

**Machine Learning to Improve Microgrid Energy Management**

**ME420 Mechanical Engineering Individual Research Project**

A project report submitted to the

Department of Mechanical Engineering,

Faculty of Engineering,

University of Peradeniya,

Sri Lanka,

in partial fulfilment of the requirements for the degree

Bachelor of the Science of Engineering,

by

M.G.I.U. Karunarathne (E/17/156)

Peradeniya, 2023

**Machine Learning to Improve Microgrid Energy Management**

by

M.G.I.U. Karunarathne

Supervised by

Dr. Damayanthi Herath

Dr, Ubaya Higgoda

Dr. Lahiru Jayasooriya

# Declaration

I confirm that I have written this report without any external help and not using sources other than those I have listed in the report. I confirm also that this report or similar version of it has not been submitted to any other examination board and has not been previously accepted as part of an exam for a qualification. Each direct quotation or paraphrase of an author is clearly referenced.

Place, 20th Dec, 2023 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

M.G.I.U. Karunarathne

Name of the project supervisor/s:

Dr. Damayanthi Herath

# Summary

Microgrids are localized energy systems that operate independently or in conjunction with the main power grid, integrating various energy sources for reliable and resilient power supply. They can disconnect during outages, ensuring uninterrupted power to critical facilities and promoting sustainability through renewable energy integration. Microgrid energy demand refers to the electrical power needed to meet consumer needs, influenced by factors like time, weather, and habits. Accurate forecasting of energy demand is crucial for efficient microgrid operation, resource management, and optimization. Machine learning offers advantages over traditional methods in forecasting, handling complex relationships, adapting to changes, and providing scalability. This report aims to forecast microgrid energy demand using machine learning, evaluate model performance, and develop optimization technologies for enhanced microgrid operation. The project seeks to advance the application of machine learning in microgrid management.

# Acknowledgement

I'd like to express my gratitude to my excellent supervisors, Dr. Damayanthi Herath and Dr. Ubaya Higgoda, for their valuable supervision, support, and direction over the course of this research project. My gratitude goes to the Faculty of Engineering at the University of Peradeniya for providing me with the opportunity to study at the Department of Mechanical Engineering and for guiding me through this research project.

Furthermore, I would like to thank all of the lecturers, for their invaluable assistance in defining my research methodology and critiquing my results. Specially I would also want to thank Dr. Lahiru Jayasooriya and Dr. Hansani Weeratunge for their valuable feedbacks, mentorship and guidance during the research project.

I'd also like to thank my colleagues and friends for all of their assistance with countless opportunities during this research endeavor. My thanks also go to my family and friends for their encouragement and support throughout my studies.

# Table of contents

[Declaration ii](#_Toc153967285)

[Summary iii](#_Toc153967286)

[Acknowledgement iv](#_Toc153967287)

[Table of contents v](#_Toc153967288)

[List of Figures vii](#_Toc153967289)

[List of Tables viii](#_Toc153967290)

[List of Symbols ix](#_Toc153967291)

[1 Introduction 1](#_Toc153967292)

[1.1 Background of the problem 1](#_Toc153967293)

[1.2 Aim of the project 2](#_Toc153967294)

[1.3 Objectives 2](#_Toc153967295)

[1.4 Proposed methodology 2](#_Toc153967296)

[2 State of the art 4](#_Toc153967297)

[2.1 Introduction 4](#_Toc153967298)

[2.2 What is microgrid 4](#_Toc153967299)

[2.3 Energy demand 5](#_Toc153967300)

[2.4 Forecasting energy demand 5](#_Toc153967301)

[2.5 Machine learning 6](#_Toc153967302)

[2.6 Machine learning models used 7](#_Toc153967303)

[2.7 Optimization of microgrid operation 8](#_Toc153967304)

[3 Methodology 10](#_Toc153967305)

[3.1 Data Collecting and Data Preprocessing 10](#_Toc153967306)

[3.1.1 Demand charge related data set 10](#_Toc153967307)

[3.1.2 Data understanding 12](#_Toc153967308)

[3.1.3 Data preprocessing 18](#_Toc153967309)

[3.2 Implementing CNN model 22](#_Toc153967310)

[3.2.1 1-D CNN model 22](#_Toc153967311)

[3.2.2 Model architecture 23](#_Toc153967312)

[3.3 Model evaluation 29](#_Toc153967313)

[3.4 Deploying forecasted results in optimization 30](#_Toc153967314)

[4 Optimization 32](#_Toc153967315)

[4.1 Why optimizing microgrid 32](#_Toc153967316)

[4.2 Operating process of microgrids 33](#_Toc153967317)

[4.3 Algorithms and how to use them 36](#_Toc153967318)

[4.4 Approach of forecasted data on optimization 37](#_Toc153967319)

[5 Results 39](#_Toc153967320)

[5.1 Output visualization 39](#_Toc153967321)

[5.2 Performance evaluation 42](#_Toc153967322)

[6 Conclusion 48](#_Toc153967323)

[References 50](#_Toc153967324)

[Appendix 52](#_Toc153967325)

# List of Figures

[Figure 3.1: Data set of demand charge data 11](#_Toc153967471)

[Figure 3.2: Overview of the data set 12](#_Toc153967472)

[Figure 3.3: Histogram plot for Total Campus Load 13](#_Toc153967473)

[Figure 3.4: Histogram plot for On Campus Generation 13](#_Toc153967474)

[Figure 3.5: Histogram plot for SDG&E Import 14](#_Toc153967475)

[Figure 3.6: Histogram plot for Adjusted Demand 14](#_Toc153967476)

[Figure 3.7: Heat map for the data set 15](#_Toc153967477)

[Figure 3.8: Time series plot for Total campus load 16](#_Toc153967478)

[Figure 3.9: Time series plot for on campus generation 16](#_Toc153967479)

[Figure 3.10: Time series plot for SDG&E import 16](#_Toc153967480)

[Figure 3.11: Time series plot for Adjusted demand 17](#_Toc153967481)

[Figure 3.12: Time series plot for 20 days 17](#_Toc153967482)

[Figure 3.13: Box plot for the features 18](#_Toc153967483)

[Figure 3.14: Statistic of the data set 18](#_Toc153967484)

[Figure 3.15: Violine Plot for the training data set 21](#_Toc153967485)

[Figure 3.16: Sample plot for Total campus load 24](#_Toc153967486)

[Figure 4.1 : operating process A 33](#_Toc153967487)

[Figure 4.2: Operating process B 33](#_Toc153967488)

[Figure 4.3 : Operating process C 34](#_Toc153967489)

[Figure 4.4 : Operating process D 34](#_Toc153967490)

[Figure 4.5: Optimization algorithm classifications 36](#_Toc153967491)

[Figure 5.1: Plot for baseline model-Repeat for window size 39](#_Toc153967492)

[Figure 5.2: Plot for “CNN” model for window size 40](#_Toc153967493)

[Figure 5.3: Plot for “1D CNN” model for window size 41](#_Toc153967494)

[Figure 5.4: Actual vs predicted data of test data set for a week 41](#_Toc153967495)

[Figure 5.5: Actual vs predicted for test data set 42](#_Toc153967496)

[Figure 5.6: Training and validation loss over epochs for “CNN” model 43](#_Toc153967497)

[Figure 5.7: Training and validation loss over epochs for “1D CNN” model 43](#_Toc153967498)

[Figure 5.8: Training and validation accuracy over epochs for “CNN” model 44](#_Toc153967499)

[Figure 5.9: Training and validation accuracy over epochs for “1D CNN” model 44](#_Toc153967500)

[Figure 5.10: MSE value (Average over all times and outputs) 45](#_Toc153967501)

[Figure 5.11: MAE values (Average over all times and outputs) 46](#_Toc153967502)

# List of Tables

[Table 4.1:Components in Operating process of microgrid 35](#_Toc153782978)

[Table 4.2: Operating process description 35](#_Toc153782979)

# List of Symbols

Abbreviations

|  |  |
| --- | --- |
| CNN | Convolutional neural network |
| ML | Machine learning |
| DERs | Distributed energy resources |
| ARIMA | Autoregressive integrated moving average |
| ANN | Artificial neural network |
| DL | Deep learning |
| SVM | Support Vector Machines |
| kNN | k-Nearest Neighbors |
| RNN | Recurrent neural network |
| LSTM | Long short-term memory networks |
| BLSTM | Bidirectional long short-term memory networks |
| DNN | Deep neural network |
| MLP | Multilayer Perceptron |
| EV | Electrical vehicle |
| PV | Photovoltaics |
| BESS | Battery energy storage system |
| SDG&E | San Diego Gas & Electric |
| EDA | Exploratory Data Analysis |
| HTML | Hypertext Markup Language |
| DF | Data frame |
| NaN | Not a number |
| 1D - CNN | One dimensional convolutional neural network |
| MSE | Mean squared error |
| MAE | Mean absolute error |
| MILP | Mixed-Integer Linear Programming |
| RES | Renewable energy sources |
|  |  |
|  |  |

# Introduction

## Background of the problem

Microgrids are localized energy systems that integrate various energy sources and operate independently or in conjunction with the main power grid. They provide reliable and resilient power supply to specific areas, incorporating renewable energy technologies, energy storage systems, and conventional generators. Microgrids can disconnect from the main grid during outages and emergencies, ensuring uninterrupted power to critical facilities. They promote sustainability by integrating renewable energy, reducing carbon emissions, and optimizing energy generation and consumption.

When considering Microgrid energy demand it refers to refers to the amount of electrical power required by a microgrid to meet the energy needs of its consumers within a specific time period. It represents the total electricity consumption of the connected loads, including residential, commercial, or industrial facilities related to microgrid. Also, the demand for energy from microgrids might change depending on things like the time of day, the day of the week, seasonal fluctuations, weather, and consumer habits.

If we can accurately forecast the energy demand of microgrid it will be more effective for efficient operation of microgrid, when managing resource and in optimizing microgrid system. Also, it will be helpful in ensuring reliable power supply, manage load balancing inside the microgrid and optimize the generating and storage resources. So, understanding and predicting energy demand will help in using resources wisely, choosing demand response tactics and in optimizing energy usage and minimize cost.

In forecasting energy demand machine learning have advantage than traditional methods as they rely on pre-defined rules and assumptions. Machine learning has ability to handle complex relationships, adapt to changing conditions and leverage diverse data source. Also, it offers data-driven approach and scalability for accurate prediction. Machine learning models continuously learn and improve over time, ensuring accurate and reliable forecasts in dynamic microgrid environments. So, machine learning is providing more advance and effective approach to forecasting energy demand in microgrids.

This report aim is to forecast energy demand of microgrid and use optimization technologies to optimize microgrid operation. The project objectives include developing forecast models using machine learning techniques, evaluating the performance of these models on microgrid data, and developing a simple optimization model for microgrid operation according to that data. By achieving these objectives, the project seeks to enhance the understanding and application of machine learning in microgrids.

## Aim of the project

The aim of the project is to forecast energy demand using machine learning techniques and optimize microgrid operation. We aim to improve accuracy of energy demand forecasts and improve the performance and efficiency of microgrid.

## Objectives

1. Implement a machine learning-based tool for forecasting microgrid energy demand. This will involve training ML models on historical data to predict future energy demand.
2. Identify the most effective algorithm for forecasting energy demand in the microgrid. In this most effective ML algorithm will be find out using trained models and outputs.
3. Deploying machine learning algorithms in optimizing the energy generation of microgrids.

## Proposed methodology

Proposed methodology to forecast energy demand using machine learning is explained briefly as following. Here it is divided mainly to four parts and their basic tasks are explained in point form, this will be discussed in detail in following chapters.

* **Data Collection and Data Preprocessing:**

Select a relevant dataset that includes historical energy demand.

Using the proper techniques, prepare the data to manage any missing or incorrect data. Techniques like data cleansing, standardization, and imputation may be used in this.

* **Forecast Model Selection and Model Training:**

Using historical data, choose a suitable prediction model or algorithm to forecast the demand for energy.

Utilize the historical dataset to train the chosen model and discover the underlying trends and connection between the input variables and energy demand

* **Model Evaluation**

Analyze the trained forecast model's performance and accuracy. This entails evaluating metrics like mean absolute error, root mean square error, and correlation coefficients while contrasting the expected values with the actual data.

To ensure reliability and generalizability of forecast model it will be tested on unseen data.

* **Optimization and Integration:**

Develop a suitable simple optimization model for microgrid operation. This model will consider various factors such as operating cost, emission and other technical constraints to determine the optimal energy demand.

Apply the forecasted data to optimize the microgrid operation. By incorporating the forecasted energy demand, the optimization model will dynamically adjust the energy supply and demand to ensure efficient and cost-effective operation.

# State of the art

## Introduction

In past few years carbon emission and energy demand have been increased. This happens due to increase in energy-consuming equipment and population. So, due to growing concerns regarding the effects of fossil fuel emissions, the importance of renewable energy resources has grown markedly in recent years.[1] . Also, the power management is essential in industrial and local consumers because it is crucial for energy conservation, cost savings, environmental protection, technological advancements, and overall user satisfaction.

The industry standard for power management is a centralized power grid. Which is largely dependent on fossil fuels. There are two major problems in Centralized power grids. First, the rigid, inflexible centralized grid is unable to accommodate the unpredictable nature of current distributed energy resources (DERs). Second, energy is often lost when travelling large distances between energy generation and consumption locations [2] ,[1]. To overcome these problems Distributed, or on-site generation, has been proposed as a next generation smart grid solution. This method has advantages than other methods as its ability to generate energy locally, also it can reduce the energy lost in transmission, and due to its small scale and isolation it has superior reliability.[1] [3] .So, under these circumstances, small-scale grids operating in small areas as fully functioning energy systems have become an interesting solution.[4]. which is known as microgrids. Microgrids promote sustainability by integrating renewable energy, reducing carbon emissions, and optimizing energy generation and consumption. In following sections look more about microgrids in details.

## What is microgrid

Microgrids are localized energy systems that integrate various energy sources. A microgrid is a small-scale grid. However, it is fully functional, operating in a limited geographical area; it can operate independently or be connected to a larger grid [4]. Microgrids are mostly used in remote or off grid areas where it is difficult to connect to main grid. Now a days with the increasing of carbon emission microgrids have been as a solution to that. So now there are microgrids where they can operate independently or connected to main grid. Microgrids are using distributed energy resources (DERs) where it includes renewable energy sources. Such as wind turbines, solar panels, combine heat and power plants and energy storage systems like batteries, these energy resources can be categorized as adjustable or non-adjustable and storage systems The microgrid is the integration of several renewable energy sources with adjustable or nonadjustable loads and storage systems [5]. In the previous 10 years, a lot of research has come out on microgrids as a potential source of energy in the near future [6]. As mentioned in above this localized on-site generation method, is implemented due to their improved reliability and the ease of inclusion of renewable energy generation

## Energy demand

Energy strategy is extremely important for developing countries. The energy demand of a microgrid can vary depending on a number of factors, including the size of the microgrid, the type of loads it serves, and the availability of renewable energy sources. In general, microgrids are designed to meet the peak load demand when they are connected to loads.

This peak load demand is the maximum electricity that a microgrid must deliver at any one moment. A fundamental microgrid system’s load demand often fluctuates hourly [6]. So, this demand fluctuates depending on the time of day, the season, and the weather. For example, peak load demand of microgrid is higher in a hot climate. Energy demand is important in determining microgrids generating cost. The two most important factors that determine a microgrid system’s generating cost are load demand and energy market pricing [6].

So, to ensure that a microgrid can meet its peak load demand, it is important to accurately forecast the demand. This can be done by using historical data, as well as weather forecasts and other factors

## Forecasting energy demand

Energy demand forecasting is the process of predicting future energy consumption based on historical data and other factors. Energy demand forecasting is one of the most important tools that decision makers use (Ediger and Akar, 2007) [7]. As it can help to ensure that there is enough available to meet demand, and that energy resources are used efficiently. There are different methods that can be used to forecast energy demand. Techniques used in energy demand forecasting studies are mainly composed of BoxJenkins models, regression models, econometric models and neural networks (Jebaraj and Iniyan, 2006) [7].

In the studies of forecasting electric energy demand, there are many methods used in the literature including autoregressive integrated moving average (ARIMA), artificial neural networks ANN, time series methods, support vector machines and fuzzy logic method [7]. In time series analysis it uses historical data to identify trends and patterns in energy consumption. These trends can then be used to forecast future demand.

In 1960 Turkey’s Planning Organization realized a demand forecast study based on simple regression. Then they continue studies on energy demand. These early forecasts consistently predicted much higher values than the consumptions that actually realized. also, Short-term energy supply and demand forecasting are necessary to make informed and reliable decisions for distributed energy systems [8]. However, although there are many forecasting methods which take into account the advances in information, metering and control technologies in order to address the challenges of forecasting problems [9] ,[10].

Here are some challenges of energy demand forecasting. Energy demand is often volatile and difficult to predict. This is due to a number of factors, such as weather, economic conditions, and technological change. There is a lack of reliable data. This is especially true for long-term forecasting. Also, the models used for forecasting are often complex and difficult to understand. This can make it difficult to interpret the results of the forecast. Energy demand forecasting is a crucial instrument that may assist to increase the effectiveness and sustainability of energy systems despite these challenges. As the world becomes more interconnected and energy markets become more volatile, the need for accurate and reliable energy demand forecasts will only increase. Machine learning (ML) and deep learning (DL) based models are promising solutions for predicting consumers’ demands and energy generations from RESs [11]. let’s look forward to ML and ML models used in forecasting energy demand.

## Machine learning

Machine learning (ML) is a type of artificial intelligence (AI) that allows computers to learn without being explicitly programmed. In ML algorithms that use available data for the training of a model, where the underlying system is more or less treated as a black box, are usually grouped under the term machine learning. Arthur Samuel was the one who coined the term ‘’Machine Learning’’ in 1959, defining it as “the ability to learn without being explicitly programmed.” Machine Learning, at its most basic form, is the practice of using algorithms to parse data, learn from them, and then make a determination or prediction about something in the world [9].

Machine learning tasks may be classified into three broad categories, namely supervised, unsupervised and semi-supervised. Another categorization of machine learning tasks is by the desired output. If the output of the model is a class, then it is a classification problem, if it is a number then it is a regression problem and if it is a set of input groups, it’s a clustering problem [9].

Machine learning techniques have been proven useful for short-term electricity load forecasting. Especially in microgrids where large variety of data should be included in the energy consumption prognosis. These Novel techniques and models are taking advantage of the advances in artificial intelligence algorithms. Where they allow for faster convergence, manipulating big data sets and solving more complex problems [9].

There are some challenges of using ML for energy demand forecasting. Such as it requires a large amount of data, it can be computationally expensive to train ML models and the results of ML models can be difficult to interpret. However, ML is a potential approach for predicting energy demand. As the technology continues to develop, it is likely to become even more effective for forecasting energy demand.

## Machine learning models used

Machine Learning has been shown to be effective for forecasting energy demand. According to researches it shows that ML was able to produce more accurate forecasts than traditional methods. As Electrical load forecasting algorithms are needed for prediction of the energy demand for the day ahead, a few weeks up to a year or a period of over a year [9]. so, the methods that are using have to be more accurate and efficient. The choice of ML model will depend on the specific needs of the energy planner or decision-maker. For example, if the planner is interested in forecasting short-term demand, then a SVR algorithm may be the best option. However, if the planner is interested in forecasting long-term demand, then an ANN may be more appropriate

In literature it has used many ML methods for forecasting energy demand. Enea Mele (2019) have given a review about short-term forecasting such as Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Random Forest and Artificial Neural Networks (ANN) and compare their performance efficiency, capabilities and limitations.[9].

In [1] it shows that there have been proposed Multiple deep learning techniques in past. These include artificial neural networks (ANN), convolutional neural networks (CNN), recurrent neural networks (RNN), Long short-term memory networks (LSTM) and bidirectional long short-term memory networks (BLSTM).

Maciej Slowik and Wieslaw Urban has developed a universal forecasting tool for energy consumption by end-use consumers. This model allows the end-users to be equipped with an energy demand prediction, enabling them to participate more effectively in the smart grid energy market. A single, long short-term memory (LSTM)-layer-based artificial neural network model for short-term energy demand prediction was developed.[4].

Among these methods deep learning Deep Learning models are a good alternative to learn patterns from customer data and then forecast demand for different forecasting horizons. deep Learning uses multiple layers of neurons composed of complex structures to model high-level data abstractions [12]. most commonly used deep learning-based methods for energy management and power forecasting, namely, artificial neural networks (ANN), deep neural networks (DNN), convolutional neural networks (CNN), and recurrent neural networks (RNN) [13].

Another paper [14] considers a load forecasting problem in residential areas as well as in commercial buildings. A deep RNN is employed for medium to long term energy consumption forecasting. Simulation results show the effectiveness of the proposed deep RNN based model over MLP for load demand prediction of commercial buildings.

In research work [15] also adapts DL based methods for load forecasting. In there a hybrid forecasting method is developed by combining the best features of CNN and K-means clustering. The results show that their hybrid CNN–K-means forecasting algorithm has higher accuracy.

So, there are more research in forecasting energy demand of microgrid. And this existing literature suggests that machine learning techniques have significant potential to improve the accuracy of energy demand forecasting [10], [16]. Further research is needed to develop and evaluate advanced machine learning algorithms for energy demand forecasting, and to investigate their potential applications in microgrid management and optimization.

## Optimization of microgrid operation

Microgrids typically include a mix of renewable energy sources, such as solar and wind power, and conventional generation sources. The effective management and optimization of microgrid operation is essential for ensuring reliable and cost-effective power supply [17].

There are some different optimization techniques that can be used to optimize microgrid operations. Such as Linear programming which can be used to optimize the operation of microgrids by minimizing the cost of electricity generation or by maximizing the reliability of the system. Also mixed integer programming can be used in microgrid operations by considering the discrete nature of some of the decisions, and heuristic optimization can be used find quick solutions that are close to optimal.

By optimizing microgrid operations we can reduce the cost. Where forecasting data can be used in here. Also using that data, it can used to improve the reliability of the microgrid. This can be done by ensuring that there is always enough generation capacity to meet demand and by ensuring that the system is resilient to disruptions. There are some challenges that could occur when optimizing microgrid operations. The optimization problems that arise in microgrid operation can be complex. This is because the system is typically nonlinear and has a large number of variables. Also, the operation of a microgrid is subject to uncertainty, such as the amount of renewable energy that is available and the demand for electricity. This uncertainty can make it difficult to optimize the system. And the availability of data can be a challenge for microgrid optimization. This is because microgrids are relatively new and there is not a lot of historical data available. However, optimizing microgrid operation is a valuable tool that can help to improve the efficiency, reliability, and sustainability of these systems.

# Methodology

In methodology it focusses on major three parts they are data collecting and preprocessing, implementing CNN model then optimize the implemented model such as hyperparameter tuning and third part is deploying forecasted results in optimization of microgrids. Based on these three parts methodologies is explained bellow.

## Data Collecting and Data Preprocessing

Data set was selected from open-source available data set as Open-source multi-year power generation, consumption, and storage data in a microgrid.[18] In this data set it includes comprehensive power dataset of part of the University of California, San Diego microgrid. This microgrid contains several distributed energy resources (DERs). Such as solar power plant, electric vehicles, electric and thermal storage. In this data set most of them contain 15-min average of real and reactive power. Overall, UC San Diego self-generate about 85% of its electricity consumptions and import the remaining 15% from the local utilities.[18] to record the data they have placed meters on 70% of campus generators, storage systems, grid import and building systems. As mentioned, the average of the preceding 15 min is archived as single data point. All available data are starting from 1 January, 2015 to 29 February, 2020. Building loads on a university campus are unique due to the academic year. Depending on the building occupancy, loads can be substantially lower during the summer break, the winter holidays, and spring break.[18]

Data set contains of real and reactive power consumption for two building categories they are building with EV charging and building without EV charging, Trade Street Wearhouse microgrid, EV charging station, Solar PV generation, Campus thermal load and storage, Demand charge related data and battery energy storage system (BESS). In this research project it focusses on energy demand forecasting so it only considers about demand charge related data set.

### Demand charge related data set

This demand charge data set includes data sets of Total Campus Load, On Campus Generation, SDG&E Import and Adjusted Demand. This data set is used to forecast energy demand of microgrid. This data is given in kW and they can be used to calculate demand charges ($/kW).

Total Campus Load provide a historic view of electric consumption, encompassing both external (imported) and internal (On-campus generation) sources. This data can be crucial for understanding the overall energy demand patterns of the entire microgrid. On Campus Generation data is used to compute the actual demand on the campus. It takes into account the electricity generated on-site by the campus's own power generation sources, such as solar panels or other generators. SDG&E import data represents the amount of electricity the campus imports from the utility company, San Diego Gas & Electric (SDG&E), measured in kilowatts (kW). Adjusted demand, which considers on-campus generation, is particularly useful for forecasting non-coincident demand charges. Analyzing the historical adjusted demand data helps in understanding how on-campus generation affects the overall demand patterns.

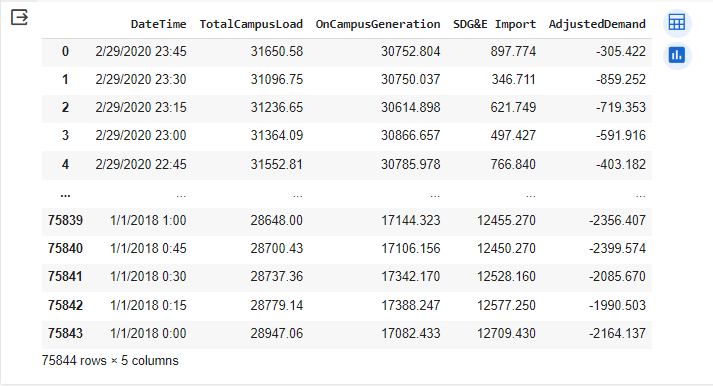
As this data can be used to calculate demand charge, they are fees based on the highest power consumption levels during specific time periods (non-coincident or peak periods). To calculate these charges, you need data on real power imports, on-campus generation, and adjusted demand to account for your on-site generation. Accurate tracking and management of these factors can help control and reduce demand charges, which can be a significant portion of a commercial or industrial electricity bill. According to this by forecasting energy demand they can also be used to forecast demand charge. Following figure shows the raw data of demand charge data set

Figure 3.1: Data set of demand charge data

Next have to identify the patterns of this data set and how it varies. So, to do that we need to understand the data set.

### Data understanding

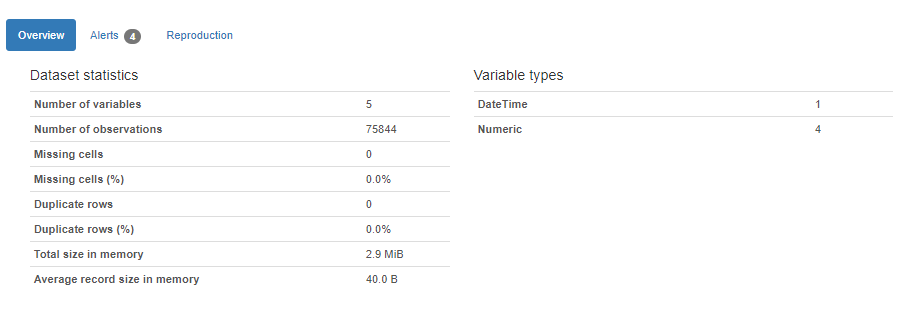
Basic data understanding is must to identify patterns of the data set. To achieve that in this research project have done an EDA on the data set. EDA stands for Exploratory Data Analysis. It is an approach to analyzing and visualizing data sets to summarize their main characteristics, often with the help of statistical graphics and other data visualization methods. The primary goals of EDA include understanding the structure of the data, uncovering patterns, identifying relationships between variables, and identifying any potential outliers or anomalies. EDA is done using pandas profiling. It is a Python library that generates an interactive HTML report with a variety of visualizations and statistical information about a DataFrame. EDA results are explained bellow.

Figure 3.2: Overview of the data set

According to this it can see that there are no missing values and there are no duplicated values in the data set. It gives alerts about correlation of the each of variables.

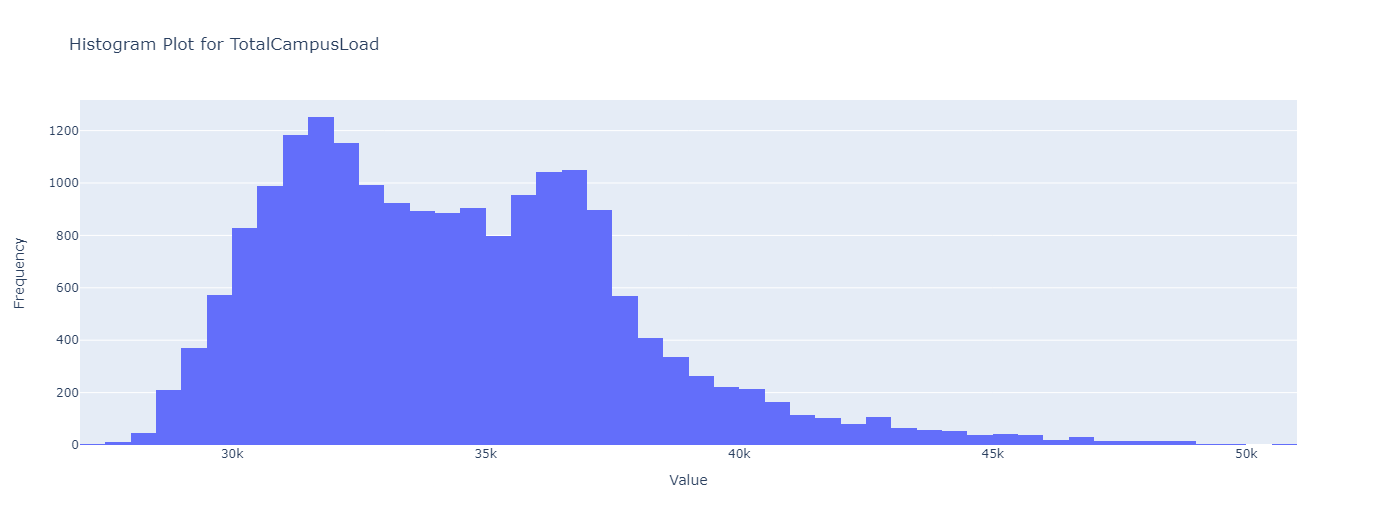
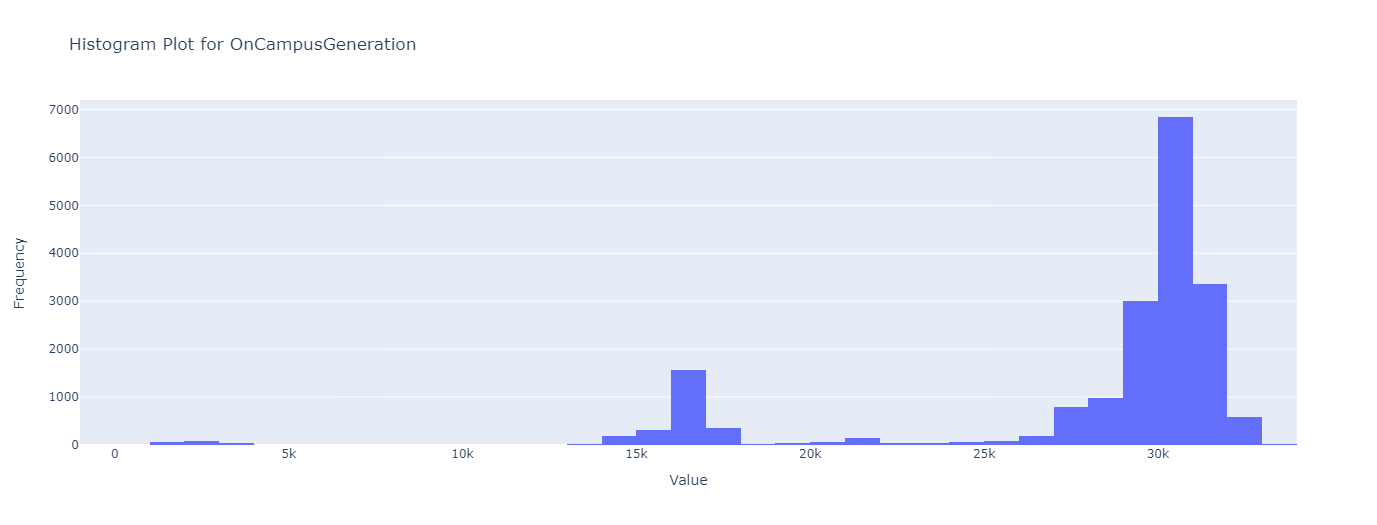
Histogram plots provides the visual distribution of a numerical variable. by plotting the histogram plots we can get basic understanding of distribution shape, central tendency, spread and dispersion, outliers etc. figure 3.3 – 3.6 shows histogram plot for each feature.

Figure 3.3: Histogram plot for Total Campus Load

Figure 3.4: Histogram plot for On Campus Generation

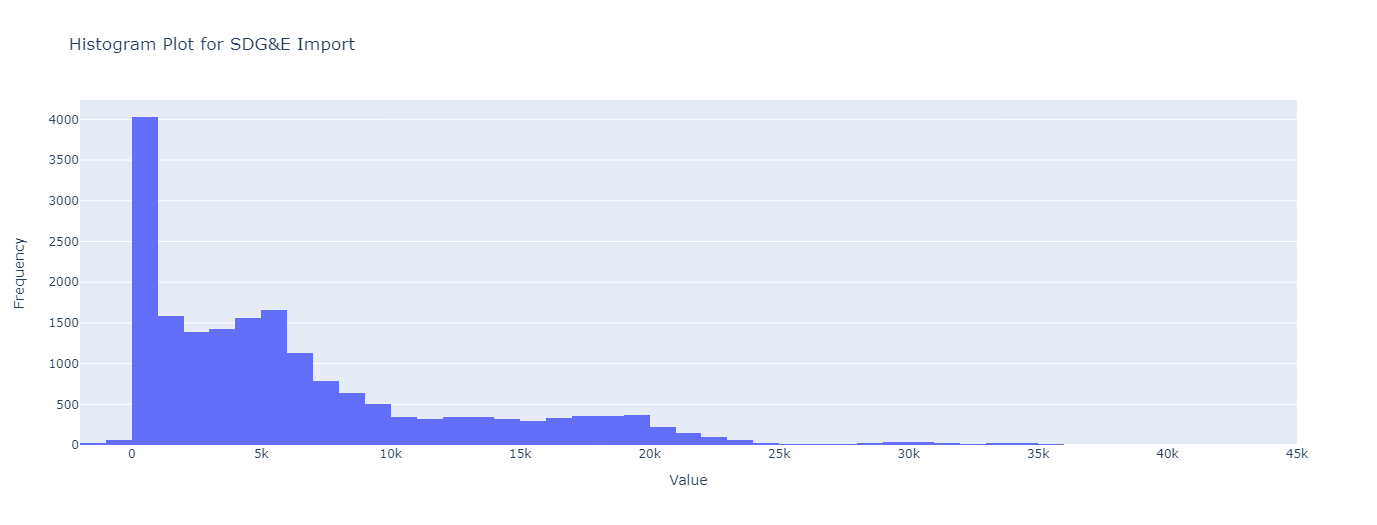


Figure 3.5: Histogram plot for SDG&E Import

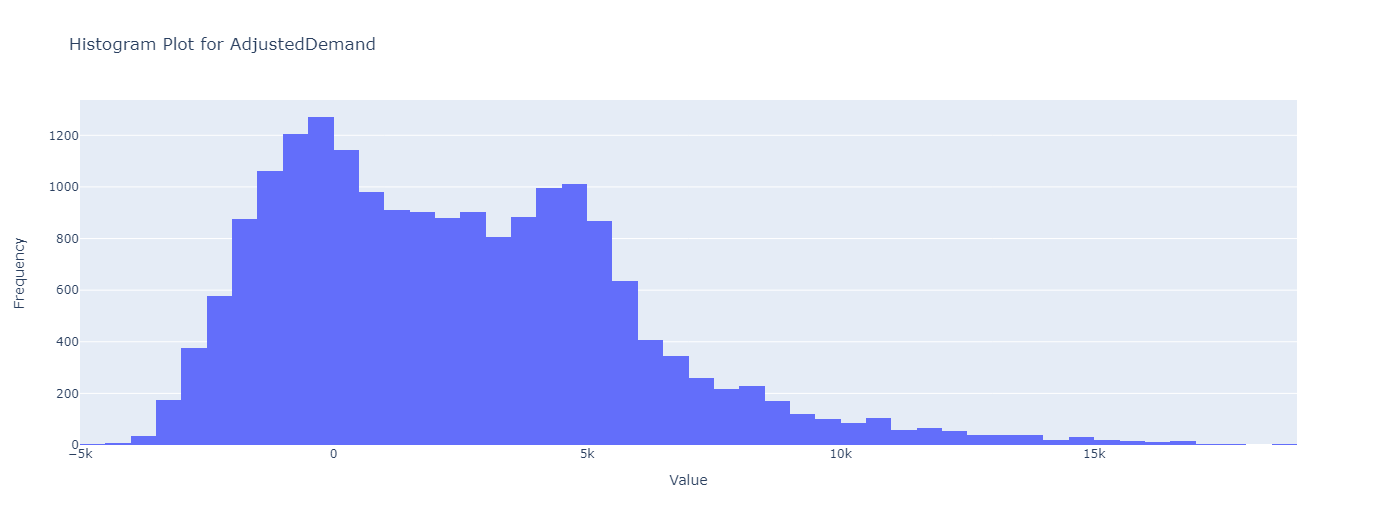


Figure 3.6: Histogram plot for Adjusted Demand

Each plot visualizes the distribution of data set, range of the data set, what are the outliers, spread of it etc. having this basic idea about data set is essential.

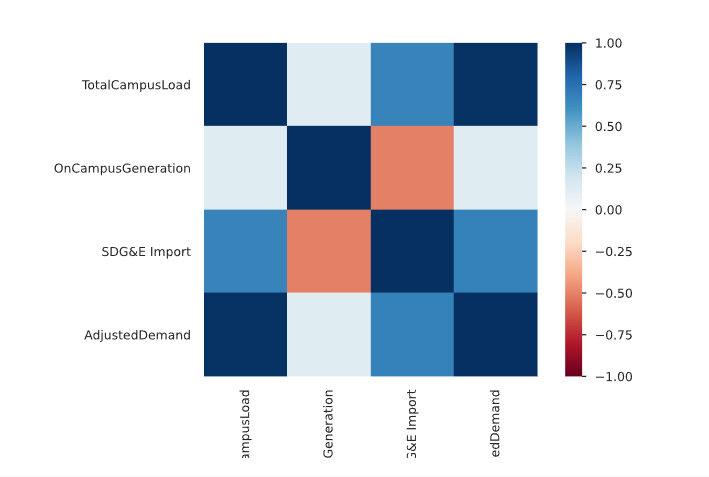
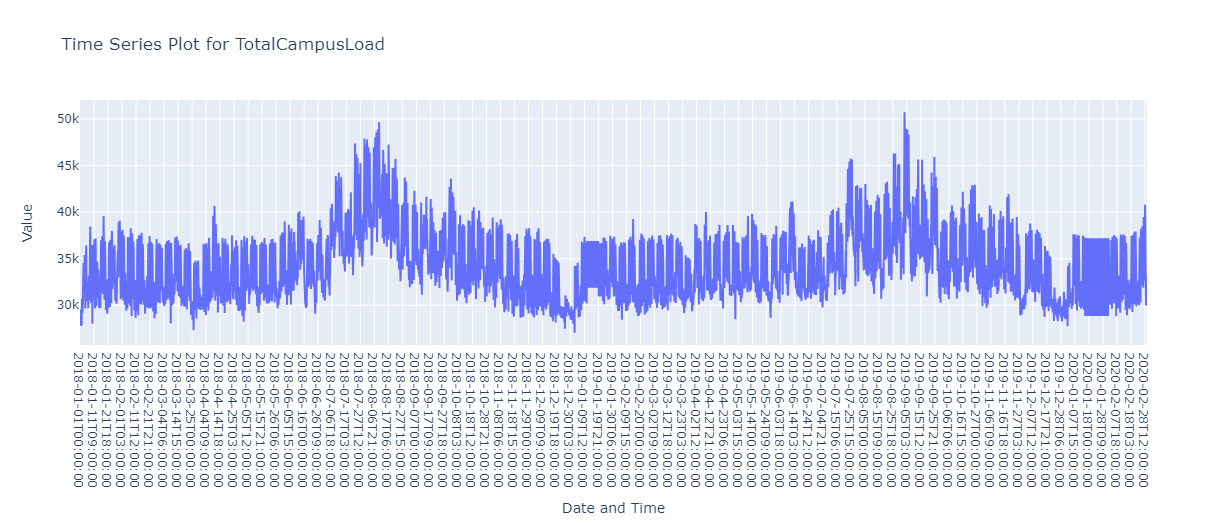
Also, it is good to know about the correlation between each of the features, this will help in understanding the dependency of each of them. This correlation can be visualized through a heat map. Heat map corresponds to demand charge data set is shown in figure 3.7.

Figure 3.7: Heat map for the data set

According to this it shows that there is high correlation between Total campus load and Adjusted demand. Then we can see the variation of the data set according to the time. By potting we can identify if there are any pattern with respect to date time or season. Usually for time series there is seasonality patterns. Following figures shows the data variations according to time.



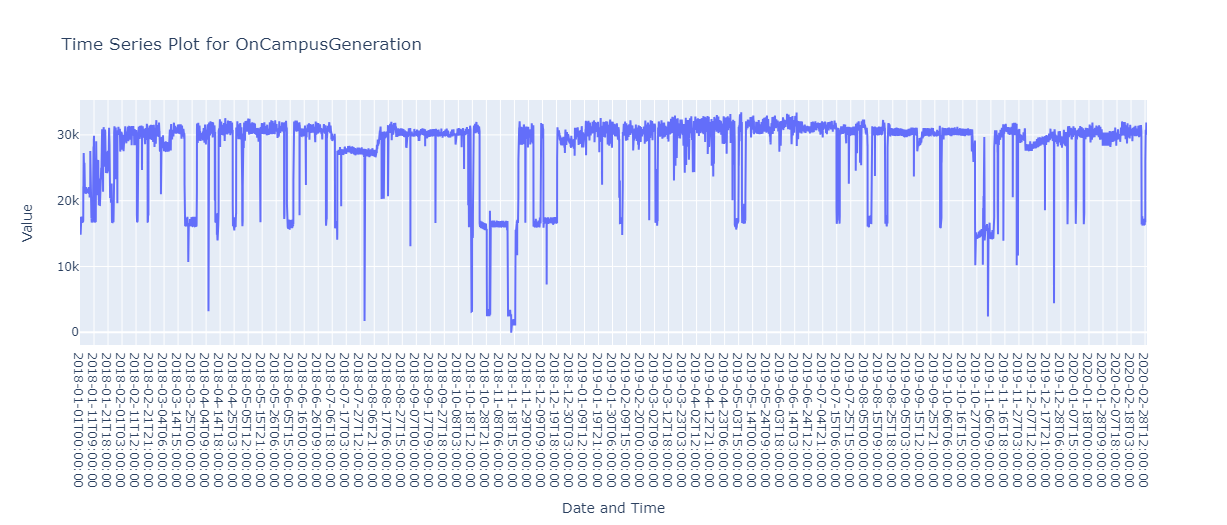
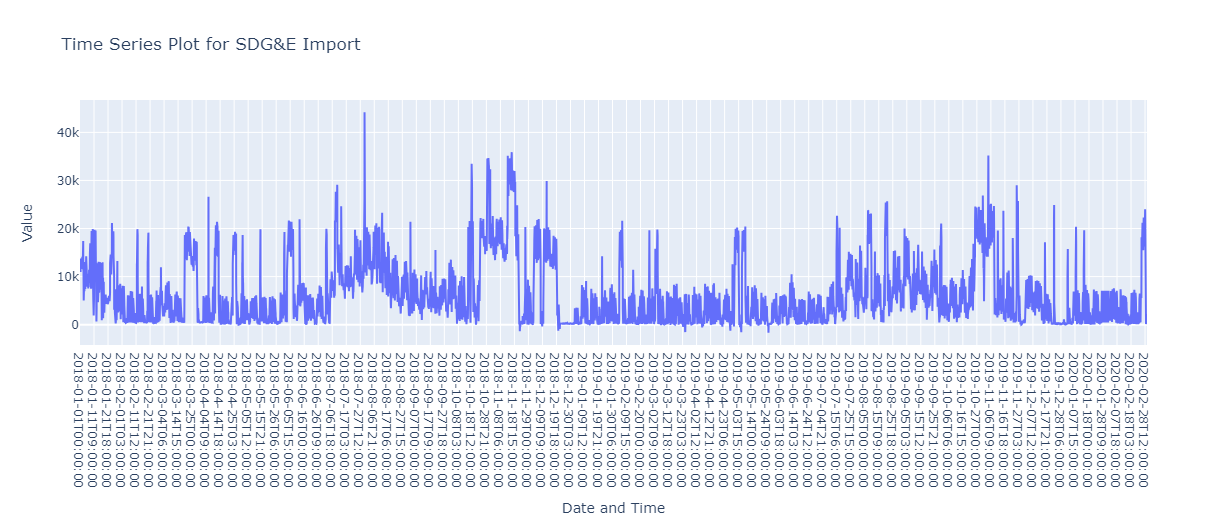
Figure 3.8: Time series plot for Total campus load

Figure 3.9: Time series plot for on campus generation

Figure 3.10: Time series plot for SDG&E import

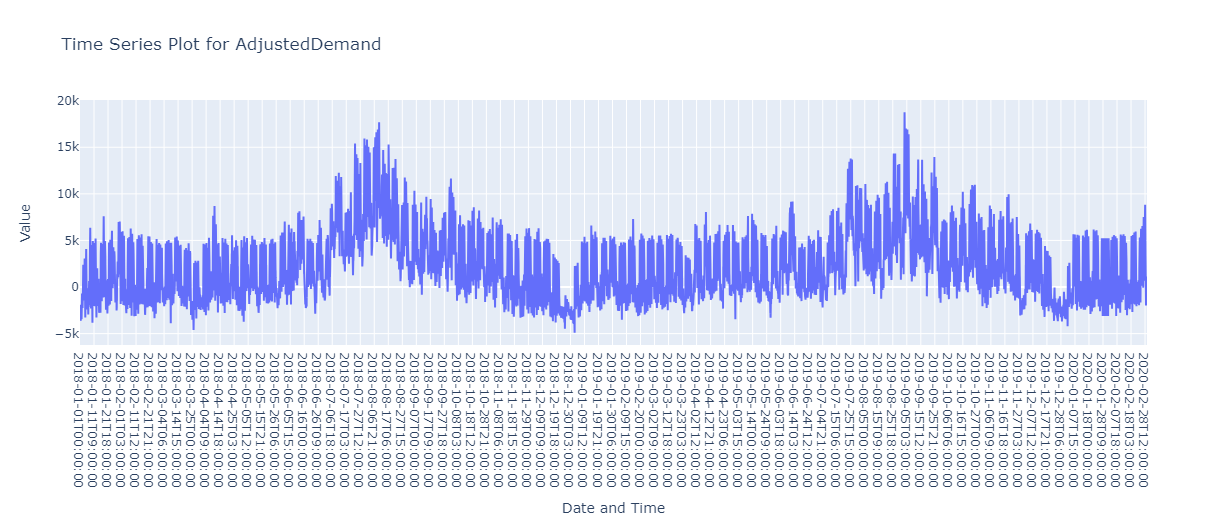


Figure 3.11: Time series plot for Adjusted demand

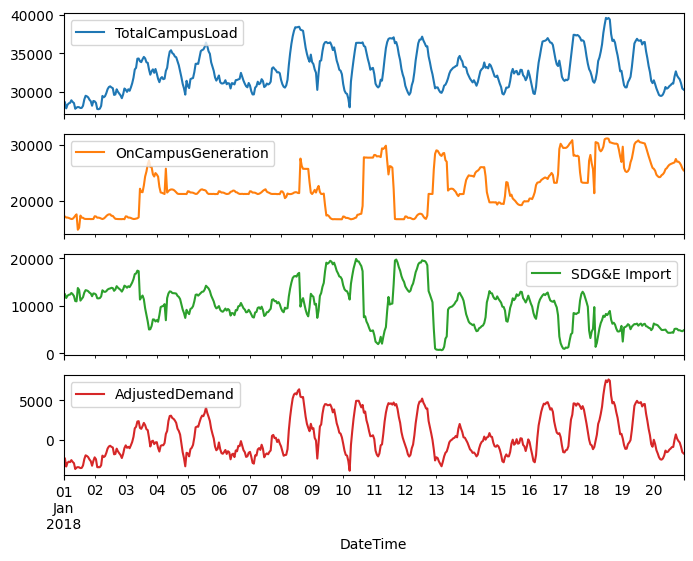
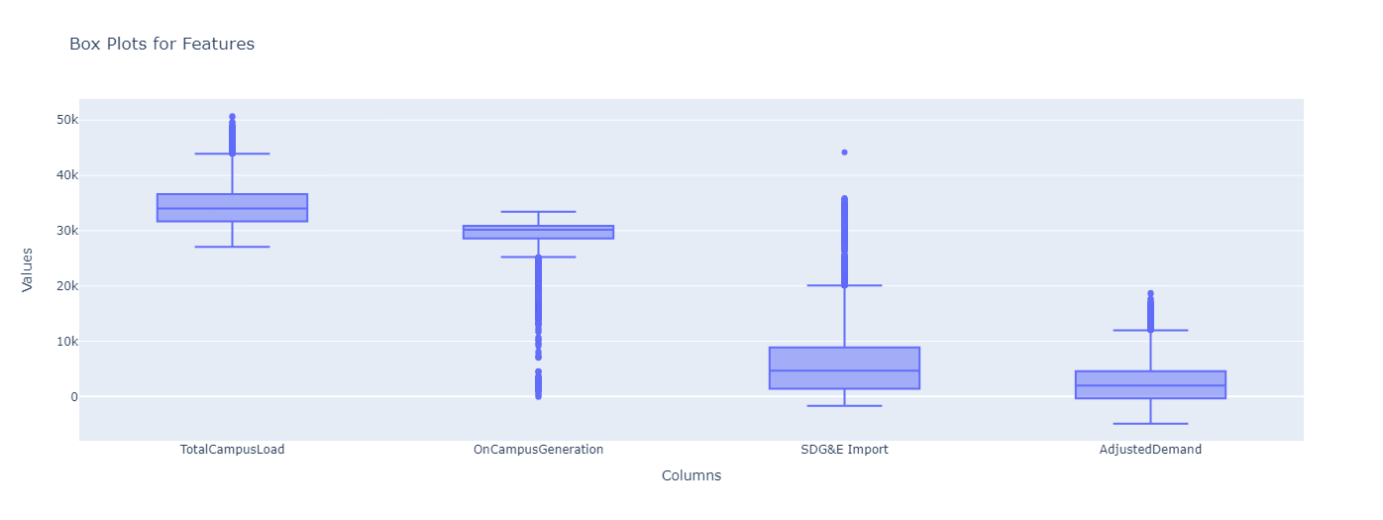
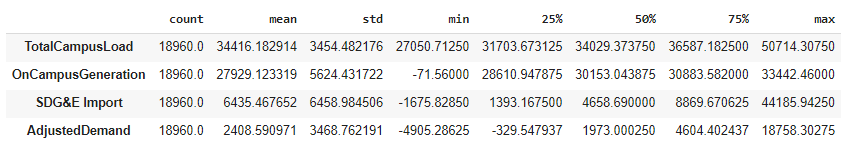
Here we can see some seasonal patterns in this plot. According to the date and weeks also it shows some pattern.

Figure 3.12: Time series plot for 20 days

Figure 3.12 shows time plot for 20 days. In this plot it can see there is a pattern. For the day time demand values are higher and they get reduce in other time. Also, it indicates that there is a peak time for a day. Box plot shows graphical representation of the distribution of a dataset. Following figure shows how box plot representation for the initial dataset.

Figure 3.13: Box plot for the features

Also, it is good to known statistic of the data set. This can obtain from a simple python code. Output is shown in following figure.

Figure 3.14: Statistic of the data set

From the data understanding we can identify what are the patterns in data set is there any seasonal patterns on that, variation of the data set and range of the data set whether there is any outlier in the data set. Then before feeding this to model, we have to preprocess the data. In next topic talk about data preprocessing.

### Data preprocessing

This part is also combined with data understanding part. Initially data set was rearrange starting from the oldest date to newest date. it was done in google colabotary. In the research project google colabotory is used as note book.

This data set was taken as 15 min time intervals. To forecast day ahead data it is good to have hourly data. So, using this 15 min time interval data we implemented hourly data. We used mean values of data within one hour to obtain hourly data. Then NaN values were removed. All these initial steps were taken before doing the data understanding part.

Then date time is converted to UNIX timestamp. As it provides a standardized way to represent time, making it easy to incorporate chronological information into your machine learning models. Then in data preparation we use sine and cosine transforms to clear "Time of day" and "Time of year" signals. This will allow to identify more seasonal patterns.

According to figure 3.13 box plot and figure 3.14 statistics of the data set indicates that basic statistics are varies from each feature so it is good to normalize the data set. This will reduce the computational power required for the model to process and it will reduce the biasedness of the outputs. In this research project we do standardization. So, to do that this data should be split to training, validation and testing sets.

This is a very critical step in modeling machine leaning model because they are essential for model development, hyperparameter tuning, and assessing generalization performance. It helps ensure that the model is not only memorizing the training data but also learning patterns that can be applied to new, unseen data. When explaining briefly about them,

**Training Set:**

The training set is used to train the model. The model learns patterns and relationships within the data, adjusting its parameters to minimize the difference between predicted and actual outcomes. A larger training set generally allows the model to learn more complex patterns.

**Validation Set:**

The validation set is used to fine-tune the model's hyperparameters and to assess its performance during training. It acts as a sort of "sandbox" where you can experiment with different configurations without contaminating the test set. This helps in preventing overfitting, where a model performs well on the training data but poorly on new, unseen data.

**Testing Set:**

The testing set is used to evaluate the final performance of the model after training and validation. This set simulates real-world scenarios where the model encounters completely new data. It provides an unbiased evaluation of the model's ability to generalize.

**Preventing Overfitting:**

Overfitting occurs when a model learns the training data too well, including its noise and outliers. By having a separate validation set, you can monitor the model's performance on unseen data during training and stop training when the model starts to overfit.

**Hyperparameter Tuning:**

Machine learning models often have hyperparameters (parameters that are not learned from the data but are set before training). The validation set helps in tuning these hyperparameters by providing a separate dataset for evaluating different configurations.

**Model Selection:**

After training multiple models with different architectures or algorithms, you can use the validation set to compare their performance and select the best-performing model for further evaluation on the test set.

**Generalization Evaluation:**

The ultimate goal of a machine learning model is to generalize well to new, unseen data. The testing set provides a final assessment of how well the model is likely to perform in real-world scenarios.

These overfittings, Hyperparameter tuning, Model selection and Generalization evaluation are done after implementing and running the model. We will talk more about them in next chapters.

Generally, this training, validation and testing sets are divided as following percentages. Training set 70% of data, validation set 20% of data and rest 10% to testing set. After splitting the dataset, they were normalized as following.

#mean and standard deviation should only be computed using the training data so that the models have no access to the values in the validation and test sets.

train\_mean = train\_df.mean()

train\_std = train\_df.std()

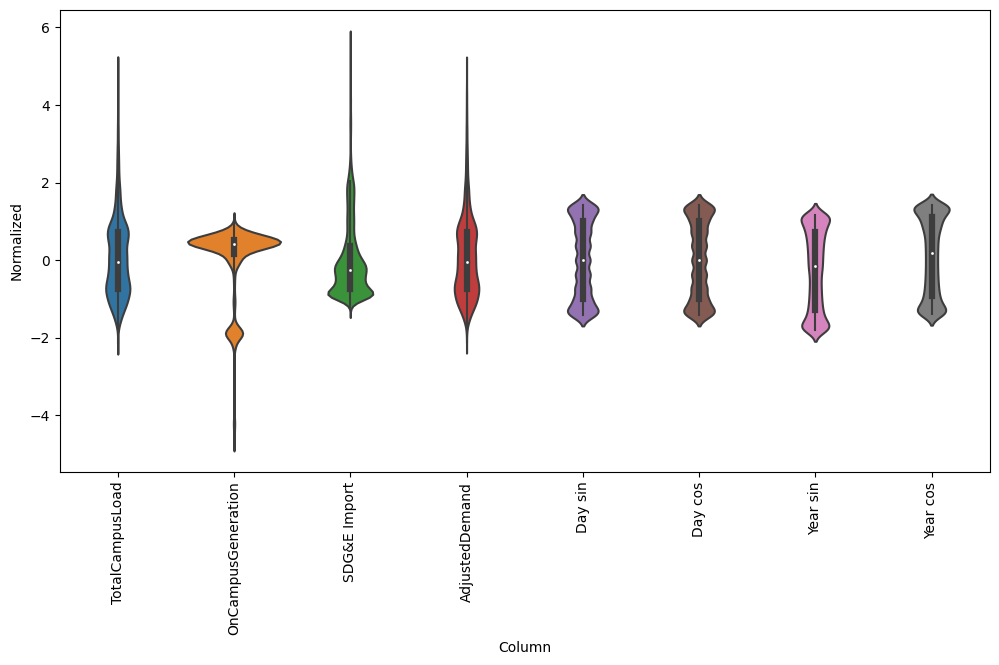
train\_df = (train\_df - train\_mean) / train\_std

val\_df = (val\_df - train\_mean) / train\_std

test\_df = (test\_df - train\_mean) / train\_std

In this normalization method it uses train mean and train standard deviation for the operation, so that the model has no access to the values in the validation and test data set.

After normalizing the data set it varies between the same region for all the features. This is good for computing the model. Following Figure 3.15 shows the violine plot for training data set

Figure 3.15: Violine Plot for the training data set

This violin plot show that numerical values of all the features. According to this their range of the features remain in between same values. From the width of the violin plot it represent the data density of that specific point. As well as using this violin plot, we can identify mediant, possible outliers and skewness of the data distribution. Doing data preprocessing critical step, it varies from the data set which is using. Using this properly processed data we can build more accurate and robust machine learning model. In the next topic going to talk about implementing of machine learning model.

## Implementing CNN model

Implementing CNN model is a major task in this research project before going to implementation part let’s look at what is CNN model and how it can be used in forecasting energy demand.

A CNN, or Convolutional Neural Network, is a type of deep neural network designed for tasks involving visual data such as images and videos. They automatically learn hierarchical features from the data, starting from simple features at lower layers (e.g., edges and textures) to more complex features at higher layers (e.g., object parts and shapes). This hierarchical feature learning makes CNNs well-suited for image-related tasks, where the spatial relationships among pixels play a crucial role.

CNN model is mainly used in image processing such as tasks involve in grid-like data. But in this research project we are using time series data set, so we are going to implement a 1-D convolutional neural network model for the forecasting process.

### 1-D CNN model

The one-dimensional Convolutional Neural Network (1D CNN) is a specialized neural network architecture uniquely designed for handling sequences of one-dimensional data. This tailored design is particularly advantageous in tasks where understanding sequential information is crucial.

Many real-world applications, data comes in the form of sequences, such as time series data, audio signals, or natural language text. These sequences have an inherent order, and the relationships between elements in the sequence are often vital for understanding and making predictions. The 1D CNN is well-suit ed for capturing and learning patterns within such sequential data.

In the context of time series data, for example, each data point represents a value at a specific point in time. By applying convolutional operations to local regions of the sequence, the 1D CNN can detect patterns and features that may be indicative of temporal relationships or trends. The ability to capture dependencies within the sequence makes the 1D CNN a powerful tool for tasks like time series forecasting, anomaly detection, and signal processing.

Using a 1D CNN has the benefit of not requiring human feature engineering because it can automatically extract pertinent features from the sequential input. As they move over the input sequence, the convolutional layers function as local filters that catch up patterns at various sizes. Pooling layers that come after aid in lowering spatial dimensions, focusing on the most crucial elements, and enhancing computing efficiency.

### Model architecture

In this section we are going to talk about how the model was implemented and what are the specifications having in the model. Initially we have to do data windowing.

Data windowing, refers to the practice of dividing a continuous sequence of data into smaller, overlapping or non-overlapping segments called windows. Each window captures a subset of the original sequence, and this technique is commonly employed for various purposes, such as time series analysis, feature extraction, analysis, and model training. In this research project this data windowing is used in time series data forecasting.

When considering time series data forecasting it breaks the time series into windows allows for the creation of training samples. Each window can be treated as a data point, with the goal of predicting the subsequent values. This is useful approach when using CNNs.

**Data Windowing**

For data windowing, WindowGenerator class was implemented. In this the main features of the input windows are.

* The width (number of time steps) of the input and label windows.
* The time offset between them.
* Which features are used as inputs, labels, or both.

This WindowGenerator function we can be used in

* Single-output, and multi-output predictions.
* Single-time-step and multi-time-step predictions.

As we are forecasting day ahead data, we are using multi output and multi time step predictions.so this helps in organizing and generating windows of data for training and evaluating time series forecasting models.

**Splitting window**

Then split\_window function was created to convert given consecutive inputs to a window of inputs and a window of labels. It separate generated windows to labels and inputs. As this project is doing day ahead data forecasting we have to create window of 48, in that window there are 24 inpluts and 24 labels. So we have been created multi step window model as multi\_window.

#create a window for multy step models

OUT\_STEPS = 24

multi\_window = WindowGenerator(input\_width=24,

                               label\_width=OUT\_STEPS,

                               shift=OUT\_STEPS)

this created window can be shown as follows.

Total window size: 48

Input indices: [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23]

Label indices: [24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47]

Label column name(s): None

As mentioned above there are total number of 48 windows with 24 inputs and 24 labels. This label means it represent actual output data points. So, using the input indices we can predict the day ahead output and it can be validated with comparing label values.

**Plotting**

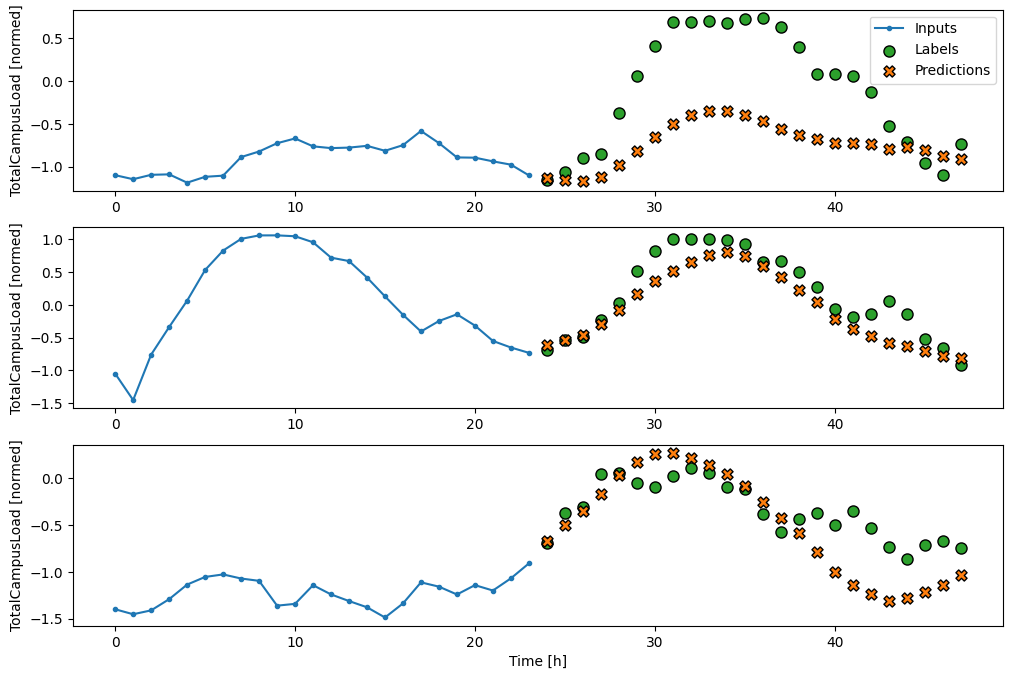
Then for the plotting implemented plot function. In this it shows outputs for random three data sets in training set or using example window we can plot for three data sets. This plot shows output related to window that we created from WindoGenerator it represents inputs label values and predicted values.

Figure 3.16: Sample plot for Total campus load

Above Figure 3.16 shows sample output for random three places in training set. As this window has 24 inputs 24 labels this shows predicted values for label values which are implemented using input values.

**Creating tf.dataset**

Then from make\_dataset method will take a time series DataFrame and convert it to a tf.data.Dataset of (input\_window, label\_window) pairs using the tf.keras.utils.timeseries\_dataset\_from\_array function. Then properties was added for accessing them as tf.data.Datasets using the make\_dataset method you defined earlier. This The tf.data.Dataset objects are particularly useful for efficiently handling large datasets and integrating seamlessly with TensorFlow's training pipelines.

**Compile and fit**

Then the training process was packaged to single function called compile\_and\_fit. This function is designed for streamline the training process for a given machine learning model. Let’s look at some details about it.

One essential component is the EarlyStopping callback, which monitors the validation loss during training. This callback interrupts the training if the validation loss fails to improve over a predefined number of consecutive epochs. By preventing overfitting, this feature ensures that the model generalizes effectively to new, unseen data.

    # Callback for early stopping with patience

    early\_stopping = tf.keras.callbacks.EarlyStopping(

        monitor='val\_loss',

        patience=patience,

        mode='min',

        restore\_best\_weights=True  # Restore the best model weights upon early stopping

)

this function includes learning rate schedule to dynamically adjust the learning rate throughout the training process. This is implemented with ExponentialDecay, starts with a relatively high learning rate and gradually reduces it over time. This will facilitate quicker convergence in the initial training phases

    # Learning rate schedule

    lr\_schedule = tf.keras.optimizers.schedules.ExponentialDecay(

        initial\_learning\_rate=1e-3,

        decay\_steps=1000,

        decay\_rate=0.9

    )

Also, it uses Adam optimizer with the aforementioned learning rate schedule. And for the compilation of the model, the function utilizes Mean Squared Error (MSE) as the loss function. MSE is a common choice for regression tasks, measuring the average squared difference between the predicted and actual values. Additionally, Mean Absolute Error (MAE) is included as a metric, providing a straightforward measure of prediction accuracy.

    # Adam optimizer with learning rate schedule

    optimizer = tf.keras.optimizers.Adam(learning\_rate=lr\_schedule)

    # Compile the model with mean squared error loss and mean absolute error metric

    model.compile(

        loss=tf.keras.losses.MeanSquaredError(),

        optimizer=optimizer,

        metrics=[tf.keras.metrics.MeanAbsoluteError()]

    )

Finally, the function returns the training history, which includes information about the model's performance metrics (e.g., loss, MAE) over the course of training epochs. This history is valuable for analyzing the model's behavior, identifying potential issues, and making informed decisions about further training or model adjustments

    # Train the model

    history = model.fit(

        window.train,

        epochs=MAX\_EPOCHS,

        validation\_data=window.val,

        callbacks=[early\_stopping]

    )

    return history

so, we can have an idea about the comple\_and\_fit function and how its functionality. Then considering implementing model. We implemented baseline model other than CNN model.

**Repeat baseline model**

Baseline model serves as crucial reference point for evaluating the performance of more sophisticated models. It provides simple, intuitive benchmark that allows practitioners to assess whether their advanced models are adding substantial value or if their complexity is justified. Baseline model is used to compare performance, check model sanity, to find resource allocation, to mitigate risk and to check interpretably. For this research project we implemented repeat baseline model. Where it takes predicted value as previous data value.

#create baseline model where it repeate the previous day values

class RepeatBaseline(tf.keras.Model):

  def call(self, inputs):

    return inputs

repeat\_baseline = RepeatBaseline()

repeat\_baseline.compile(loss=tf.keras.losses.MeanSquaredError(),

                        metrics=[tf.keras.metrics.MeanAbsoluteError()])

this model is compiled using compile method which is implemented. In the loss function it measure the mean squire difference between actual values and predicted values. Additionally, the model is configured to track the mean absolute error as a metric. Then performance of this model is stored in,

multi\_train\_performance = {}

multi\_val\_performance = {}

multi\_test\_performance = {}

lists with indicating relevant model

multi\_train\_performance['RepeatBaseline'] = repeat\_baseline.evaluate(multi\_window.train)

multi\_val\_performance['RepeatBaseline'] = repeat\_baseline.evaluate(multi\_window.val)

multi\_test\_performance['RepeatBaseline'] = repeat\_baseline.evaluate(multi\_window.test, verbose=0)

then the plotting was done multi\_window.plot(repeat\_baseline)

calling this command.

Comparing performance and data evaluation is done in next chapter. Next let’s look at the CNN model.

**1-D CNN model**

CNN model is build using TensorFlow and Keras. This model captures temporal patterns in the input data. The model is compiled and trained using the provided windowed dataset.

CONV\_WIDTH = 8

CNN\_model = tf.keras.Sequential([

    # Lambda layer to select the last CONV\_WIDTH time steps

    tf.keras.layers.Lambda(lambda x: x[:, -CONV\_WIDTH:, :]),

    # Convolutional layer with ReLU activation and He initialization

    tf.keras.layers.Conv1D(256, activation='relu', kernel\_size=CONV\_WIDTH, kernel\_initializer='he\_normal'),

    # Batch Normalization layer

    tf.keras.layers.BatchNormalization(),

    # Dropout layer for regularization

    tf.keras.layers.Dropout(0.2),

    # Dense layer with linear activation and He initialization

    tf.keras.layers.Dense(OUT\_STEPS\*num\_features, activation='linear', kernel\_initializer='he\_normal'),

    # Reshape layer to match the output shape

    tf.keras.layers.Reshape([OUT\_STEPS, num\_features])

])

history = compile\_and\_fit(CNN\_model, multi\_window)

this is the implemented CNN model. It defines the structure of the model, outlining the layers and operations crucial for capturing temporal patterns in the input time series data. The Lambda layer, with its custom operation, selectively focuses on the last CONV\_WIDTH time steps, offering a targeted window for the subsequent convolutional operation. CONV\_WIDTH represents the width of the convolutional layer, indicating the number of time steps the model will consider in each convolutional operation.

The tf.keras.Sequential class to organize the layers in a linear stack. Each layer contributes to the model's ability to capture and understand patterns within the time series data. The incorporation of a dense layer with a linear activation function and an appropriate reshaping operation ensures the model's output aligns with the desired format for time series forecasting.

The training process is facilitated by the compile\_and\_fit function, which configures the model for training by specifying the loss function, optimizer, and training metrics. The use of early stopping and a learning rate schedule with exponential decay further enhances the model's training stability and efficiency.

Data evaluation and plotting is done as follows,

# Evaluate

multi\_train\_performance['CNN'] = CNN\_model.evaluate(multi\_window.train) #training data performance

multi\_val\_performance['CNN'] = CNN\_model.evaluate(multi\_window.val) #validation data performance

multi\_test\_performance['CNN'] = CNN\_model.evaluate(multi\_window.test, verbose=0) #test data performanve

# Plot the performance

multi\_window.plot(CNN\_model)

More about model evaluation and model optimization is discussed in next chapter. After discussing about implementation of CNN model how is its architecture next part of methodology is how the model evaluation is done.

## Model evaluation

Model evaluation is a crucial part in machine learning pipeline that assesses the performance of a trained model. It helps to understand how well your model generalizes to new, unseen data and whether it meets the desired criteria for a specific task. There are various metrics and techniques used for model evaluation, depending on the nature of the problem. As this is a time series regression problem, we can use regression matrices and techniques such as Mean Absolute Error (MAE), R squared value, cross validation, lag plots.

When considering metrics, commonly used metrics include the Mean Absolute Error (MAE), which calculates the average absolute differences between predicted and actual values, the Mean Squared Error (MSE), which computes the average of squared differences, and the Root Mean Squared Error (RMSE), providing a scaled interpretation of MSE. Mean Absolute Percentage Error (MAPE) expresses errors as a percentage of the actual values, offering a relative measure of accuracy. These metrics quantify the precision of predictions but may not fully capture the model's ability to handle temporal dependencies.

Other methods like lag plots where it Plot the predicted values against the actual values with a time lag. This will visually assess the model's ability to capture temporal patterns. And cross validation methods, like rolling origin or walk-forward validation, take into account the chronological order of data points.

As well as we can get idea about overfitting and underfitting by loss function. Overfitting occurs when a model learns the training data too well. The model becomes overly complex, fitting the training data with high precision but failing to generalize effectively. Signs of overfitting include low training error but high test (or validation) error. These overfit models can exhibit poor performance on new, unseen examples. Underfitting occurs when a model is too simple to capture the underlying patterns in the training data. It fails to learn the relationships and exhibits poor performance both on the training set and new data. Signs of underfitting include high training error and high test (or validation) error. So, to reduce this we are using drop out layer in the CNN model. This can be visualized by plotting training loss and validation loss over epoch.

In this research project we evaluate the performance by calculating regression metrics, using MSE, MAE and R squared values. They are calculated separately for training validation and test sets. For the loss function considered MSE. By plotting training loss and validation loss we get idea about overfitting and underfitting. As well as R squared value also calculated this will give a idea about how well the model is fitted to the data set. This also can be considered as accuracy. So plotting R squared value over epoch gives an idea about accuracy. Using lag plots visualize the predicted values vs actual values variations. In this research project we have plotted actual vs predicted value plot for random three places in the training data set this is for the data window size and plotted for all test set. Then we can visualize the variations.

For the model evaluation part, we have followed above mentioned procedures. More about these evaluation results are discussed in the results section. next step in methodology is how to deploy forecasted data in optimization.

## Deploying forecasted results in optimization

Optimization of microgrid is very important because it brings about various economic, environmental, and reliability benefits. As we know Microgrids play a crucial role in enhancing energy efficiency by utilizing distributed energy resources (DERs) like solar panels, wind turbines, and energy storage systems. from the optimization, these resources can be effectively managed, minimize waste and significantly improve overall energy efficiency. Also, optimization strategies are contributed in cost saving. By intelligently dispatching energy resources based on demand patterns, operators can minimize operational costs. Although optimizing microgrid useful in grid resilience, demand response etc. optimizing microgrid gives huge benefits to customers.

How to use this data in optimization process? To answer that we need to find what are the ways that can be used in microgrids, what are the algorithms can be used, how to use them, how will they prepare. These things have to be considered in this part.

Considering how these data can be used. Predictive modeling, powered by machine learning algorithms, provides accurate forecasts for energy demand and renewable energy production based on historical data. Energy storage optimization ensures the efficient use of storage systems by dynamically adjusting charging and discharging patterns in response to predicted demand and generation.

By allocating jobs at times when renewable energy is most abundant and motivating users to modify their usage during peak hours, load shifting and demand response systems maximize energy consumption. Buildings that are connected to the grid electronically enhance efficiency by modifying energy usage in response to current grid circumstances. By combining these tactics, microgrid optimization may be approached holistically and data-centrically, fostering sustainability, efficiency, and agility in microgrid operations. These ways we can allocate data to optimization process.

Microgrid optimization involves employing various optimization techniques to achieve efficient resource allocation, considering both constraints and objectives. A widely used approach is linear programming, which involves formulating microgrid optimization issues as models for linear programming. This technique allows for the linear combination of various resources under certain limitations to identify optimal solutions. To handle more complex scenarios with discrete decision variables, Mixed-Integer Linear Programming (MILP) extends linear programming. Additionally, metaheuristic algorithms, such as Genetic Algorithms, Particle Swarm Optimization, or Simulated Annealing, provide an alternative means to find near-optimal solutions for intricate microgrid optimization problems. These varied optimization techniques improve microgrid performance and efficiency in a range of operational scenarios.

More about these techniques and what are the possible scenarios can be used are discussed in next chapter.

# Optimization

This chapter will be discussed more about how this predicted data is deployed in optimizing microgrid operation. Let’s initially look at what is why optimizing microgrid is important.

## Why optimizing microgrid

optimizing microgrids is a multifaceted approach that considers economic, environmental, and reliability factors. It plays a pivotal role in advancing sustainable and resilient energy systems, particularly in the context of a changing energy landscape and the increasing adoption of distributed energy resources. Here are some reasons why optimizing microgrid is important

Microgrids play a crucial role in enhancing energy efficiency by utilizing distributed energy resources (DERs) like solar panels, wind turbines, and energy storage systems. Through optimization, these resources are effectively managed, minimizing waste and significantly improving overall energy efficiency, particularly in regions with limited or unreliable access to centralized power grids.

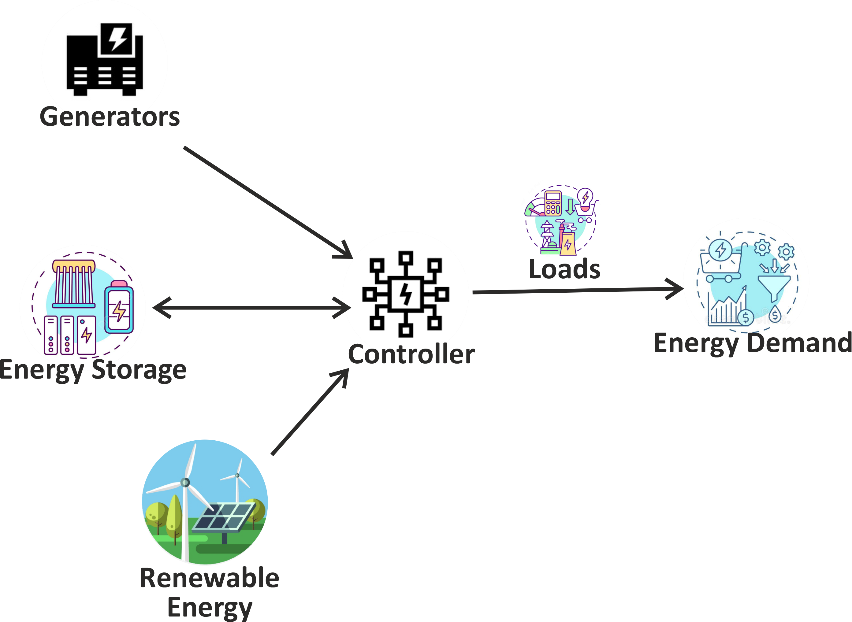
Microgrid optimization techniques also result in significant cost reductions. Operators can reduce operating costs by strategically allocating energy resources according to patterns of demand. In order to promote economic efficiency, this involves lowering the costs related to obtaining power from the grid during peak hours and maximizing the usage of renewable energy sources.

Another key benefit of optimizing microgrids is the enhancement of grid resilience. Microgrids can operate independently or in conjunction with the main power grid, providing a reliable source of power during grid outages or emergencies. The integration of renewable energy is facilitated by optimization in many microgrids. This process helps balance the intermittent nature of renewables, ensuring a stable and continuous power supply. Also allows consumers to adjust their energy usage based on supply conditions and pricing

In remote or isolated areas, microgrids offer a pathway to energy independence. By optimizing the use of locally available resources such as solar and wind energy, these communities can reduce dependence on external energy sources, enhancing their overall resilience and sustainability. So, optimization supports microgrid operators in complying with regulatory standards and environmental policies. Ensuring adherence to regulations related to energy efficiency, renewable energy integration, and emissions reduction is critical. Optimization helps avoid potential legal or financial consequences, ensuring the responsible and compliant operation of microgrids.

In this research project we are focusing on how this forecasted data can be deployed in optimizing microgrids. Initially let’s identify some operating process of a microgrids.

## Operating process of microgrids

As we know microgrid are consist with distributed energy resources. And it can be connected to main gird. This distributed energy resources can be wind turbine, solar panels, combine heat and power plants and energy storage systems. So, in here we are considering microgrid system consist of wind turbine, solar panel, energy storage system like batteries, generator and main grid. This is considered as common type microgrid. We can identify different kind of operation process.

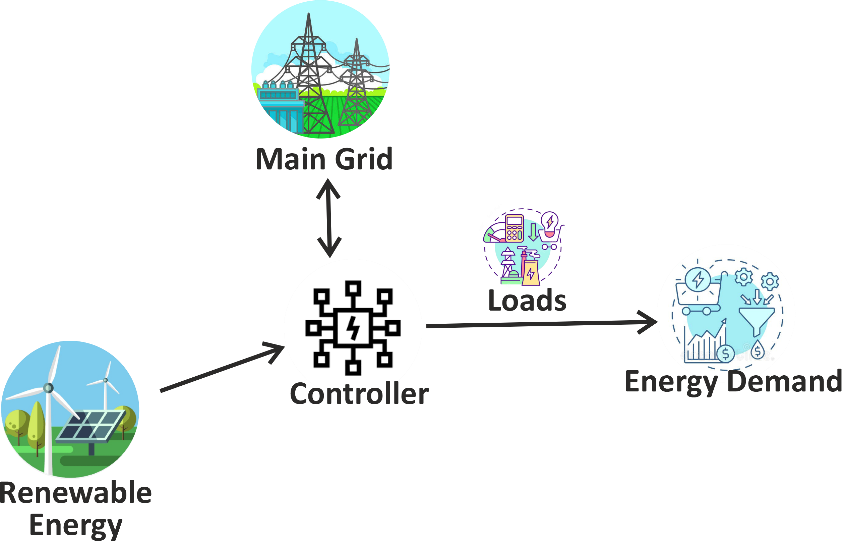
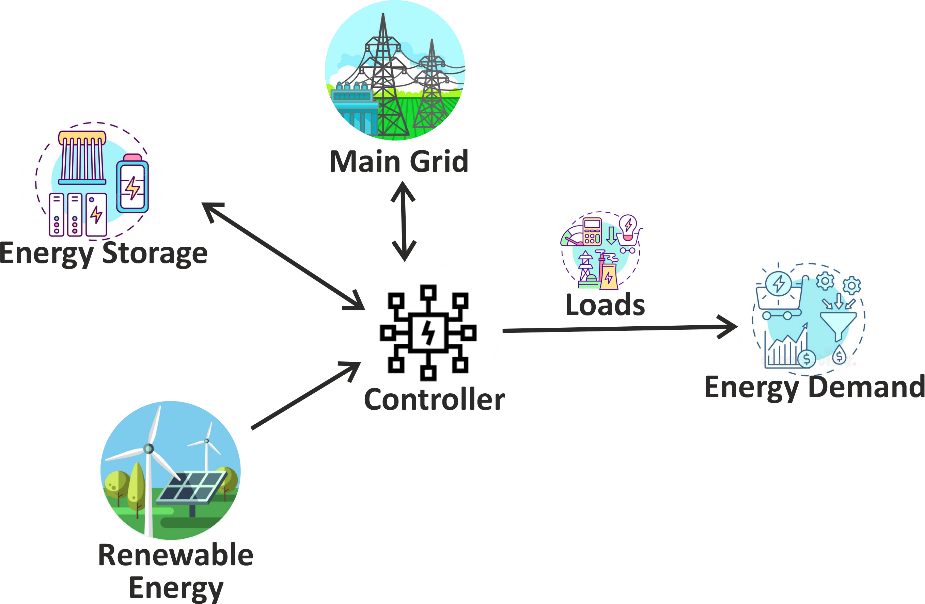
Figure 4.1 : operating process A

Figure 4.2: Operating process B



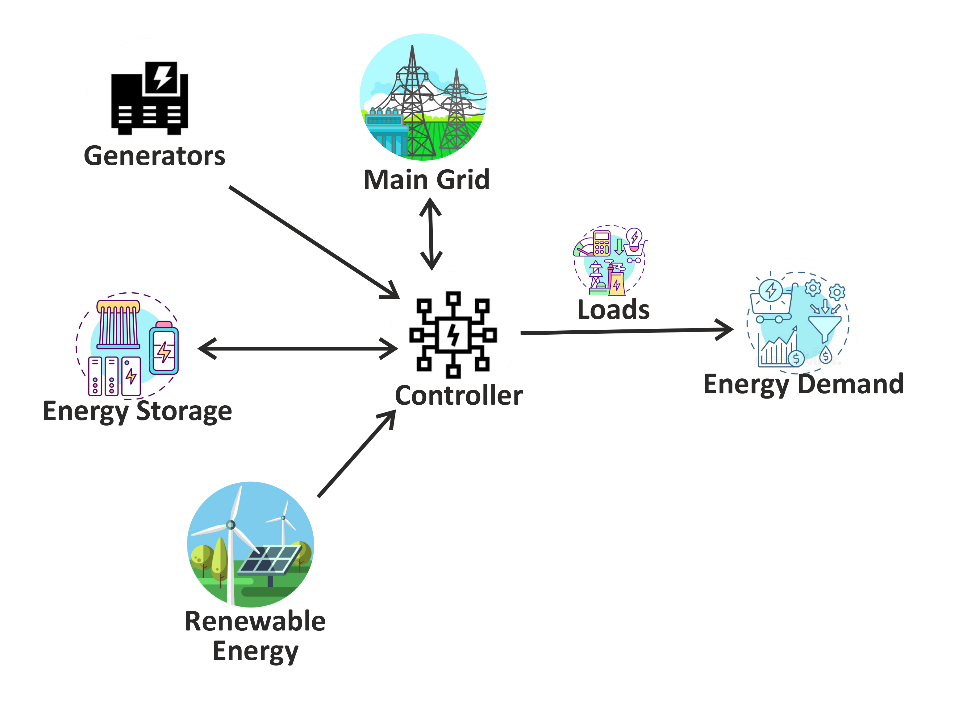
Figure 4.3 : Operating process C

Figure 4.4 : Operating process D

These operating processes are different combinations of components which included in microgrids. They can be operated in different conditions. Following table includes components of each operating process. All of them are connected to the controller and through the controller it determines what are the components are going to activate at that particular time and connections between them

Table 4.1:Components in Operating process of microgrid

|  |  |
| --- | --- |
| Operating Process | Components |
| A | Include Renewable Energy sources, generators, battery storage system. Not connected to the grid. |
| B | Include Renewable Energy sources. Connected to the grid. |
| C | Include Renewable Energy sources, battery storage system. Connected to the grid. |
| D | Include Renewable Energy sources, generators, battery storage system. Connected to the grid. |

Let’s consider description of each operating process.

Table 4.2: Operating process description

|  |  |
| --- | --- |
| Operating process | Description |
| A | Energy generation is done using RES. As this is not connected to the main grid. Execs energy is store in the battery. If there is surplus in energy initially, they consider energy store in battery then they use generators. |
| B | Energy generation is done using RES. If there is execs energy they were send to main grid. In surplus of energy amount required is taken from main grid |
| C | Energy generation is done using RES. If there are execs energy, they were store in battery. As well as if there are more energy they were send to main grid. In surplus of energy amount required is taken from battery and main grid. |
| D | Energy generation is done using RES. If there are execs energy, they were store in battery. As well as if there are more energy they were send to main grid. In surplus of energy amount required is taken from battery, main grid and generators. This can be determined according to the situation. If cost is reduced when taking more from grids at off peak time this can be done in this operating process. |

In above table include description about operating process of each method. Microgrid can be operated in any method of this. Then we have to look at what are the optimization algorithms and how they can be used.

## Algorithms and how to use them

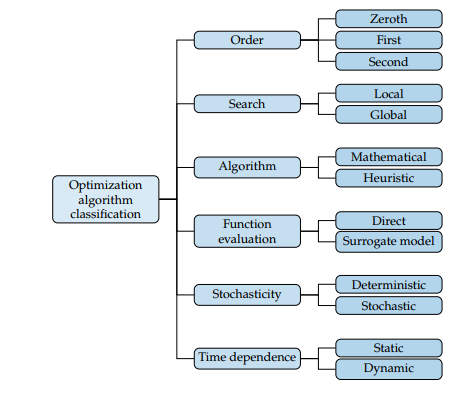
There are several optimization algorithms available. Among them we have to choose what is the most suitable for this scenario. Initially we have to find the objective of optimization. In this research project we have chosen objective functions as to reduce cost and emission. This can be achieved independently or both at once. Reducing cost means optimizing microgrid with currently available energy sources. Such as for day time we can use main grid combined with DERs. At the peak times we can used microgrid energy. So, we can decide what are the amount and what is the time we need to use each resource. To achieve that we can use optimization algorithms. And we and we identified some algorithm types that can be used in this scenario. Here are some classifications of optimization algorithms.

Figure 4.5: Optimization algorithm classifications

According to figure there are different kind of optimization algorithms. Such as according to algorithm, stochasticity, time dependency. Here are some algorithms that identified among them.

First of all, considering problem formulation we can use single objective optimization problems and multi objective optimization algorithms. As mentioned, objective of this optimization has been identified as reducing cost and emission. single objective optimization algorithms can be used when we are going to achieve this objective one by one like if we need to optimize only cost, we can use single objective optimization algorithms. Then we define relevant parameters and use the algorithm. If we are up to use more than one objective in optimization, we have to use multi objective optimization algorithms. Here we can define weights according to each objective then use algorithm as necessary.

So, according to figure 4.5 optimization algorithms we can use many combinations in our optimization problem. When considering accuracy of the forecasted data there are two types of algorithms can be used. If we assume forecasted data are accurate can use deterministic optimization algorithms. Else forecasted data are uncertain we can use stochastic optimization algorithms. As A deterministic optimization algorithm always evaluates the same points and converges to the same result, given the same initial conditions. In contrast, a stochastic optimization algorithm evaluates a different set of points if run multiple times from the same initial conditions.

Another attribute that has to consider is time dependency. as demand is depend on the time. there are patterns and variation with the time for demand. Considering time dependency is must. We can use dynamic optimization algorithms on this case. dynamic optimization problems solve a sequence of optimization problems to make decisions at different time instances based on information that becomes available as time progresses. Also, from above algorithm classifications we can use mathematical algorithms such as linear algorithms.

No single optimization algorithm is effective or even appropriate for all possible optimization problems. This is why it is important to understand the problem before deciding which optimization algorithm to use. By “effective” algorithm, we mean that the algorithm can solve the problem, and secondly, it does so reliably and efficiently. So, selecting appropriate algorithms drives us to finest results. In this scenario we can use above mentioned any combination of algorithms.

## Approach of forecasted data on optimization

Applying forecasted data to optimize microgrid operation is a strategic approach to dynamically adjust energy supply and demand for efficient and cost-effective performance. By incorporating forecasted energy demand, the optimization model gains insights into future consumption patterns, enabling proactive decision-making. This process involves leveraging predictive analytics and forecasting techniques to anticipate energy demand variations.

Optimization model uses forecasted data to adapt the microgrid's energy production, storage, and distribution strategies. For instance, if the forecast predicts a surge in energy demand during specific hours, the microgrid can proactively allocate additional resources or adjust the energy distribution to meet the expected requirements. Conversely, during periods of lower forecasted demand, the microgrid can optimize resource utilization to avoid unnecessary energy generation or storage.

This dynamic adjustment based on forecasted data contributes to several benefits in microgrid operation. First of all, it reduces energy waste by coordinating output with anticipated demand, improving energy efficiency. Second, it makes operations more cost-effective by making the most use of the resources at hand and minimizing reliance on outside energy sources during periods of high demand, which lowers the price of purchasing electricity.

The integration of forecasted data in microgrid optimization also supports grid resilience. Anticipating fluctuations in energy demand allows the microgrid to prepare for contingencies, ensuring a stable power supply during unexpected events. This resilience is crucial for maintaining uninterrupted power to critical infrastructure and improving the overall reliability of the microgrid.

Also. the utilization of forecasted data enables the microgrid to contribute to environmental sustainability goals. By optimizing the integration of renewable energy sources based on forecasts, the microgrid can enhance the share of clean energy in its overall energy mix. This aligns with efforts to reduce carbon emissions and promote sustainable energy practices.

Here we can see that applying forecasted data to optimize microgrid operation gives predictive insight to enhance energy efficiency, cost-effectiveness, grid resilience, and environmental sustainability. It Enables microgrid operators to make knowledgeable judgments and proactively address fluctuations in energy use enhances the overall efficiency and dependability of the microgrid system.

# Results

This research project we have implemented three models for forecasting they are repeat baseline model and two CNN models. Two CNN models are differing from their model architecture this chapter discuss about result and outputs of the CNN model and how model performance varies. this helps to understand how well your model generalizes to new, unseen data and whether it meets the desired criteria for a specific task.

## Output visualization

Plotting actual data vs predicted data involves visually inspecting the alignment of the predicted values with the actual values. Here we are assessing the alignment of data points, overall trend accuracy, and the model's ability to predict peaks and troughs. For this in this CNN model we plotted actual vs predicted data for one window size in training data set and then for all data points in testing data set. There we can visualize data variations.

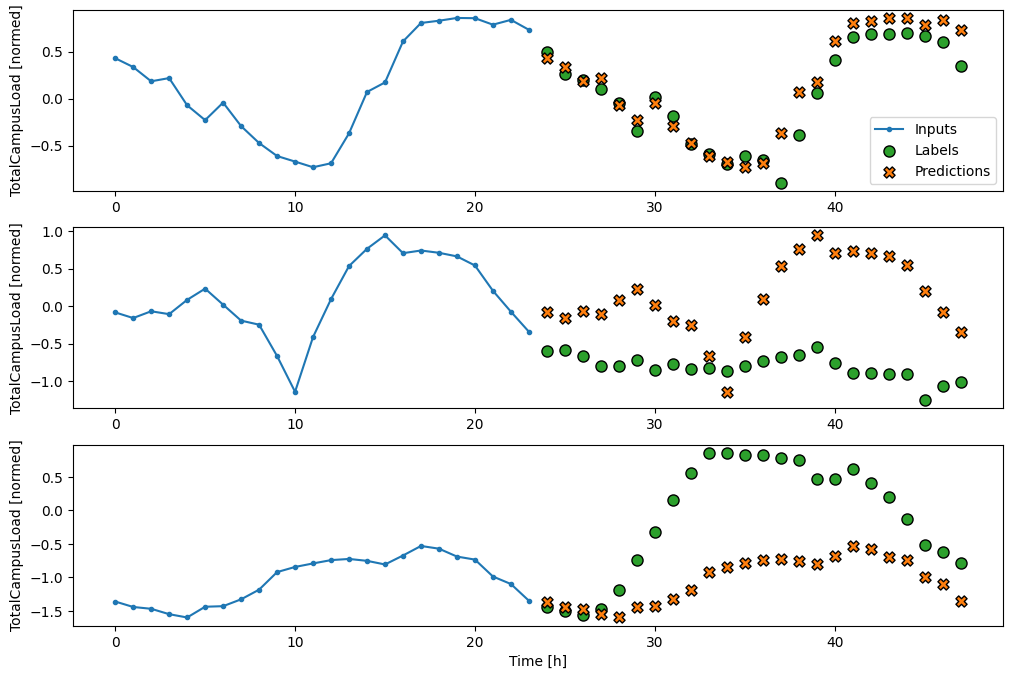
When plotting data for window size we considered random three data points in the training data set. As we predicting day ahead demand data window size is 48 data points. It includes 24 input data and 24 predicted data points combined with labels. Label data points means actual data points of forecasted data points. In the model we have plotted this for repeat baseline model and CNN models. For plotting we have considered Total Campus Load data set.

Figure 5.1: Plot for baseline model-Repeat for window size

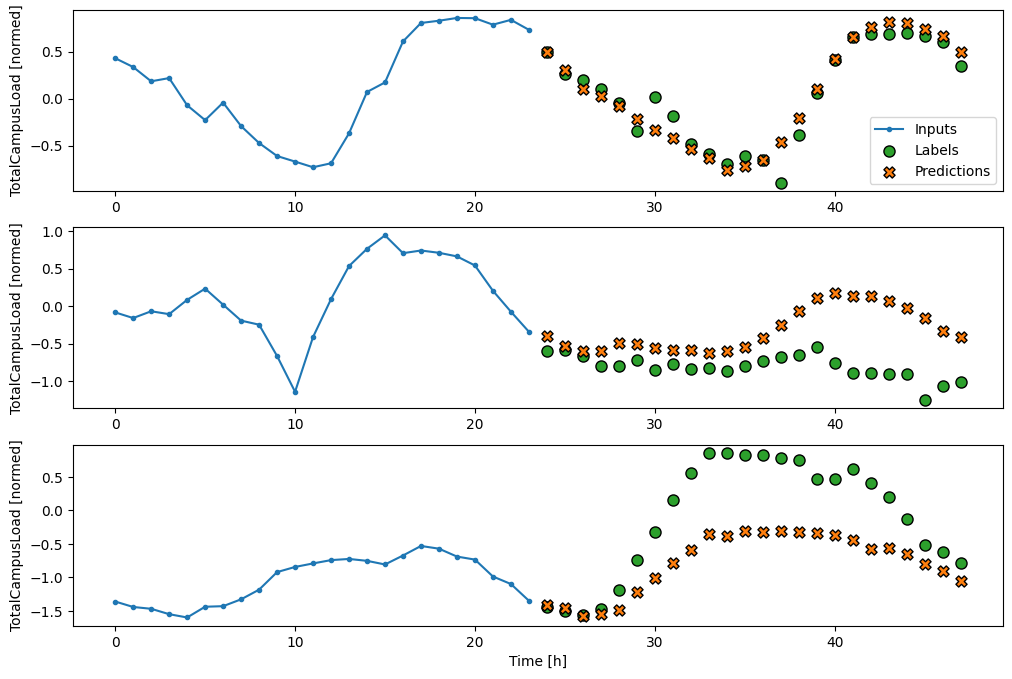
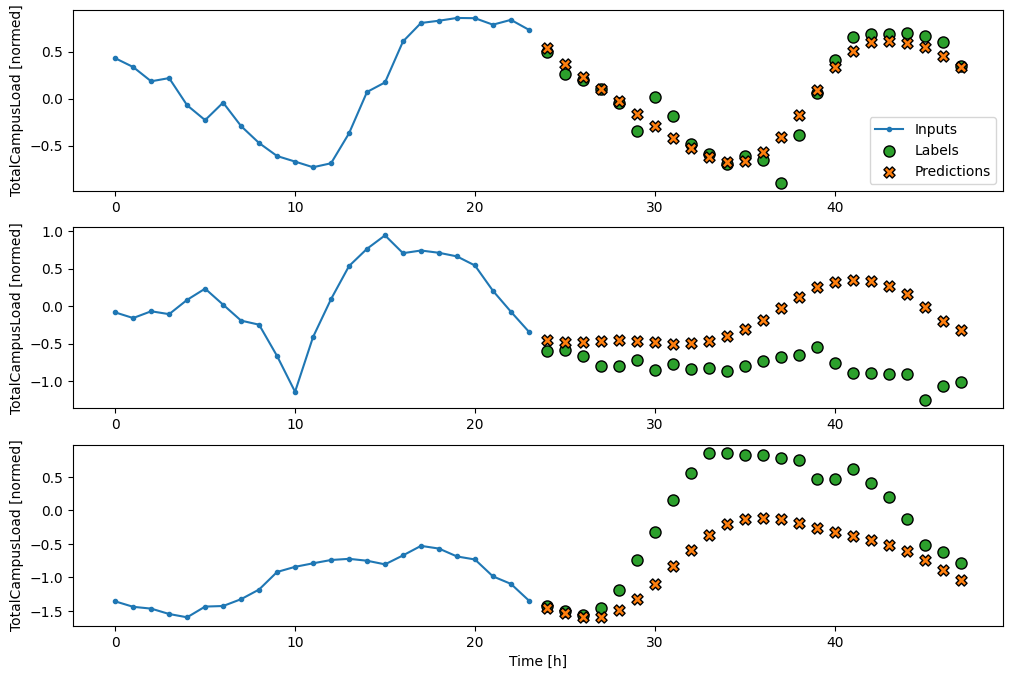
Figure 5.1: shows window size plot for repeat baseline model. Here we can identify that forecasted data are taken as previous day values and variation of each data point can be seen clearly.

Figure 5.2: Plot for “CNN” model for window size

This is plot for “CNN” model this model contains with dropout layer than other “1D CNN” model. Drop out layer is used in regularization. According to this plot we can see that difference between predicted data and actual data are less and most of the data points are same. Other CNN model which doesn’t include dropout layer also gives same results as other model. Following Figure 5.3 shows the plot for “1D CNN” model.

According to these plots we can visualize CNN models gives better fitting forecasted data than baseline model. As this plotting has done for training set it doesn’t give an idea about the accuracy of the of the model. Training set usually used to train model plotting its output we can get some idea about the model variation but to have full understanding about model variation and outputs this model has to be run with an unseen data set. Testing data set is used as this new data set. so, we need to find the actual vs predicted data variation for the test data set. Next, we have plotted actual vs predicted data for testing data set. This plotting has done for all testing data set of Total campus load data.

Figure 5.3: Plot for “1D CNN” model for window size

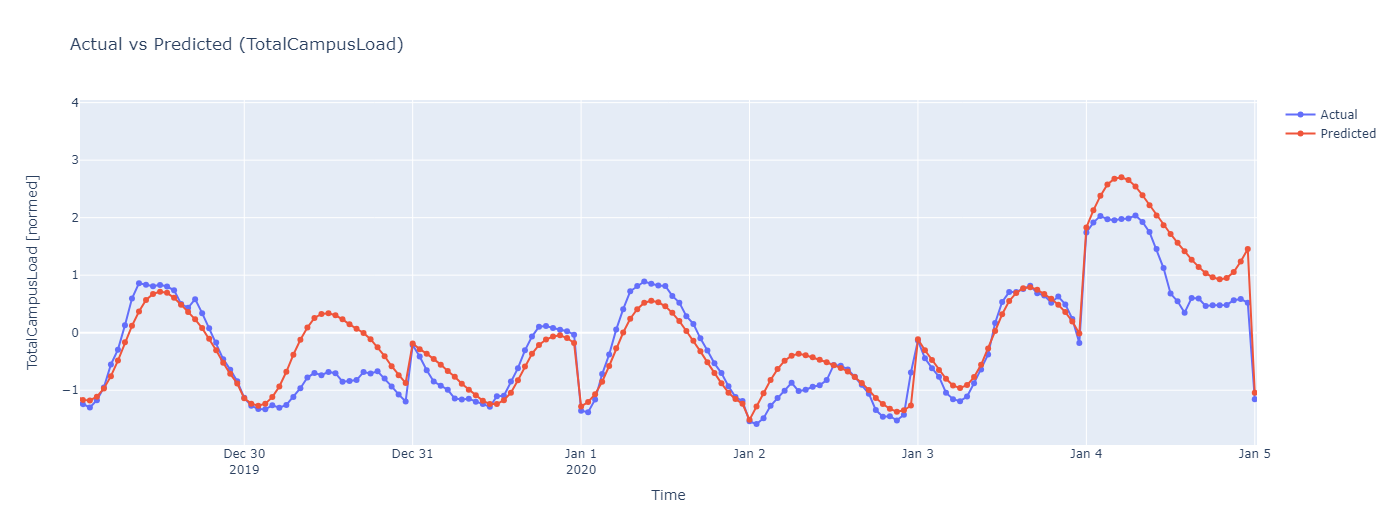
Following figure show the plot between actual vs predicted data of test data set for a week.

Figure 5.4: Actual vs predicted data of test data set for a week

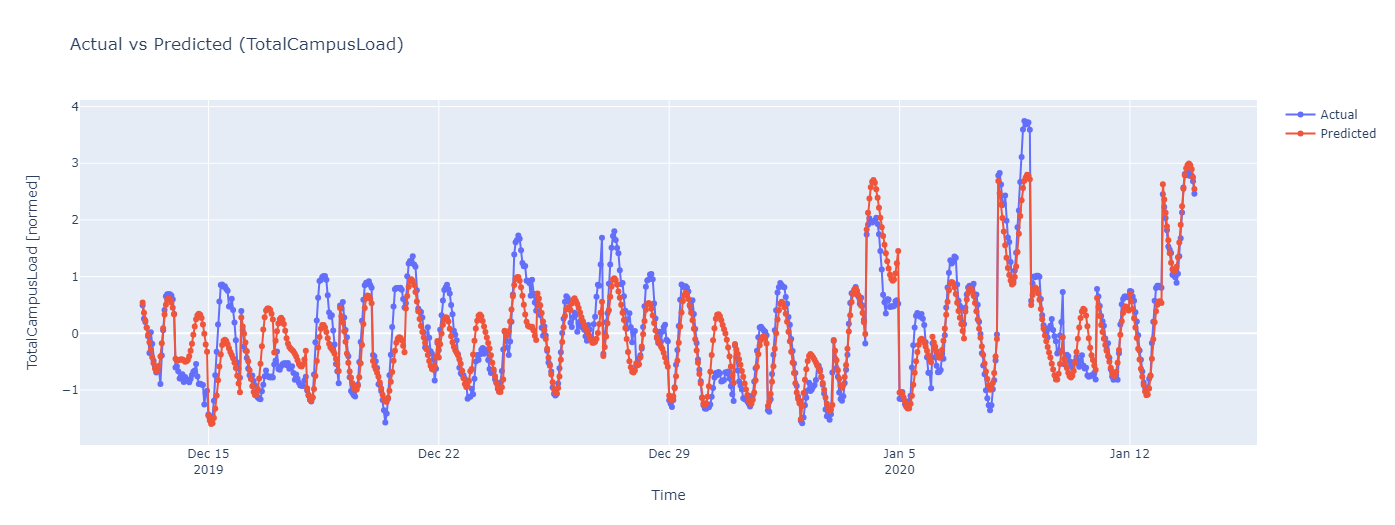
This plot shows output for a week time. There we can clearly see a pattern in each day demand is high in peak times and it get reduce at off peak times. Also, model has cached that pattern and it predict the output most accurate way. According to these visualizations we can decide that CNN model has identified the patterns of input data and it give the most accurate prediction. As this done for test data set it gives idea that model works on unseen data set accurately. It can be more clarify from the plot for all test data set.

Figure 5.5: Actual vs predicted for test data set

This Figure 5.5 shows actual vs predicted data variation for all test data set. Model has identified the pattern and it has given better output according to that.

## Performance evaluation

Then we have to do the performance evaluation of the model. For that we have calculated some matrices by comparing these values we can get better idea about the model. This research project use Mean square error (MSE) as the loss function. There it calculates average squared difference between the predicted values and the actual values. By plotting training loss and validation loss over epochs we can get idea about underfitting and overfitting of the model. For CNN model this plot is shown in following Figure 5.6 and Figure 5.7

If there is overfitting in the model this can be seen by the gap between training and validation losses, if the gap is increasing over epoch model becomes overfit. Underfitting can be identified from the values of the training and validation losses. If the values are higher than model is underfit so loss remain higher with the epoch.

Considering this model gap between training and validation losses are reducing over epochs this means model is not overfit. And also, in “CNN” model we are using dropout layer this also prevent overfitting the model by doing regularization technique that randomly sets a fraction of input units to zero during training. “1D CNN” model doesn’t include this dropout layer but, in that plot, also we cannot see big gap between training and validation losses. It shows in Figure 5.7. Both of these plots’ loss values doesn’t have higher values which means model are not underfit.

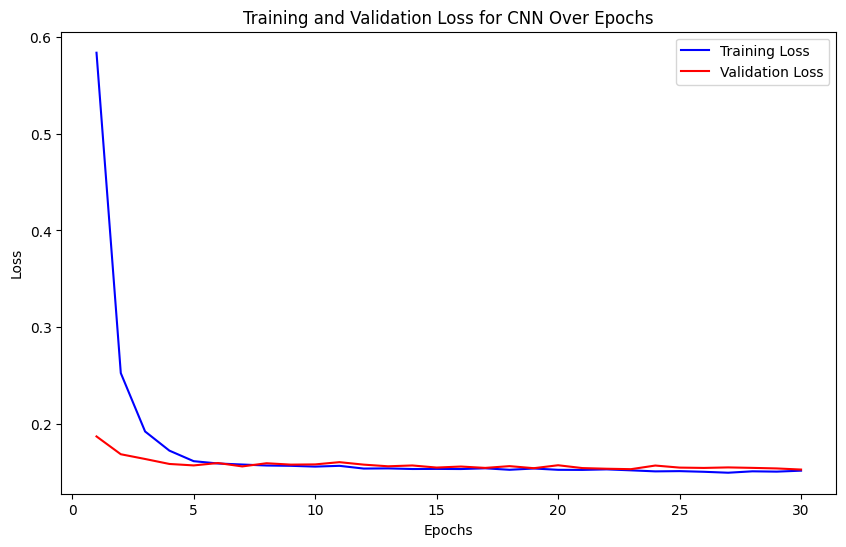
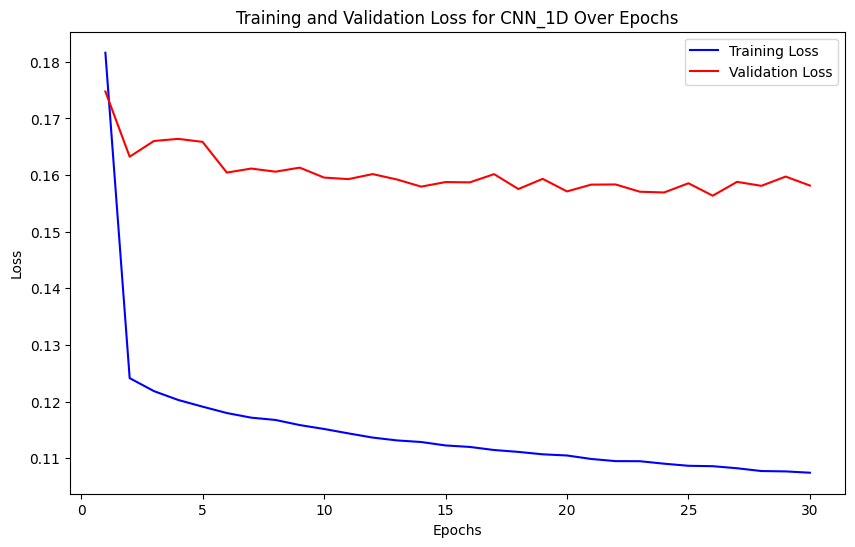
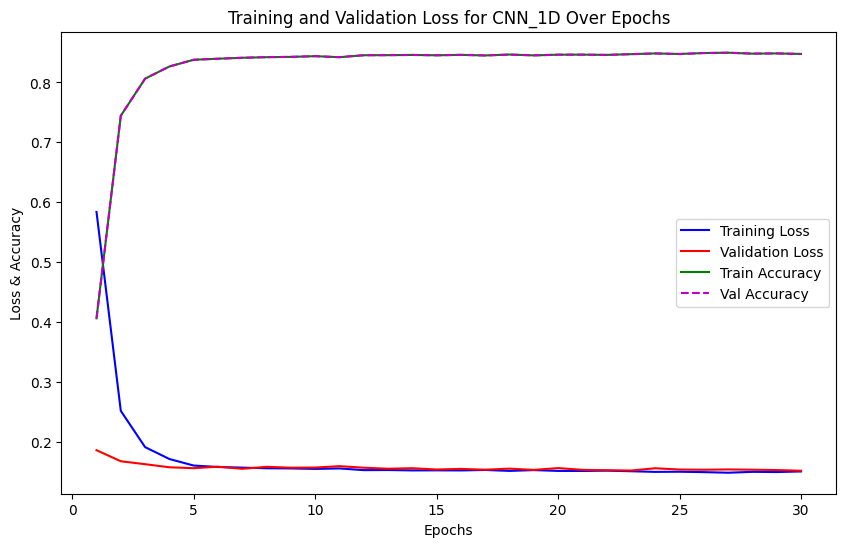
Figure 5.6: Training and validation loss over epochs for “CNN” model

Figure 5.7: Training and validation loss over epochs for “1D CNN” model

This variation of training and validation losses can be seen clearly in above figures. Next, we can check the accuracy of the model over epochs. To perform accuracy, we are using R squared value for each epoch. This plotting has done for both training and validation sets.

Figure 5.8: Training and validation accuracy over epochs for “CNN” model

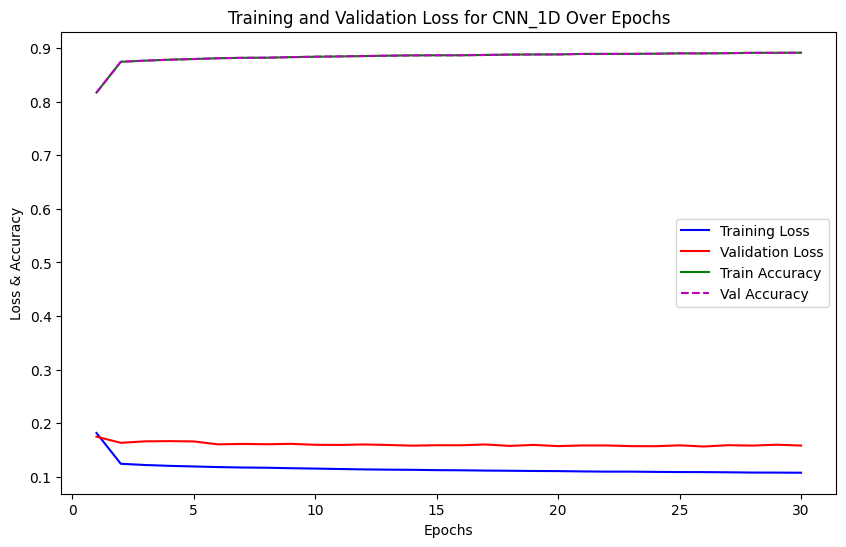
Here we can see that accuracy increase with the epoch and both training and validation accuracy values are almost same. So, we can decide that with the loss reduction accuracy has been increased. For the “1D CNN” model also we can see same kind of results.

Figure 5.9: Training and validation accuracy over epochs for “1D CNN” model

Then let’s consider the matrices that have been calculated. As mentioned in this model we are using Mean squared error (MSE), Mean average error (MAE) and R squared value for matrices. These values are calculated for training, validation and testing sets separately. According to values of these matrices we can decide which model gives the better performance.

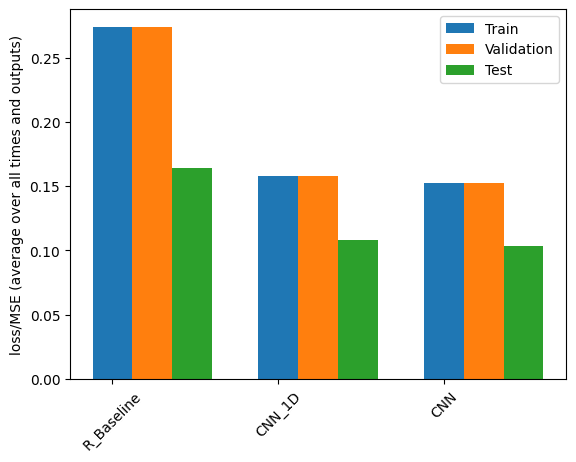
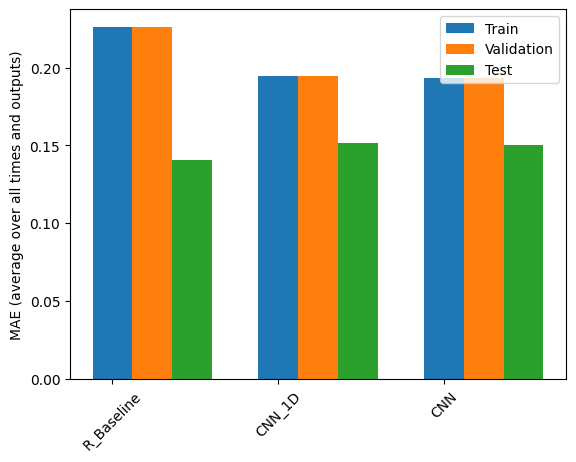
Such as having lover MSE and MAE value indicates that model has better accuracy than other models. R squared values is used to identify how well model values are fit to actual values. This can also take as accuracy measuring factor. So, we have visualized both MSE and MAE value variation for these models considering training validation and test sets separately.

Figure 5.10: MSE value (Average over all times and outputs)

We can clearly see that baseline model have higher values than CNN models so its accuracy is less than CNN models. Considering values of the MSE values they are as follows.

Here for both models’ values vary in same range. So, we can suggest that both models give accurate predictions than baseline model.

Figure 5.11: MAE values (Average over all times and outputs)

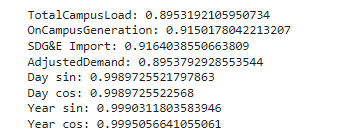
As this graph we can clearly see MAE values of baseline model is higher than other models. Values of CNN model are as follows.

Same as for MSE values MAE values for both CNN models vary in same range and their values are less than baseline model. This can suggest that CNN model give better accuracy than baseline model.

For testing data set MSE and MAE values are less than training and validation sets, which means as this model is works good for new, unseen data set. Which implies the model has been well generalized.

Then considering R squared value, this gives idea about how well the forecasted data fit to the actual values. In other words, model fitness or What percentage of data can be explained using the model. For this models R squared values are as follows.



According to these values we can see that baseline model can explain around 78% of the data and CNN models explain around 87% of data. This implies CNN model is more accurate in forecasting demand data. Also, in this model we have checked R squared value for each feature in data set. There values as follows.

These values imply around 89% of each model can be explained from the CNN model. So, we can come to a conclusion CNN model gives more accurate forecasted data and it explain around 88% of the data.

# Conclusion

Aim of this research project is to forecast energy demand of microgrid using machine learning techniques and use that data in optimizing microgrid operations. Following that we have implemented CNN model to forecast energy demand data. To compare these values and to make benchmark we implemented baseline model which takes forecasted data as previous day inputs.

Implementing CNN model for energy demand forecasting in microgrids offers several advantages. First and foremost, CNNs excel in extracting pertinent temporal features from sequential data. This capability allows the model to effectively capture intricate patterns and variations in energy consumption over time, providing a robust foundation for accurate forecasting. Additionally, the convolutional operations within CNNs enable both local and global pattern recognition. This means the model can discern short-term fluctuations and long-term trends in energy demand, enhancing its ability to adapt to different temporal contexts

Before implementing CNN model, we have done data preprocessing. There we have done handling missing values, handling outliers, handling Nan values, splitting data into training, validation and testing sets and normalizing data to reduce computational power. In the next stage we build CNN model using Tensorflow. There we have used data windowing method and implemented packaged training function to train the model. Within that calculated loss function MSE, MAE and R squared values to evaluate the performance of each model. In model evaluation we have done plotting actual vs predicted data to visualize the output and get idea about model variations.

Optimizing microgrid operation is crucial step that will impact in performance of microgrid. In the research project we did an investigation on how forecasted data can be deploy in optimizing microgrid operation. There we identified different operating process of microgrid and optimization algorithms can be used in that operating process. Then investigated approaches of forecasted data on optimization. There we discussed about importance of forecasted data in optimization of microgrid, how can they be applied in operating process. Results of this research project proved CNN model gives better accuracy than baseline model. And CNN model is explained around 88% of the data set.

We can come to conclusion that CNN model gives better results in forecasting energy demand of a microgrid. So, CNN this implemented CNN model can be used to forecast energy demand. And if we can run this model to a localized data set or new data set. Then comparing these values will get idea

Furthermore, the flexibility and adaptability of CNNs make them well-suited for accommodating varying time scales inherent in the dynamic nature of energy consumption within microgrids. Whether capturing rapid, short-term changes or addressing long-term shifts, CNNs can adjust to different temporal contexts, providing a versatile tool for forecasting tasks. With this model there are more future works that can be done.

This Future work should focus on integrating external factors such as weather and events for a more comprehensive understanding. Multivariate time series analysis, involving additional features, can enhance the model's grasp of influencing factors. Systematic hyperparameter tuning and exploration of ensemble methods provide opportunities to optimize performance and increase robustness. Improving explain ability ensures transparency in the decision-making process, fostering trust. Dynamic adaptation to changing conditions and real-time implementation are essential for responsiveness. Comparative studies with other models aid in selecting the most suitable approach. Scalability assessment across diverse microgrid scenarios and real-world validation contribute to evaluating the model's generalization and practical applicability. Collaboration with industry partners for real-world implementation will provide crucial insights into operational efficacy and potential challenges.

# References

[1] A. Xu, “Pecan: A Novel Approach to Energy Supply and Demand Forecasting in a Photovoltaic Microgrid,” *CJSJ*, [Online]. Available: https://scholar.google.com/scholar?hl=en&as\_sdt=0%2C5&q=Pecan%3A+A+Novel+Approach+to+Energy+Supply+and+Demand+Forecasting+in+a+Photovoltaic+Microgrid&btnG=

[2] J. Silvente, G. M. Kopanos, E. N. Pistikopoulos, and A. Espuña, “A rolling horizon optimization framework for the simultaneous energy supply and demand planning in microgrids,” *Applied Energy*, vol. 155, pp. 485–501, Oct. 2015, doi: 10.1016/j.apenergy.2015.05.090.

[3] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, and C. Weinhardt, “Designing microgrid energy markets,” *Applied Energy*, vol. 210, pp. 870–880, Jan. 2018, doi: 10.1016/j.apenergy.2017.06.054.

[4] M. Slowik and W. Urban, “Machine Learning Short-Term Energy Consumption Forecasting for Microgrids in a Manufacturing Plant,” *Energies*, vol. 15, no. 9, p. 3382, May 2022, doi: 10.3390/en15093382.

[5] S. N. V. B. Rao *et al.*, “Day-Ahead Load Demand Forecasting in Urban Community Cluster Microgrids Using Machine Learning Methods,” *Energies*, vol. 15, no. 17, p. 6124, Aug. 2022, doi: 10.3390/en15176124.

[6] B. Dey, S. Misra, and F. P. Garcia Marquez, “Microgrid system energy management with demand response program for clean and economical operation,” *Applied Energy*, vol. 334, p. 120717, Mar. 2023, doi: 10.1016/j.apenergy.2023.120717.

[7] G. Oğcu, O. F. Demirel, and S. Zaim, “Forecasting Electricity Consumption with Neural Networks and Support Vector Regression,” *Procedia - Social and Behavioral Sciences*, vol. 58, pp. 1576–1585, Oct. 2012, doi: 10.1016/j.sbspro.2012.09.1144.

[8] F. Ünal, A. Almalaq, and S. Ekici, “A Novel Load Forecasting Approach Based on Smart Meter Data Using Advance Preprocessing and Hybrid Deep Learning,” *Applied Sciences*, vol. 11, no. 6, p. 2742, Mar. 2021, doi: 10.3390/app11062742.

[9] E. Mele, “A Review of Machine Learning Algorithms Used for Load Forecasting at Microgrid Level,” in *Proceedings of the International Scientific Conference - Sinteza 2019*, Novi Sad, Serbia: Singidunum University, 2019, pp. 452–458. doi: 10.15308/Sinteza-2019-452-458.

[10] M. S. Khayaty, A. Movludiazar, R. Fotouhi, and M. K. Sheikh-El-Eslami, “Intelligent Microgrid Energy Management System Based on Deep Learning Approach,” in *2021 11th Smart Grid Conference (SGC)*, Tabriz, Iran, Islamic Republic of: IEEE, Dec. 2021, pp. 1–7. doi: 10.1109/SGC54087.2021.9664022.

[11] S. Aslam, H. Herodotou, S. M. Mohsin, N. Javaid, N. Ashraf, and S. Aslam, “A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids,” *Renewable and Sustainable Energy Reviews*, vol. 144, p. 110992, Jul. 2021, doi: 10.1016/j.rser.2021.110992.

[12] J. M. Aguiar-Pérez and M. Á. Pérez-Juárez, “An Insight of Deep Learning Based Demand Forecasting in Smart Grids,” *Sensors*, vol. 23, no. 3, p. 1467, Jan. 2023, doi: 10.3390/s23031467.

[13] S. Aslam, H. Herodotou, N. Ayub, and S. M. Mohsin, “Deep Learning Based Techniques to Enhance the Performance of Microgrids: A Review,” in *2019 International Conference on Frontiers of Information Technology (FIT)*, Dec. 2019, pp. 116–1165. doi: 10.1109/FIT47737.2019.00031.

[14] A. Rahman, V. Srikumar, and A. Smith, “Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks,” *Applied Energy*, vol. 212, pp. 372–385, Feb. 2018, doi: 10.1016/j.apenergy.2017.12.051.

[15] Xishuang Dong, Lijun Qian, and Lei Huang, “Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach,” *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 119–125, Feb. 2017, doi: 10.1109/BIGCOMP.2017.7881726.

[16] A. J. Del Real, F. Dorado, and J. Durán, “Energy Demand Forecasting Using Deep Learning: Applications for the French Grid,” *Energies*, vol. 13, no. 9, p. 2242, May 2020, doi: 10.3390/en13092242.

[17] D. A. Widodo, N. Iksan, E. D. Udayanti, and Djuniadi, “Renewable energy power generation forecasting using deep learning method,” *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 700, no. 1, p. 012026, Mar. 2021, doi: 10.1088/1755-1315/700/1/012026.

[18] S. Silwal, C. Mullican, Y.-A. Chen, A. Ghosh, J. Dilliott, and J. Kleissl, “Open-source multi-year power generation, consumption, and storage data in a microgrid,” *Journal of Renewable and Sustainable Energy*, vol. 13, no. 2, p. 025301, Mar. 2021, doi: 10.1063/5.0038650.

# Appendix