \quad (5)$$

Where, a_{DG} and b_{DG} are the coefficients concerned with fuel consumption. Whereas P_{Dggen} and P_{DGrat} are the hourly power produced and rated power of the DG set [3].

The CO_2 emissions associated with the DGs is be expressed by (6)

$$CO_2(t) \ = \ SE_{CO_2}(kg/l) \times F_{DG}(t) \ (l/h) \quad (6)$$

4.1.5 Micro hydro power plant (WT)

Micro hydro plants are waterpower plants with capacities lower than 0.5 MW, in recent times their adoption has increased [4]. Unlike traditional mega hydro power plants, they do not require a significant amount of capital investment and changes to landscape. They operate efficiently in smaller streams with low heads and average discharge of water.

The electric power produced by a waterpower plant can be defined by equation (7) and (8):

$$P_{hgen} \ = \ \eta \ \times \ \rho \ \times g \ \times h \ \times Q \quad (7)$$

$$Q \ = \ A \ \times V \quad (8)$$

Where, P_{hgen} is the electrical power output generated by the plant. η is the efficiency of the turbine used. This study utilises a Francis turbine that for micro hydro plants has an efficiency from 80 – 90%. The efficiency taken in this study is 80%, similar to others found in the literature[4]. ρ is the density of water at 998 kg/m^3, g is the acceleration due to gravity at 9.8 m/s^2, h is the head of the water resource measure in m and for this study the head is at 12 m and Q is the discharge of water measure in m^3/s. For discharge calculation, A is the cross-sectional area of the stream measured in m^2 whereas the velocity of the stream is measured in m/s.

For this study it is assumed that the discharge of the water is fully controllable and that water is available in the stream year-round. The range of discharge is maintained between 0.7 and

0.11 m^3/s. This allows the energy management system to take power from the waterpower plants as decided by the optimization algorithm based on its objective function.

4.2 Economic analysis

In order to ensure profitability and feasibility of the microgrid setup it is imperative to conduct an economic analysis of the entire installation. In this direction the study has decided to use the Levelized Cost of Energy (LCOE) metric for comparing costs from different sources of energy present in the microgrid. The LCOE is a widely adopted benchmark in order to compare the cost-effectiveness of generators from different sources of energy. It's popular since it is able to remove biases associated in cost comparison of different energy technologies [2], [5]–[7]. It is important to mention that the LCOE includes numerous costs incurred throughout the lifetime of the project and considers the total energy produced over its lifetime to arrive at a price per unit of energy produced for a particular generator. It does not include calculations from various financial instruments and risks that might be part of a particular project. The LCOE is a metric that is static in nature. It does not vary overtime while it is very well known that electricity prices in the market are very dynamic. Moreover, the metric is calculated based on extrapolations of the total energy produced and costs over long time periods such as 20, 25 and 30 years and it is important to keep in mind that there can be significant changes and events during such time periods. Also, the LCOE itself varies from region to region and from project to project since renewable energy generators vary in performance from region to region and also their associated efficiencies. Hence, it is a metric that is more accurate when determined by the project proponents themselves.

The LCOE is defined in (9):

$$LCOE = \frac{\sum_{t=0}^{T} C_t/(1+r)^t}{\sum_{t=0}^{T} E_t/(1+r)^t} \quad (9)$$

Where, C_t represents the total costs per year and the numerator represents the sum of the present value of the costs incurred over the lifetime of the project whereas E_t represents the total energy produced in a year and the denominator represents the sum of the present value of the energy produced over the lifetime of the project. T is total number of years; t represents the year under consideration whereas r is the discount rate. It is interesting to note that the summation begins from zero in the above formula, this is done to include any down payment or initial costs at the beginning of the project. These are costs that should not be discounted and be kept out of the summation. It is a manner of representation [5]. A more comprehensive breakdown of the above formula is shown below.

The LCOE includes cash outflows such as the capital costs of the components in the microgrid, costs associated with their installation, fuel costs for the generators utilising fuel and operation and maintenance costs. Out of these for this study the capital and installation costs are treated as one-time costs whereas fuel and operation and maintenance costs are yearly and are discounted over the lifetime of each component. The extended version of (9) with all costs is shown in (10).

$$LCOE = \frac{C_c + I_c + \sum_{t=1}^{T} (F_{ct} + O\&M_{ct})/(1+r)^t}{\sum_{t=1}^{T} E_t/(1+r)^t} \quad (10)$$

Where, C_c is the total capital cost, I_c is the installation cost, F_c is the fuel cost per year, $O\&M_c$ is the operation and maintenance cost per year, E is the energy produced in a year, t is the year under consideration, T is the total number of years and r is the discount rate.

It is important to keep in mind that even though from (9) and (10) it looks as if the energy produced is being discounted, it is actually the LCOE that is discounted. The equations are just rearranged in a convenient manner.

4.2.1 Capital costs

The microgrid to be considered in the succeeding section consists of all types of generators previously described. The capital cost calculation for all the generators are as follows.

$$C_{cpv} = N_{pv} \times C_{pv} \tag{11}$$

$$C_{cbss} = N_{bss} \times C_{bss} \tag{12}$$

$$C_{cfc} = N_{fc} \times C_{fc} \tag{13}$$

$$C_{cdg} = N_{dg} \times C_{dg} \tag{14}$$

$$C_{cwt} = N_{wt} \times C_{wt} \tag{15}$$

Where, $C_{cpv}, C_{cbss}, C_{cfc}, C_{cdg}, C_{cwt}$ are the total capital costs of the solar PV panels, the battery storage system, the fuel cell apparatus, the diesel generators and the water turbines. The capital cost calculations for the solar pv panels are quite straightforward where N_{pv} represents the total number of pv panels installed whereas C_{pv} is the cost per pv panel. Similarly, the cost of a dg unit and a water turbine are represented by C_{dg} and C_{wt} and the number of units are represented by N_{dg} and N_{wt}. The cost of the fuel cell (C_{cfc}) is more elaborate since it includes the capital cost of not only the fuel cell but also the hydrogen storage tank and the electrolyser. $N_{fc}$is the total number of such units. C_{bss} and N_{bss} represent the cost of a storage unit and the total number of units respectively.

4.2.2 Installation, operation and maintenance costs and fuel costs

The installation cost (I_c) is taken as 20%, 5%, 3%, 3.5% and 10% of the total capital cost for the solar pv panels, BSS, FC, DG and WT.

The operation and maintenance costs are quite different for different energy technologies. Generating systems such as DGs which have numerous moving parts have higher maintenance costs. Moreover, the microgrid is setup with a lifetime of 25 years in mind hence, certain components need to be replaced.

For the solar pv panels, the calculations are quite straightforward. Per panel the maintenance costs come to about 6.5 $/yr [2]. Since the inverters associated with the panels have a life span of about 10 years they have to be replaced twice. Hence, the O&M costs include twice replacement of inverters as well.

The O&M costs for the BSS are little to none since it has not got any moving parts but the standard Li-ion battery has a life span of 5000+ cycles. In order to determine the life of the Li-ion BSS in this study, simulations were run for an entire year and the number of cycles were calculated. The authors are aware that since the BSS runs of excess energy produced within the microgrid this value can change year after year since the energy produced is very much weather dependent. Overall analysis from the 5-year solar pv data available at the Wroclaw University of Science and Technology has shown that yearly power produced by all solar panels does not vary significantly. Apart from this the waterpower plants have a more stable power output throughout the year. From the analysis it was seen that yearly the Li – ion BSS has a total of

671.5 cycles. With this number in mind and assuming a minimum of 5000 cycles before replacement. The lifespan of the BSS comes out to be 7.4 years. Hence, the BSS would be replaced 3 time within the span of 25 years. The fuel costs for the BSS would be based on the power used for charging the BSS [8].

The yearly O&M costs for the fuel cell based generating system is a sum of the O&M costs for the fuel cell, electrolyser and the HST. The costs are 120 $/kw/yr, 16$/kw/yr and 5 $/yr respectively for the FC, electrolyser and the HST respectively [2]. The life of a stationary fuel cell can be as high as 40,000 hours. Hence, given that it works on the same principle as the BSS the total number of cycles are much higher than the 25 years for which the LCOE calculations are made. Therefore the FC will not be replaced but the electrolyser has a lifespan of 15 years [2] and will be replaced once. The fuel costs for the FC would be the power used for conversion of excess power into hydrogen by the electrolyser and storing it in HST. [8].

The DGs used in this study are 2 Hyundai 5.2 kW/ 6.5 kVA diesel generators. The generator has a 16-litre fuel tank and the minimum output power in this study is 30% of the rated power which is 1.56 kW. Due to the unpredictable nature of energy sources in the microgrid the DG provides stability and security in terms of its operation. Due to this fact the DG is run throughout the year and is allowed to vary between its minimum and maximum output power. This results in significant fuel costs, O & M costs and replacement costs. The O&M costs are taken as 510 $/unit/yr. Since the life of a DG unit is about 20,000 hrs, the DG's are replaced every 2 years up to 25 years. The fuel cost was calculated looking at the operation of DG for an entire year and using the formula (5).

The O&M for the Francis turbine of the water power plant is taken as 8% of the capital cost per year [4]. The life of such a turbine is about 30 years hence, it does not need a replacement.

4.3 Microgrid Layout

The microgrid layout is shown in Fig 3. There are a total of 13 loads and 10 generators. The generators are a mix of both renewable sources of energy, energy storage systems and diesel generators. There is a total of 4 solar pv modules with a peak power of 12 kW, 15 kW, 6 kW and 15.9 kW. They are located of nodes 1, 2 and 3. There are 2 DG units with a 5.2 kW/6.5 kVA rating. One each is located at node 2 and node 5 respectively. A li – ion BSS is located at node 3 with a capacity of 9.8 kWh and a fuel cell with a hydrogen storage tank with an equivalent capacity of 4.2 kWh is located in node 5. A micro – hydro power plant with a max power output of 11.63 kW is located at node 4. While the power produced by the solar panels remain fixed, other sources of energy can be controlled. There are a total of 13 loads, two of which represent the power consumed by the BSS and the FC + HST. This is usually observed during their charging process. Other loads are distributed across all nodes and have a varying power factor. The source for reactive power in the microgrid are only from the main grid and the DG sets.

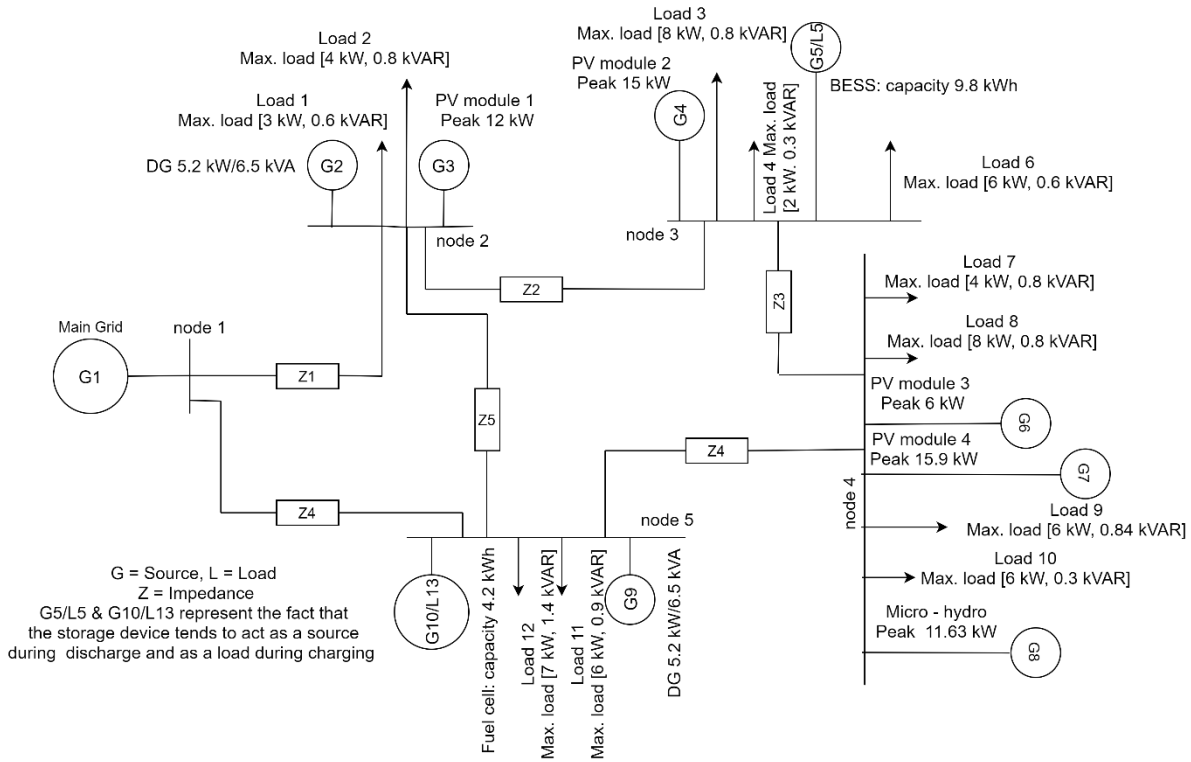

Fig. 3 Microgrid layout

The line data of the cables used are shown in table 1. They are aluminium cables that have an XLPE insulation with a PVC outer sheath having a single core with $120mm^2$ cross-section.

Table 1: Cable data

From	To	Distance (m)	r + jx (Ω) (10^{-1})
node 1	node 2	200	0.5060 + 0.1630j
node 2	node 3	110	0.2780 + 0.0900j
node 3	node 4	135	0.3420 + 0.1100j
node 4	node 5	215	0.5440 + 0.1760j
node 5	node 1	150	0.3790 + 0.1230j
node 2	node 5	250	0.6330 + 0.2040j

4.4 Problem Formulation

The objective function used for the EMS of the microgrid is shown in (18). It attempts to minimise the overall cost of energy in the microgrid. T_c is the total cost, C_i represents the LCOE of each generator in the microgrid and P_i represents the power produced by each generator. N_{gen} is the total number of generators in the microgrid. It should be mentioned here that for the power taken from the main grid (P_1) there is no LCOE but hourly grid prices (C_1) are taken from the European Network of Transmission System Operators (ENTSO-E) website.

$$Min(T_c) = \sum_{i=1}^{N_{gen}} C_i P_i + P_1 C_1 \tag{16}$$

$\sum_{i=1}^{N_{gen}} C_i P_i$ when expanded looks like the equation (17).

$$\sum_{i=1}^{N_{gen}} C_i P_i = C_{solar} P_{solar} + C_{BSS} P_{BSS} + C_{FC} P_{FC} + C_{DG} P_{DG} + C_{hydro} P_{hydro} \tag{17}$$

Where, $C_{solar}, C_{BSS}, C_{FC}, C_{DG}, C_{hydro}$ are the LCOE for solar panels, BSS, FC + HST, DG and micro hydro plant respectively. $P_{solar}, P_{BSS}, P_{FC}, P_{DG}, P_{hydro}$ are the total output power from the solar panels, BSS, FC + HST, DG and micro hydro plant respectively.

The minimisation of the objective function in (16) is subject to numerous equality and inequality constraints as shown below. (18) and (19) representing the power balance equations for both active and reactive power are the equality constraints whereas (20), (21) and (22) are the inequality constraints. (20) and (21) represents the fact that the amount of active and reactive power produced in every node is constrained within its minimum and maximum whereas (22) represents the fact that the voltage magnitude in every node is constrained within a specified range.

$$P_{gi} = |V_i| \sum_{k=1}^{n} |V_k||Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i) + P_{di} \quad (18)$$

$$Q_{gi} = |V_i| \sum_{k=1}^{n} |V_k||Y_{ik}| \sin(\theta_{ik} - \delta_k - \delta_i) + Q_{di} \quad (19)$$

$$P_{gi}^{min} \leq P_{gi} \leq P_{gi}^{max} \; \forall \; i \; \epsilon \text{ No. of nodes} \quad (20)$$

$$Q_{gi}^{min} \leq Q_{gi} \leq Q_{gi}^{max} \; \forall \; i \; \epsilon \text{ No. of nodes} \quad (21)$$

$$V_i^{min} \leq V_i \leq V_i^{max} \; \forall \; i \; \epsilon \text{ No. of nodes} \quad (22)$$

Where, P_{gi} and Q_{gi} represent the active and reactive power generated in each node. P_{di} and Q_{di} represent the active and reactive power demand in every node. Term $|V_i| \sum_{k=1}^{n} |V_k||Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$ represents the total active power loss in the system and the term $|V_i| \sum_{k=1}^{n} |V_k||Y_{ik}| \sin(\theta_{ik} - \delta_k - \delta_i)$ represents the total reactive power losses in the system. V_i and V_k are the voltage magnitudes at nodes i and k respectively. Y_{ik} is the admittance between nodes i and k. θ_{ik} is the admittance angle between nodes i and k whereas δ_k, and δ_i are the voltage angles at nodes k and i respectively. P_{gi}^{min} is the minimum active power that can be produced in node i whereas P_{gi}^{max} is the maximum active power that can be produced in a node. Q_{gi}^{min} is the minimum reactive power that can be produced in node i whereas Q_{gi}^{max} is the maximum reactive power that can be produced in a node. V_i^{min} is the minimum voltage magnitude permissible in node i whereas V_i^{max} is the maximum voltage that permissible in a node.

Vital to solving the above-mentioned set of equations is to identify appropriate state and control variables [9]. The control variables also called as independent variables serve as the decision variables for MIDACO whereas the state variables are determined by MATPOWER. The methodology of searching the solution has been discussed in detail in chapter 3. In this study the control variables are the voltage magnitudes at all nodes and active power injections at all nodes. The state variables include reactive power outputs at all nodes, voltage angles and slack bus power.

4.5 Economic dispatch

In order to make a wholesome analysis about the functioning of the microgrid and its associated costs, clustering was used to identify representative days across the year. For this purpose, k-means clustering was used and applied to the yearly hourly recorded load data. In order to do so, first the right number of clusters had to be identified which was done with the help of the elbow method. The results are shown in Fig. 4 (a) and it can be seen that the ideal number of clusters for this data set is 3 after which the Within Cluster Sum of Squared Errors (WCSS) does not reduce significantly. The k-means clustering is a well-documented and well-

established clustering algorithm hence, it will not be explained in detail in this section. More on the same can be found in [10].

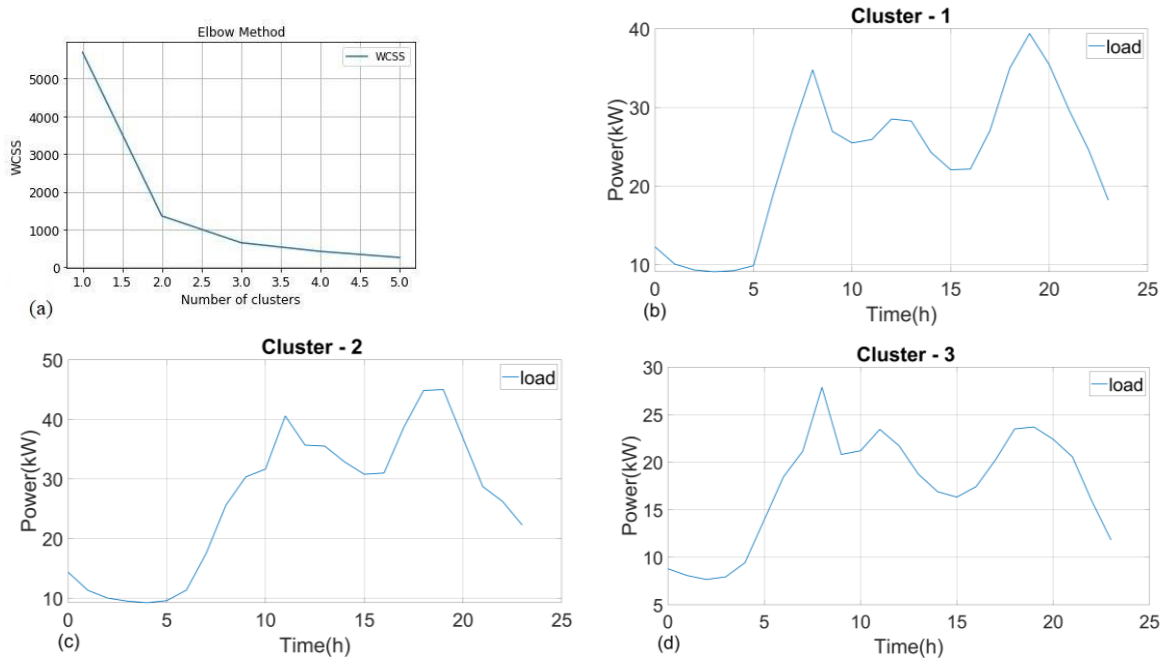

Fig. 4: (a) elbow method (b) load cluster – 1 (c) load cluster – 2 (d) load cluster - 3

The load demand on representative days as a result of clustering can be seen in Fig. 4(b), Fig. 4(c) and Fig. 4(d). It can be seen that over the year the load demand does vary significantly. Fig. 4(b) represents the load demand typical of autumn or spring months where it is in between the load demand seen during the winter and the summer. Fig. 4(c) represents the load demand typically observed during winter wherein it is quite high as the winters are cold. Fig. 4(c) is typical of the load demand observed during the summer months when the weather is pleasant and not too hot. It is important to mention here that the load curves were taken from a nearby village and multiplied by a suitable constant for the purpose of analysis in the microgrid. Hence, the underlying characteristic is from the real world.

Fig. 5 presents the renewable energy sources power output on the same representative days as decided by the clustering algorithm. It can be seen that Fig. 5(a) is characteristic of the spring and autumn where the renewable energy output of the microgrid is in between that of the summer and winter. This is also because the major renewable energy component of the microgrid are the solar panels. Fig, 5(b) is representative of the winter where the output power throughout the day is quite low. Fig. 5(c) is characteristic of the summer where the renewable energy output in the microgrid peaks. In general, the characteristic of the load demand and the renewable energy output seem to be inversely correlated which makes power balancing within the microgrid a complicated issue. Hence, the presence of DGs and connection to the grid are of paramount importance and provide stability.

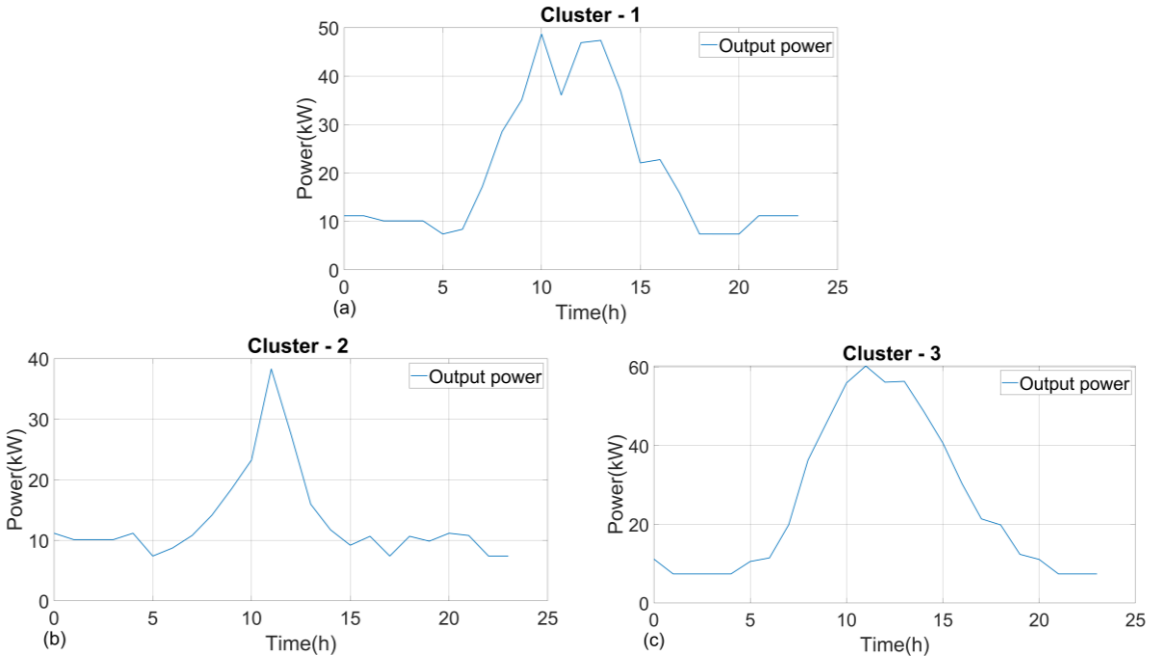

Fig. 5 : renewable energy output (a) cluster – 1 (b) cluster – 2 (c) cluster - 3

Analysing the above results brings 3 scenarios to mind.

Scenario 1: When the renewable energy produced in the microgrid is at par with the load demand.

Scenario 2: When the renewable energy produced in the microgrid is lower than that of the load demand.

Scenario 3: When the renewable energy produced in the microgrid is higher than that of the load demand.

The characteristics of the system and the price signal for each scenario is now discussed.

Scenario 1:

For this scenario, the charging of the BSS and HST are shown in Fig. 6 (left) and Fig. 6 (right) respectively. Both the storage systems operate on the same principal with the exception that during charging the priority is given to the BSS. Hence, during situations when the excess power in the microgrid is less than the maximum charging power of the battery, the entire excess power is sent to the BSS. This is the reason why the BSS charges to a small amount at the beginning and not the HST. Later in the day when the renewable energy sources peak in the microgrid both systems charge to their maximum capacity.

Fig. 7 has two figures depicting the exchange of power with the grid (left) and the price per unit of energy consumed (right). It is quite clear from comparing the two figures that when the power imported from the main grid is low or power is being exported to the main grid the price per unit of energy remains low. This is because there are time periods during which the power from the renewable energy sources remains high. Later in the day when the load peaks in the evening and the renewable energy sources' power output reduces, the amount of power imported from the main grid increases due to which the price per unit of energy also increases. The peak at hour 17 of the day is mainly due to the FC + HST, amongst all LCOE, FC + HST has the highest price.

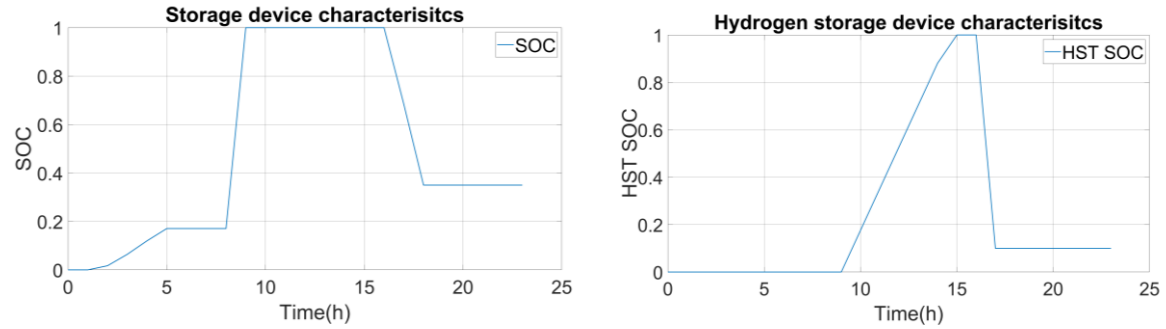

Fig. 6 : storage system characteristics – BSS (left), HST (right)

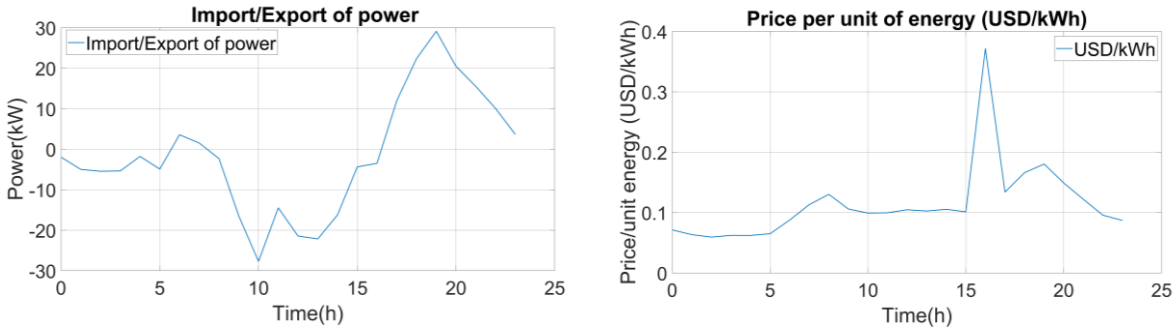

Fig. 7: Exchange of power with the grid (left), price signal (right)

Scenario 2:

During scenario 2 it is clear that the BSS in Fig. 8 (left) and the HST Fig. 8 (right) struggle to reach full capacity, it can even be observed that the HST does not even reach half capacity. This is typical of winter, in fact during many days in the winter the BSS and the HST do not charge at all since the renewable energy produced in the microgrid remains very low due to the low output power from the solar panels.

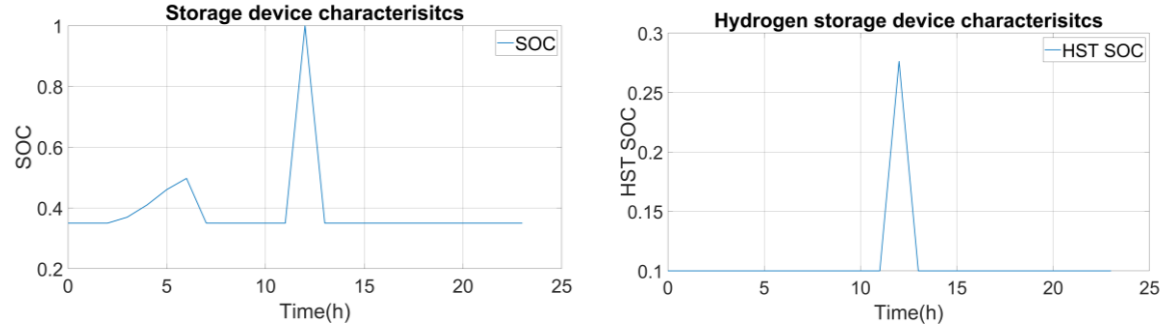

Fig. 8 : storage system characteristics – BSS (left), HST (right)

It can be observed in comparison with scenario 1 (Fig. 7 (right)) that the price per unit of energy in scenario 2 (Fig. 9 (right)) is around the same. In general, it can be seen that the price remains around 0.1 USD/kWh in scenario 1 and scenario 2. The similar price performance in both scenarios is due to the fact that during the day on scenario 1 the price of the grid is very close to the cost-coefficient of the energy produced by the solar panels and the fact that the renewable energy produced in the microgrid is still not close to its maximum capacity. In comparison with scenario 1 (Fig. 7 (left)), in this scenario there is little to no export of power to the main grid. From Fig. 9 (left) it can be seen that much of the load in the microgrid is satisfied by importing power from the main grid. This import of power increases throughout the day and peaks around

the evening during which the price is increases significantly. The peak observed in the price is once again due to the FC + HST.

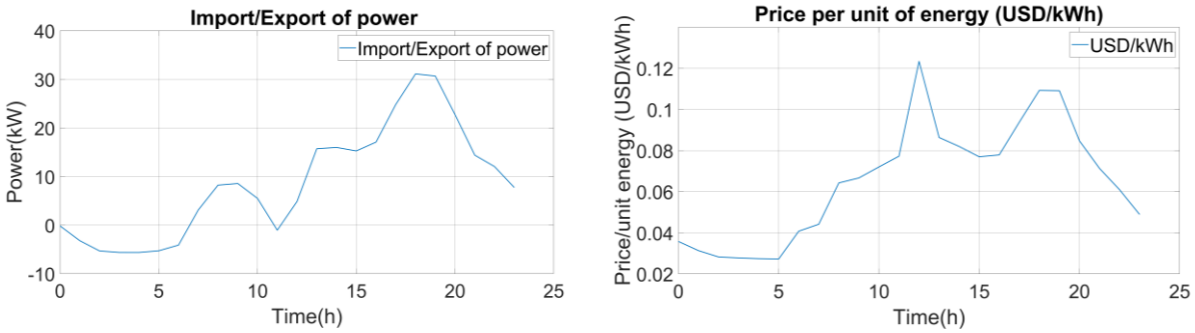

Fig. 9 : Exchange of power with the grid (left), price signal (right)

Scenario 3:

In the earlier two scenarios it was noticed that the HST never charged earlier in the day when usually a small excess in renewable generation is observed. This was because the priority is always given to the BSS over the FC + HST. In this scenario it is observed that there is enough excess energy to charge both the BSS and the FC + HST. It is because this scenario is typical of days in the summer when availability of sunlight is maximum and the total energy produced by the microgrid is quite high. It can be seen from Fig. 10 (left) and Fig. 10 (right) that both storage devices reach maximum charge quite early in the day and stay at full charge for a long time before they discharge when the load demand is greater than the energy produced in the microgrid.

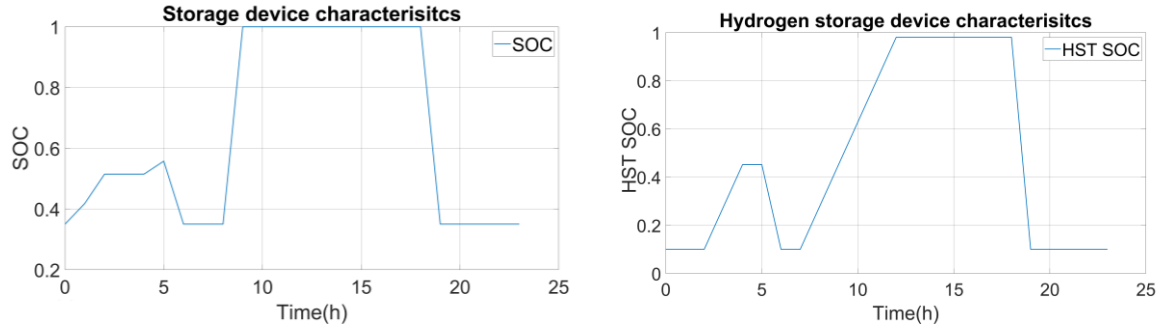

Fig. 10: storage system characteristics – BSS (left), HST (right)

From Fig. 11 (left) it can be seen that for a major portion of the day power is being exported to the main grid. The reason for this is the increased renewable energy production during the summer and also the reduced load demand during the same. Around 20:00 the import of power begins due to increasing load and reduced renewable energy generation. Compared to both scenarios the export of power is higher. Moreover, the price signal also remains well short of 0.04 USD/kWh except for 2 peaks characterising the discharge of the FC + HST. The curve is at a lower position than the price curves at both scenarios since renewable energy produced during the summer is the highest.

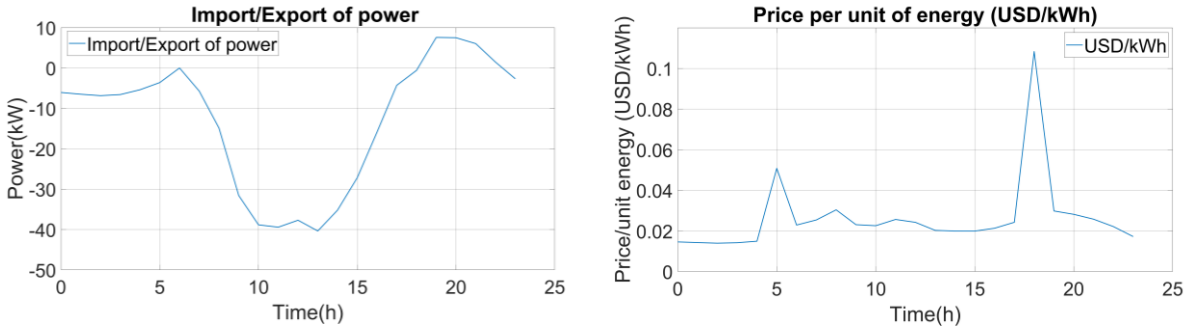

Fig. 11 : Exchange of power with the grid (left), price signal (right)

In general, it can be concluded that scenario 1 is balanced with regard to the microgrid's operation, scenario 2 makes it highly dependent on the main grid with regard to import of power and scenario 3 makes it highly dependent on the main grid with regard to export of power. The microgrid is partially real and a few elements such as the FC + HST is added with an expansion in mind. Based on this analysis decisions are to be made regarding the microgrid and its components and their sizing.

Table 2: Parameters and values for calculating the LCOE for every generator type

Parameters	Value	Parameters	Value
Capital cost of PV panels	0.6 $/kW	Annual O&M for entire installation	141 $
Annual O&M cost of PV	6.5 $/module	Erection cost of FC + HST + electrolyser	5% of capital cost of entire FC installation
Total installed capacity	48.9 kW	Fuel cost of FC + HST	[1]0.033 * Energy consumed
Erection cost PV system	20% of capital cost of PV system	Electrolyser replacement cost	Cost of replacement at capital cost at discounted rate
Capital cost of BSS	1500 $	Capital cost of DG	2099 $/unit
Erection cost of BSS	5% of capital cost of BSS	Total number of units	2
Fuel cost of BSS	[1]0.033 * Energy consumed	Replacement cost	Cost of replacement at capital cost at discounted rate
O & M cost of BSS	Cost of replacement at capital cost at discounted rate	AO & M cost for DGs	1020 $
Capacity of BSS	9.8 kWh	Fuel Cost of DG	Calculated based on yearly fuel consumption
Capital cost of FC	2400 $/kW	Capital cost of Francis turbine	15200 $
Installed capacity of FC	3 kW	Total installed capacity	11.6 kW
Capital cost of electrolyser	800 $/kW	Erection cost for Francis turbine	20% of capital cost of turbine
Installed rating of electrolyser	3 kW	Annual O&M costs	1216 $
Capital cost of HST	600 $	Replacement cost	No replacement, life > 25 years

Table 2 presents the cost components of all microgrid elements and the LCOE for every generating type.

Conclusions:

This chapter attempts to solve the ED problems of the modern microgrid with the objective of reducing costs involving numerous components in the microgrid. It began with the mathematical modelling of numerous sources of energy in the microgrid including the Solar PV panels, BSS, FC + HST, DG and hydro plant. This was followed by determination of the

LCOE for every source of energy which resulted in the micro hydro plant having the lowest LCOE and the FC + HST the highest. The LCOE includes the capital , erection, fuel and operation and maintenance costs and the calculations were made for a time period of 25 years. For obtaining a wholesome picture on the economic benefits of using microgrids the yearly load data was clustered using the k-means algorithms to obtain days representing the whole year. This resulted in 3 clusters which can be imagined as representing the summer, winter and spring and autumn. Based on this, 3 scenarios were created and it was shown clearly that the economic benefits remain high during days of high renewable energy production like in the summer and the benefits remain low during the winter months. This was done by plotting and analysing the price curve for each scenario.

References

[1] Y. Levron, J. M. Guerrero, and Y. Beck, "Optimal power flow in microgrids with energy storage," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 3226–3234, 2013, doi: 10.1109/TPWRS.2013.2245925.

[2] P. P. Kumar and R. P. Saini, "Environmental Effects Optimization of an off-grid integrated hybrid renewable energy system with various energy storage technologies using different dispatch strategies ARTICLE HISTORY," *Energy Sources, Part A Recover. Util. Environ. Eff.*, vol. 00, no. 00, pp. 1–30, 2020, doi: 10.1080/15567036.2020.1824035.

[3] A. S. O. Ogunjuyigbe, T. R. Ayodele, and O. A. Akinola, "Optimal allocation and sizing of PV/Wind/Split-diesel/Battery hybrid energy system for minimizing life cycle cost, carbon emission and dump energy of remote residential building," *Appl. Energy*, vol. 171, pp. 153–171, 2016, doi: 10.1016/j.apenergy.2016.03.051.

[4] A. Roque, D. M. Sousa, C. Casimiro, and E. Margato, "Technical and economic analysis of a micro hydro plant - A case study," *2010 7th Int. Conf. Eur. Energy Mark. EEM 2010*, pp. 10–15, 2010, doi: 10.1109/EEM.2010.5558735.

[5] K. Branker, M. J. M. Pathak, and J. M. Pearce, "A review of solar photovoltaic levelized cost of electricity," *Renew. Sustain. Energy Rev.*, vol. 15, no. 9, pp. 4470–4482, 2011, doi: 10.1016/j.rser.2011.07.104.

[6] W. Short and D. J. Packey, "A Manual for the Economic Evaluation of Energy Efficiency and Renewable Energy Technologies," no. March, 1995.

[7] R. Jones-albertus, D. Feldman, R. Fu, K. Horowitz, and M. Woodhouse, "Technology advances needed for photovoltaics to achieve widespread grid price parity," no. 2016, pp. 1–12, 2020, doi: 10.1002/pip.

[8] O. Schmidt, A. Hawkes, O. Schmidt, S. Melchior, A. Hawkes, and I. Staffell, "Projecting the Future Levelized Cost of Electricity Storage Technologies Projecting the Future Levelized Cost of Electricity Storage Technologies," *Joule*, vol. 3, no. 1, pp. 81–100, 2019, doi: 10.1016/j.joule.2018.12.008.

[9] C. J. Hatziadoniu, "Selection of Most Effective Control Variables for Solving Optimal Power Flow Using Sensitivity Analysis in Particle Swarm Algorithm."

[10] A. Geron, "Neural networks and deep learning" in Hands-On Machine Learing With Scikit-Learn & Tensor Flow, O'Reilly Media, 2017. Available: https://learning.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

5. Electrical Vehicle Charging Station (EVCS) and optimal location

One of the planned additions to the microgrid apart from the FC + HST is the EVCS. Since already an economic analysis of the microgrid was performed. This section will deal with a model of the microgrid from the technical point of view. Faculty of electrical engineering in Wroclaw University of Science and Technology has been mulling the idea of eventually moving to a microgrid that is standalone. Hence, with that objective in mind the optimization algorithm's objective function is changed. A slightly different architecture of the microgrid will be considered but the underlying data of the generators and the loads is the same. They are multiplied with suitable constants to achieve an appropriate scale of the microgrid.

5.1 Microgrid layout

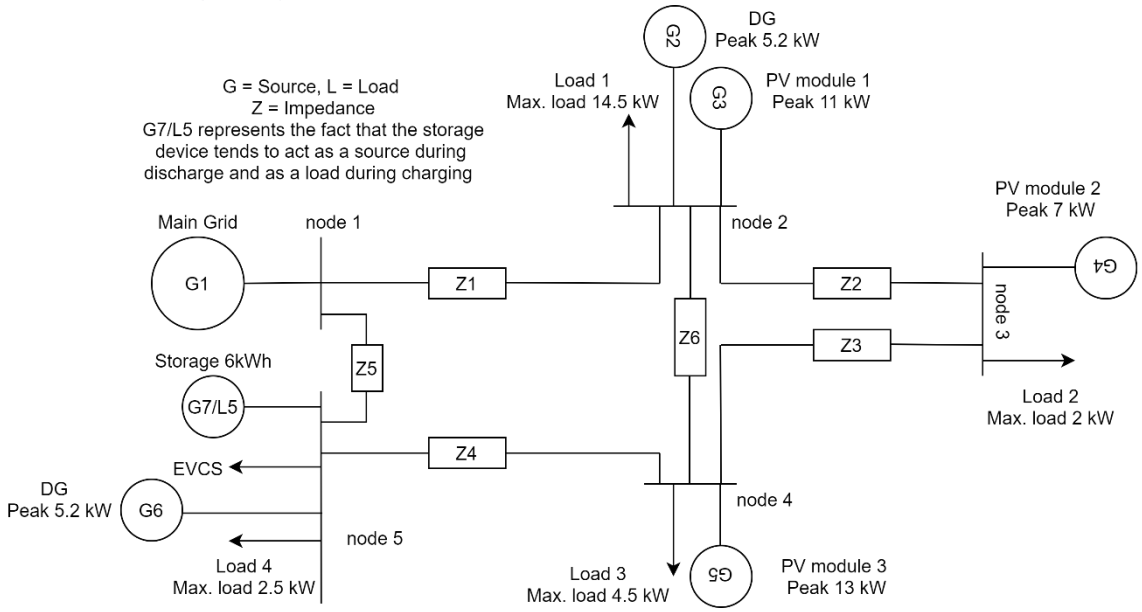

Fig. 1: Microgrid layout

The layout is shown in Fig. 1 and in total has 7 generators. The main grid is represented by G1, it is connected to the microgrid via the point of common coupling PCC. The solar panels that are situated at nodes 2, 3 and 4 are represented by G3 with a peak power of 11 kW, G4 with a peak power of 7 kW and G5 with a peak power of 13 kW respectively. There are 2 DGs in the microgrid located at nodes 2 and 5. They are identical to one another and have a rating of 5.2 kW/6.5 kVA. The DGs as discussed in the previous chapter will be made to run year-round and will be maintained at a minimum of 30% of the rated power.

Table 1. Line data.

From	To	Distance (m)	R + jX ($\cdot 10^{-1}$)
node 1	node 2	180	0.455 + j0.147
node 2	node 3	98	0.248 + j0.080
node 3	node 4	145	0.367 + j0.118
node 4	node 5	205	0.519 + j0.167
node 5	node 1	172	0.435 + j0.140
node 2	node 4	245	0.620 + j0.200

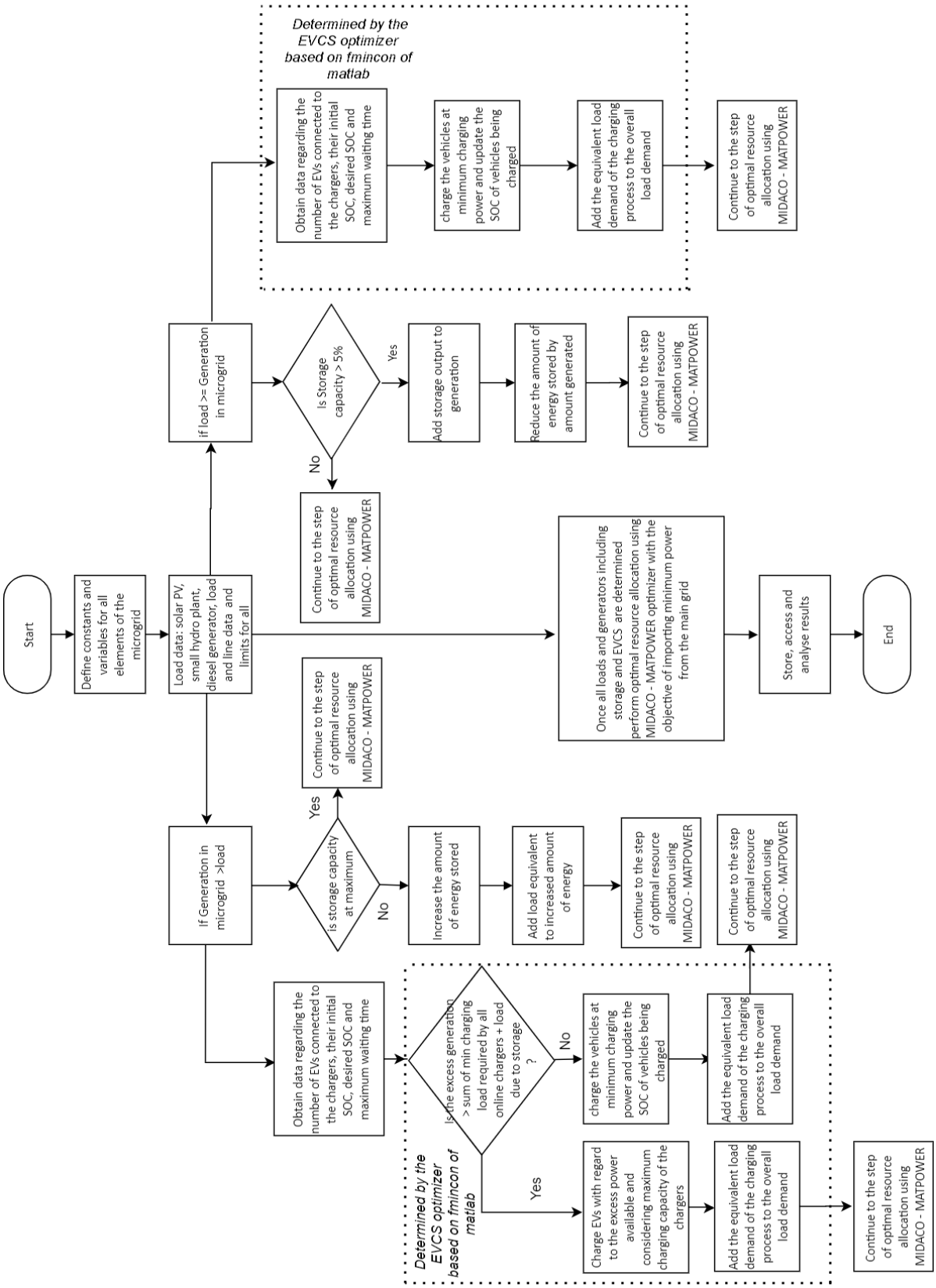

Fig. 2 Energy management procedure

The BSS has a capacity of 9.8 kWh and is situated at node 5 in the Fig. 1 but its optimal position will be determined using the optimization algorithm. It is represented as G7/L5 since it is viewed as a generator during discharge and a load during charging. The line data utilised is shown in table 1. The lines used are of the same type as used earlier but they differ both in length and their interconnection with respect to the nodes in the microgrid. The EVCS in Fig.

1 is located in node 5 but an appropriate node will be selected by running the EMS. This will be explained later in the chapter

5.2 Energy management strategy

The energy management strategy of the EVCS along with the storage in the microgrid is shown in Fig. 2. The strategy can be explained by means of 2 conditions.

Condition 1: When the energy produced in the microgrid is greater than the load demand within the microgrid.

Under this condition firstly, the SOC of the storage device is checked and if it is not fully charged some of the excess power is used to charge the BSS. For the EVCS, the minimum load demand is ascertained. This is done by taking the total sum of the minimum charging power of all the chargers that are connected. If this demand from the EVCS is lower when compared to the excess power available minus the power consumed in charging the storage device, the chargers deliver power at a rate decided by the EMS limited by their individual maximum charging capacities. If the EVCS demand is higher than excess power available minus the power consumed by the storage device, all the chargers are set to deliver power at their minimum rated power.

Condition 2: When the energy produced in the microgrid is lower than the load demand in the microgrid.

Under this condition, first the operation of BSS is decided. In this case if the SOC is greater than 10% it starts discharging. With regard to the EVCS, the minimum load demand is calculated as it was done in the earlier condition and it is added to the overall demand of the system with the connected chargers charging at their minimum rated power.

Once the demand and power produced of all components in the microgrid is decided. The EMS is run where the control variables are decided by MIDACO and the state variables by MATPOWER. The process is the same as the one described in chapter 3 and 4.

5.3 Electrical vehicle charging station model

The EVCS in this study is equipped with 5 Tesla level 2 chargers that can deliver power in the range of 3.7 to 17.2 kW per hour. There is always randomness involved in the operation of an EVCS. In this direction the arrival of EVs, their capacities, their level of charge, their maximum waiting time and the desired SOC input by each EV owner is modelled with the help of gaussian distributions. After which the values for each parameter is selected using the random number generator of MATLAB. The initial SOC for every is obtained from a gaussian curve that has a mean on 0.2 and a standard deviation of 0.2. The desired SOC input by every EV user is represented by a normal curve with a mean of 0.7 and a standard deviation of 0.3. The maximum waiting time has a mean of 6 hours with a standard deviation of 2 hours. The capacity of each EV is also taken from a normal distribution with a mean of 42 kWh and a standard deviation of 10 kWh. 42 kWh was chosen as the mean since it is the capacity of Nissan leaf which is the most popular EV in the city of Wroclaw.

As mentioned earlier the charging rate of the EVs depends upon the two conditions mentioned in the previous section. This optimal charging rate is decided utilising ‘fmincon’ of MATLAB. The EMS is very flexible and the charging rate can also be manually determined by the users of the EMS. For this optimization algorithm the objective function is minimising the power imported from the main grid. As explained earlier this objective function is in line with objectives of the university to eventually move to a microgrid that is standalone. The

minimisation of cost objective has already been discussed in chapter 3. The objective function is shown in (1)

$$\min (P_G) = P_d + P_l - P_{mg} \tag{1}$$

Where, P_G is the total power that is taken from the main grid. P_d is the load demand within the microgrid. P_l is the active power losses in the microgrid and P_{mg} is the power produced in the microgrid.

Furthermore, the constraints and the mathematical equations describing the charging process are as follows.

$$N_{EV}^t \leq N_c^t \tag{2}$$

$$P_{cmin} \leq P_{ct} \leq P_{cmax} \tag{3}$$

$$SOC_i^{initial} \leq SOC_i \leq SOC_i^{desired}, i = 1,2 \ldots\ldots N_{EV} \tag{4}$$

$$add_{SOC} = \Delta t * P_{ct} \tag{5}$$

$$T_c^i \leq \min (T_{max}^i, T_{des}^i) \tag{6}$$

Here, equation (2) represents the fact that the number of EVs connected (N_{EV}^t) to the charging station are always lower than that of the total number of chargers (N_c^t). (3) ensures that power delivery of every charger is fixed within the range of their maximum (P_{cmax}) and minimum rated power delivery (P_{cmin}). The SOC of every EV i is constrained within its initial SOC ($SOC_i^{initial}$) at arrival and the desired SOC value ($SOC_i^{desired}$) input by the user. (5) describes the SOC that is added (add_{SOC}) in every time frame which is equal to the length of the time frame Δt multiplied by the charging power (P_{ct}). The waiting time for every EV i is constrained within the maximum waiting time T_{max}^i input by every user or the time taken to reach the desired SOC (T_{des}^i) whichever is shorter.

It should be kept in mind that the above-mentioned process is only deciding the optimal charging scenario for the EVCS from available options. Apart from this the entire microgrid is also optimized by the EMS described in the previous chapter. The equality and inequality constraints remain the same. The only difference is the objective function which is the same as (1).

5.4 Analysis and optimal location of the EVCS

In general, from Fig. 3 the effects of controlled and uncontrolled charging on the load demand and power imported from the main grid can be seen. From Fig. 3 (left) it can be seen that when the charging is uncontrolled the load demand rises and falls rapidly and that at times the load demand is almost twice than when the charging rate is controlled. This puts tremendous pressure on the power system and would need significant changes to the power system before it is installed. It can be seen from Fig. 3 (right) that much of this excess demand from the EVCS is met by importing power from the main grid when it is in the uncontrolled mode.

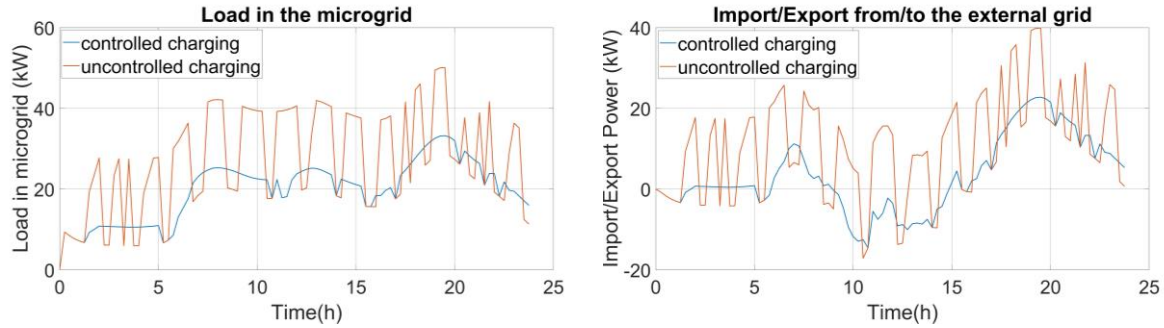

Fig. 3: Microgrid load characteristics (left) and power exchange with the main grid (right).

In the uncontrolled mode the chargers are all charging at their maximum rated power delivery. Using the EMS, it is possible to monitor and track the increase in SOC of the incoming EVs. This process is shown in Fig. 4 wherein the data from 2 chargers are taken.

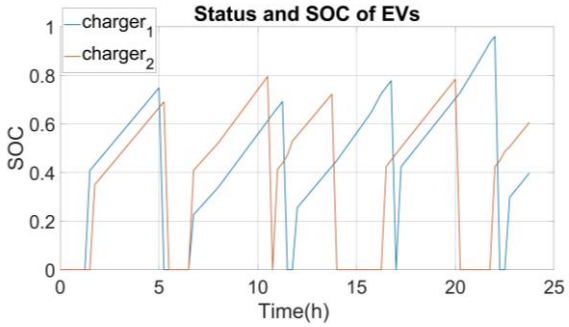

Fig. 4: Charging characteristics of incoming vehicles at chargers 1 and 2.

It can be seen that numerous EVs with different initial SOCs utilise the EVCS and they have different times for which they remain at the EVCS. This is decided either by the time taken to reach the desired SOC input by the user or the maximum waiting time that has been input. Which ever is shorter.

One of the things that was noticed is that this approach represents two extremes. One where is the power delivery of the charger is maintained at its minimum rating (controlled) and one where it is maintained at its maximum rating (uncontrolled). It is seen that in the former case the burden on the grid is quite minimal but the waiting time for the EV users is very high. In the latter case it was observed that the waiting time for the users was very short but the impact on the power system was significant. With this in mind it was decided that controlled charging up to 25%, 50% and 75% of the charger's maximum rated power delivery would be allowed and the effects on both the power system and the users waiting time would be explored.

Fig. 5 (left) and (right) show the cases where the EMS can control the power delivery of the chargers and choose a value between 25%, 50% and 75% of the maximum rated power delivery of the charger as the minimum charging power. It can be noticed immediately that when the minimum charging power is set at 75 % of the maximum rated power the peaks and falls are frequent and high in magnitude but when it set to 25% of the maximum rated power the curve is smooth. Fig. 5 (left) shows the case for the load in the microgrid whereas Fig. 5 (right) shows the impact on the exchange of energy with the main grid. It can be seen that for all cases except during mid-time when the renewable energy production is high much of the excess demand due to the EVCS is fulfilled by importing power from the main grid.

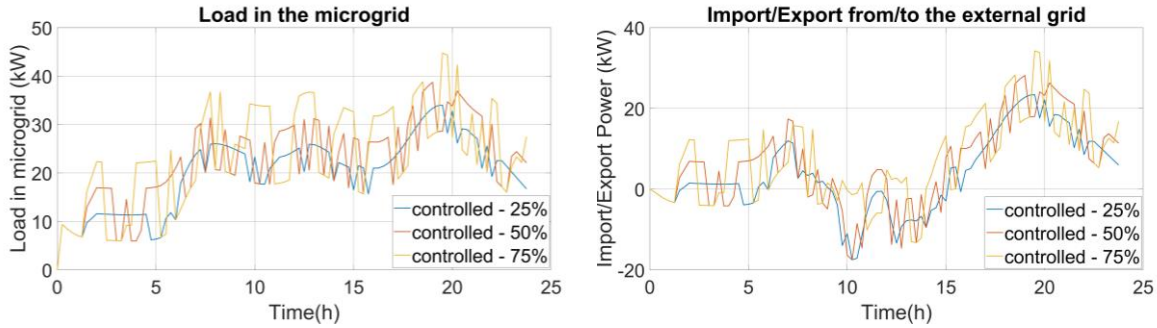

Fig. 5: Microgrid load characteristics (left) and power exchange with the main grid (right).

The average waiting time for all cases: controlled, uncontrolled, controlled – 25%, controlled – 50% and controlled – 75% are tabulated in table 2. It can be immediately noticed that with over 3 hours the charging waiting time is too high for the consumer when the charging is controlled fully. In the uncontrolled mode while the time is quite low Fig. 3 has shown the adverse effects it has on the power system. Hence, in order to find a balance between the interests of the consumer and the power system, the charging power of 50% of max. rated power delivery as the minimum of the charger would be chosen as the optimal setting for this study and further analysis is based on it.

Table 2. Charging scenarios' average waiting time.

Charging scenario	Average waiting time (minutes)
Charging rate – minimum (fully controlled)	208.30
Charging rate – limited up to 25% of max. rating	183.66
Charging rate – limited up to 50% of max. rating	74.90
Charging rate – limited up to 75% of max. rating	42.78
Charging rate – maximum (Uncontrolled)	31.27

Fig. 6 shows the SOC of the incoming vehicles at chargers 1 and 2 of the EVCS. In comparison to Fig. 4 it can be seen that a much higher number of EVs visit the chargers 1 and 2 throughout the day. The data from all 5 chargers are not shown since the figure would be difficult to interpret. It can be seen in this figure as usual that numerous EVs with different capacities, different waiting times, different initial SOCs and different desired SOC's come to the EVCS and charge until their individual stopping criteria is achieved.

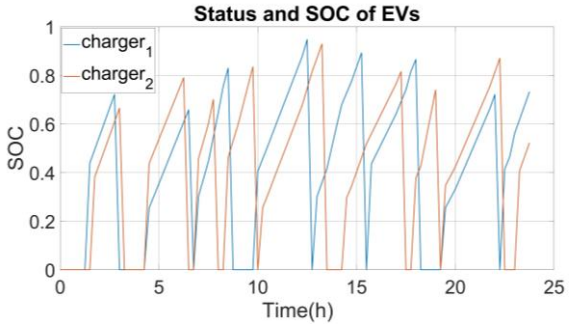

Fig. 6: Charging characteristics of incoming vehicles at chargers 1 and 2.

In order to ascertain the optimal location for the EVCS in the microgrid its effect on the power system has to be studied throughout the whole year. Since the computational burden involved in doing this is significant a clustering approach as adopted earlier has been used. The load data available for the whole year has been clustered as done previously using the K-means clustering and the ideal number of clusters was decided using the WCSS. Since the underlying data

remains the same as earlier the ideal number of clusters also remain the same at 3. Fig. 7 shows the 3 days representing each cluster which represent the 3 types of days that can exist in a year.

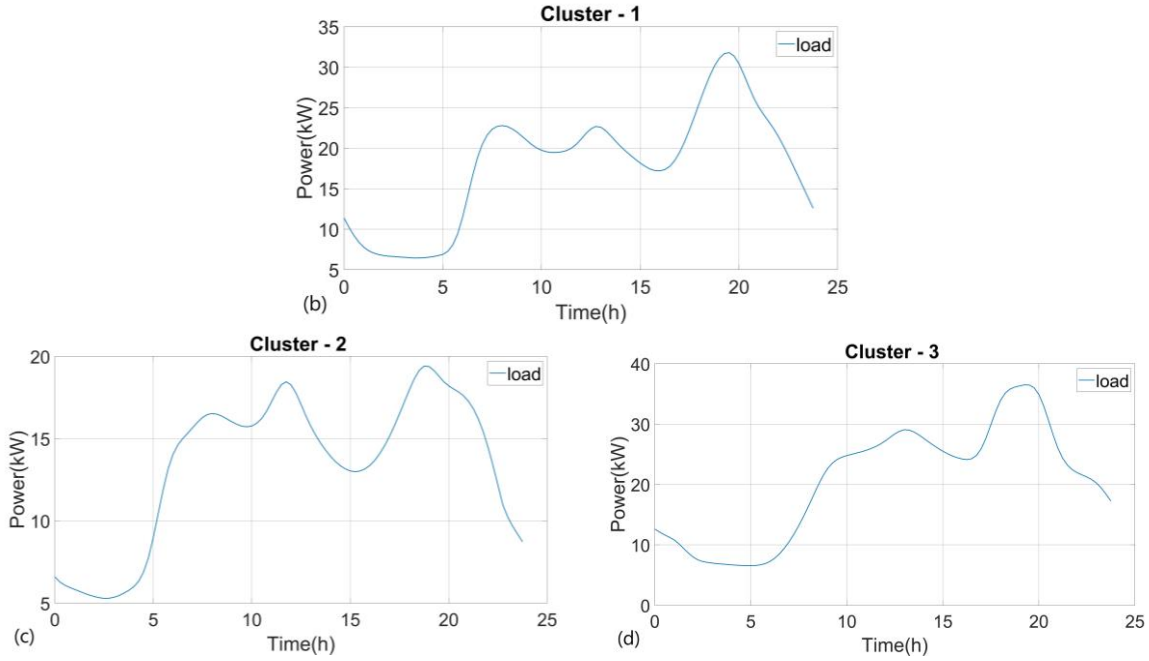

Fig. 7: (a) load cluster – 1 (b) load cluster – 2 (c) load cluster – 3

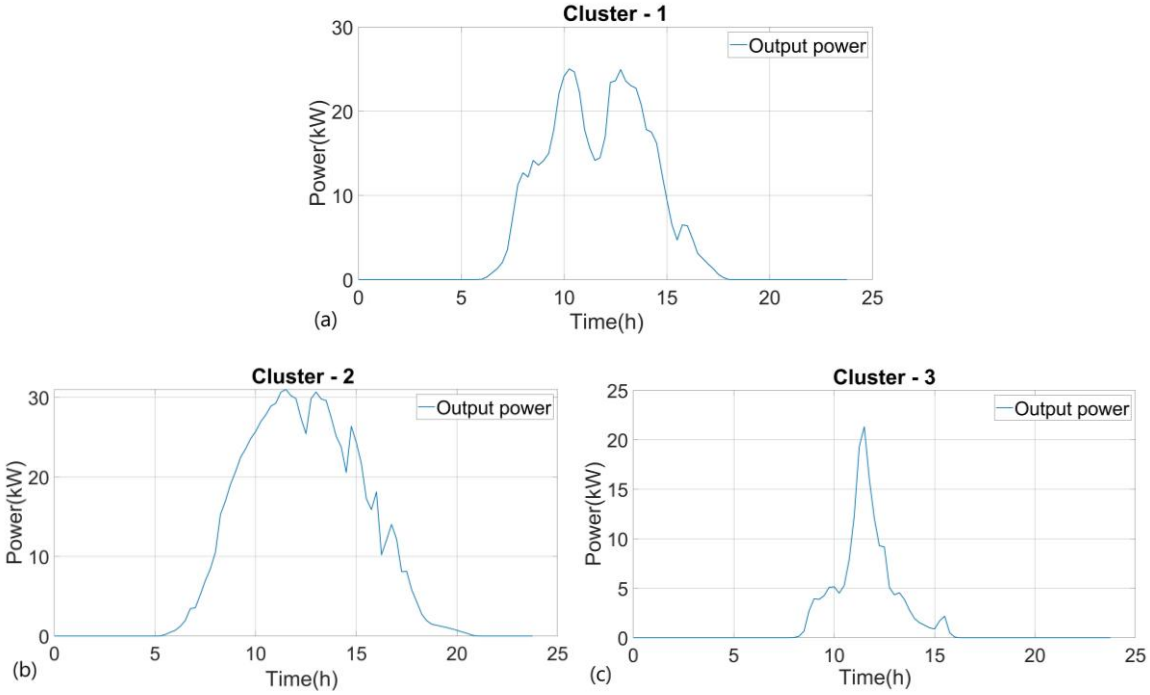

Fig. 8: Solar power output (a) cluster – 1 (b) cluster – 2 (c) cluster – 3

The day from cluster-1 is typical of the spring and autumn months when the load demand is between the minimum and maximum values observed during the year. Cluster-2 is more typical of summer when the load demand is usually lower and cluster – 3 is representative of the winter when the load demand usually peaks. The winter months are quite cold requiring the use of heaters whereas the summers are not typically harsh. This is the main reason for the load characteristic to behave the way it does.

Corresponding to the days representing the load throughout the year the solar power outputs from the same days were obtained. These days are shown in Fig. 8 and it can be noticed that Fig. 8 (a) is typical of the output observed from the solar panels during the spring and autumn months where it is not at the highest and not at the lowest values observed in a year. Fig. 8 (b) representing cluster 2 is typical of the summer where the solar panels are at their highest productivity whereas Fig. 8 (c) representing cluster 3 is typical of the winter where the solar power output remains quite low.

The total power imported from the main grid and the total line losses observed in the system when the EVCS is placed at different nodes is shown in Fig. 9. These figures are only for the cluster representing the summer months. The same metrics for days representing all clusters are tabulated in table 3 which will be used to derive wholesome conclusions about the optimal location of the EVCS.

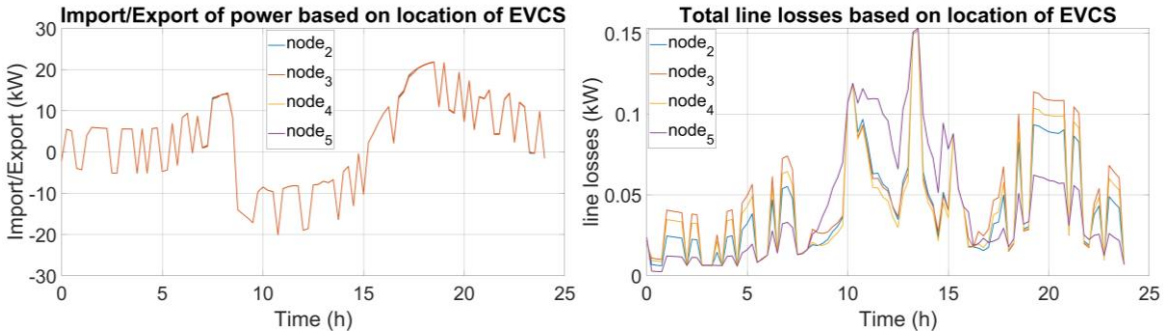

Fig. 9 Microgrid power exchange with the main grid (left) and total line losses (right) based on EVCS location.

In Fig. 9 (left) it can be noticed that the exchange of power with the main grid remains almost the same with regard to the specific day under consideration. There are minor differences amongst different curves in the figure which cannot be seen easily but when the overall power exchanges are summed up over the entire day the differences are visible. In Fig. 10 (right) it can be seen that more or less the total line losses in the microgrid remain the same irrespective of where the EVCS is placed but amongst all lines it can be seen that when placed at node 5 the overall line losses are the lowest. This is because node 5 is closest to the PCC of the microgrid and when the EVCS representing a massive load is placed at node 5 the overall line losses are slightly reduced.

Table 3 presents the total line losses and the net – import of power for when the EVCS is placed at different nodes for different days representing all clusters. It can be seen that the net-import of power remains the lowest when it is placed on node 5 for days representing spring, autumn and winter, also the export of power is highest during the summer months when it is placed at node 5.

Table 3. Total net - import of energy based on different location of the EVCS on different days.

Node	Cluster 1		Cluster 2		Cluster 3	
	Net-import (kWh)	Line losses Sum (kW)	Net-import (kWh)	Line losses Sum (kW)	Net-import (kWh)	Line losses Sum (kW)
2	63.35	4.68	-24.61	4.02	304.93	9.03
3	62.82	5.45	-25.00	4.63	303.37	10.13
4	62.69	4.96	-25.10	4.21	303.20	9.52
5	59.72	4.10	-26.06	4.14	300.58	7.06

The line losses follow a similar pattern where they are the lowest when placed on node 5 for days when power is imported from the main grid. This is because node 5 is closest to the PCC and the EVCS has a big load demand. Only for the summer month the line losses are lowest when placed in node 2. In conclusion looking at the performance of these two metrics across different days representing different parts of the year it can be said that node 5 is the ideal position to place the EVCS. It should be remembered that these conclusions are made based on results obtained from cluster analysis. In all clusters there are days that are exceptions and outliers and differ from typical days representing that cluster where the results may not hold the same conclusions but having said that the above conclusions would be true for most days in a year.

Conclusions

This chapter presented an EVCS model consisting of 5 tesla level 2 chargers. Numerous charging scenarios were explored where the minimum setting of the chargers was varied, and the optimal power delivery of the charger was determined using 'fmincon' of MATLAB. The EVCS was integrated in microgrid consisting of renewable energy sources and DGs. The objective function considered in this chapter is different from the economic objective considered in the previous chapter. The power imported from the main grid was minimized in this case since it is in alignment with the objective of the faculty to eventually establish a stand-alone microgrid. It was shown that the optimum setting for the charger was in the range between its maximum and minimum when it is set to 50% of its maximum rated power delivery. This setting was in favor of both the EV users and the electrical grid. This was followed by in-depth analysis where again clustering was performed using k-means algorithm to obtain clusters representing the whole year. This resulted in 3 clusters. A representative day for each cluster was selected and then the EVCS was cycled across all nodes of the microgrid for every one of the days during which the total imported energy and line losses of the system were calculated based on which it was decided that node 5 was the optimal location for the EVCS. For modelling the stochastic behavior of EVs, their capacities, initial SOCs, maximum waiting time, desired SOCs and time of arrivals were modelled as gaussian functions in order to keep them as realistic as possible.

Summary

The thesis provides research concerning 2 crucial parts of an energy management system for microgrids which are forecasting and optimization.

With regard to forecasting it explores novel deep learning architectures for 10-min ahead and 1-hour ahead forecasts. The models considered include the CNN, multi – headed CNN, CNN – LSTM and LSTM autoencoders. The results of all the approaches made were compared with each other and an ARIMA model for the specific data set. It was concluded that the LSTM autoencoder provided the most accurate and quickest forecasts.

During this process numerous data processing techniques were incorporated, and a sliding window approach was utilized to feed data into the deep learning algorithms. Even during the time, the current thesis is being written, new deep learning models that are faster and more efficient are being introduced which could potentially pave way for improved solar forecasting models for EMS applications and others. Despite the accuracy being satisfactory with the current models, an increase in the horizon of forecasting the leads to deep learning models becoming less accurate. Hence, there is an imperative to explore appropriate models for the higher time horizons. There are also other data processing techniques to be considered such as clustering, principal component analysis and frequency-based transforms of the input data. All of which could be considered as a part of the future investigation.

With regard to optimization, the gamut of approaches and algorithms is endless. This thesis focuses particularly on the meta-heuristic optimization approaches to OPF. It started with the description of power flow, numerical solvers for power flow and considered both meshed and radial networks for obtaining the steady state solution of the network. This was followed by application of traditional meta-heuristics for optimization such as the GA, PSO and MIDACO. Following this was the application of 2 novel optimization algorithms that were introduced in 2020 such as the PO and LA.

All the optimization algorithms were integrated with MATPOWER for improved performance. The results were compared with one another and conventional non heuristic optimization approaches. It was seen that the most balanced and effective algorithm was MIDACO. The conclusions were made based on the value of the final solution and the speed of convergence to a solution. The area of optimization is also being developed rapidly which results in new approaches and algorithms, the future work in this regard would be continuous exploration and application of the incoming optimization approaches to OPF and then subsequent application to ED.

With regard to ED and their application to microgrids, their scheduling and energy management MIDACO with MATPOWER was chosen as the optimization algorithm. A comprehensive modelling of the components of the microgrid was made and the LCOE was calculated for every generator in the microgrid considering their capital costs, costs of installation, fuel costs and operational and maintenance costs. The generators included were the solar PV panels, BSS, FC + HST, DGs and micro hydro plants. It was seen that the LCOE of the micro-hydro was the lowest and that of the FC + HST was the highest. While importing and exporting power to the main grid actual grid prices were used. In order to make a thorough analysis the entire year was considered by clustering the load data. This resulted in 3 days representing the different types of days encountered annually. It was seen that the economic benefits were at maximum during the summer and minimum during the winter. This was directly correlated with the fact that during the summer the power exported to the grid is high

whereas power imported from the grid is high during winter. The future work in this section is concerned with updating the LCOE of the generators used in the microgrid since the costs involved keeps reducing especially with solar PV, BSS and FC + HST. Incorporate other generating sources and also, more complex microgrid networks need to be explored and multi-objective functions for optimization can be considered which not only reduce the costs but also other objectives such as emissions and voltage profile improvements amongst others.

In the final chapter an EVCS model was introduced consisting of 5 Tesla level 2 chargers. This chapter was introduced in order to study the effects of introducing an EVCS in the university campus. Several charging scenarios were introduced where the optimal power delivery was determined by 'fmincon' of MATLAB within the EMS of the microgrid. In this case the objective function was modified and the import of the power from the main grid was minimized. This was chosen since the university plans to convert the existing facility into a standalone one in the future and wanted to study the dynamics of the system by utilising internal generating capacities to their maximum. Different charging scenarios were considered where the minimum charging rate was set at 25%, 50% and 75% of the maximum rated power delivery of the charger. These scenarios were considered so that a balance between consumer interest and the power system can be obtained.

From the results it was concluded that the range of power delivery between 50% of its maximum rated power delivery and maximum rated power delivery was optimal. Using this range, an analysis was performed where the entire year was clustered and days representing the whole year were chosen. In every one of these days the EVCS was placed in every node of the microgrid and the most optimal location of the same was chosen by looking at overall system loses and the energy imported from the main grid. For the microgrid layout used in chapter 5 this was node 5. Future work in this regard would be the use of EVCS as both a generator and a load. Currently, in this study it only represents a load. Also, important would be obtaining the data of city traffic and study the behavioural patters of EVs in order to create an optimal plan for the EVCS. Since this data was unavailable much of the randomness involved in EV arrival, initial SOCs, desired SOCs, EV capacities and times of arrival were determined using gaussian functions.

In general, the area of power system research is fast evolving especially considering renewable energy sources, microgrids and electrical vehicles. The once predictable and consistent power network is now becoming stochastic and much more complex. This has created the need for improved forecasting, optimization and control activities into managing the power system. This in turn has resulted in massive research into this domain.

Appendix

A1 Control variables and their limits

Control variables	Min	Max
P1 (MW)	50	200
P2 (MW)	20	80
Ps (MW)	15	50
P8 (MW)	10	35
P11 (MW)	10	30
P13(MW)	12	40
V1 (p.u.)	0.95	1.1
V2 (p.u.)	0.95	1.1
Vs (p.u.)	0.95	1.1
V8 (p.u.)	0.95	1.1
V11 (p.u.)	0.95	1.1
V13 (p.u.)	0.95	1.1
T11	0.90	1.1
T12	0.90	1.1
T15	0.90	1.1
T36	0.90	1.1
Qc10 (MVAr)	0.00	5.0
QC12 (MVAr)	0.00	5.0
Qc15 (MVAr)	0.00	5.0
QC17 (MVAr)	0.00	5.0
Qc20 (MVAr)	0.00	5.0
QC21 (MVAr)	0.00	5.0
Qc23 (MVAr)	0.00	5.0
Qc24 (MVAr)	0.00	5.0
Qc29 (MVAr)	0.00	5.0

A2 Generator cost coefficients

Bus no.	Cost coefficients		
	α	β	γ
1	0.00375	2.00	0.00
2	0.01750	1.75	0.00
5	0.06250	1.00	0.00
8	0.00834	3.25	0.00
11	0.02500	3.00	0.00
13	0.02500	3.00	0.00

A4 Branch data

Line no	From bus	To bus	R (p.u.)	X (p.u.)	B (p.u.)	Tap settings	Line no	From bus	To bus	R (p.u.)	X (p.u.)	B (p.u.)	Tap settings
1	1	2	0.0192	0.0575	0.0264	-	22	15	18	0.1070	0.2185	0.0000	-
2	1	3	0.0452	0.1852	0.0204	-	23	18	19	0.0639	0.1292	0.0000	-
3	2	4	0.0570	0.1737	0.0184	-	24	19	20	0.0340	0.0680	0.0000	-
4	3	4	0.0132	0.0379	0.0042	-	25	10	20	0.0936	0.2090	0.0000	-
5	2	5	0.0472	0.1983	0.0209	-	26	10	17	0.0324	0.0845	0.0000	-
6	2	6	0.0581	0.1763	0.0187	-	27	10	21	0.0348	0.0749	0.0000	-
7	4	6	0.0119	0.0414	0.0045	-	28	10	22	0.0727	0.1499	0.0000	-
8	5	7	0.0460	0.1160	0.0102	-	29	21	22	0.0116	0.0236	0.0000	-

9	6	7	0.0267	0.0820	0.0085	-	30	15	23	0.1000	0.2020	0.0000	-
10	6	8	0.0120	0.0420	0.0045	-	31	22	24	0.1150	0.1790	0.0000	-
11	6	9	0.0000	0.2080	0.0000	1.078	32	23	24	0.1320	0.2700	0.0000	-
12	6	10	0.0000	0.5560	0.0000	1.069	33	24	25	0.1885	0.3292	0.0000	-
13	9	11	0.0000	0.2080	0.0000	-	34	25	26	0.2544	0.3800	0.0000	-
14	9	10	0.0000	0.1100	0.0000	-	35	25	27	0.1093	0.2087	0.0000	-
15	4	12	0.0000	0.2560	0.0000	1.032	36	28	27	0.0000	0.3960	0.0000	1.068
16	12	13	0.0000	0.1400	0.0000	-	37	27	29	0.2198	0.4153	0.0000	-
17	12	14	0.1231	0.2559	0.0000	-	38	27	30	0.3202	0.6027	0.0000	-
18	12	15	0.0662	0.1304	0.0000	-	39	29	30	0.2399	0.4533	0.0000	-
19	12	16	00945	0.1987	0.0000	-	40	8	28	0.0636	0.2000	0.0214	-
20	14	15	0.2210	0.1997	0.0000	-	41	6	28	0.0169	0.0599	0.0065	-
21	16	17	0.0824	0.1932	0.0000	-							

A4 Load values (Active and Reactive power demand)

Bus	Active Power (MW)	Reactive Power (MVAr)	Bus	Active Power (MW)	Reactive Power (MVAr)
1	0.00	0.00	16	3.50	1.80
2	21.70	12.70	17	9.00	5.80
3	2.40	1.20	18	3.20	0.90
4	7.60	1.60	19	9.50	3.40
5	94.20	19.00	20	2.20	0.70
6	0.00	0.00	21	17.50	11.20
7	22.80	10.9	22	0.00	0.00
8	30.00	30.00	23	3.20	1.60
9	0.00	0.00	24	8.70	6.70
10	5.80	2.00	25	0.00	0.00
11	0.00	0.00	26	3.50	2.30
12	11.20	7.50	27	0.00	0.00
13	0.00	0.00	28	0.00	0.00
14	6.20	1.60	29	2.40	0.90
15	8.20	2.50	30	10.6	1.90