

**CIS7016**

**Research Methods for Technology Dissertation**

**Time-series Forecasting (ML) for Energy Management System**

**Module Leader: Dr Issam Damaj**

**MSc Programme: Advanced Computer Science**

**Student Name: - Ajay Reddy Sykam**

**Student ID: ST 2024657**

Contents

[CHAPTER 1: INTRODUCTION 5](#_Toc141715966)

[1.1- The Introduction 5](#_Toc141715967)

[1.2- Research Motivation: 6](#_Toc141715968)

[1.3- Research Aim: 7](#_Toc141715969)

[1.4- Research Objectives: 7](#_Toc141715970)

[1.5- Research Questions: 8](#_Toc141715971)

[1.6- Summary: 8](#_Toc141715972)

[CHAPTER 2: LITERATURE REVIEW 9](#_Toc141715973)

[2.1- Existing Literature: 9](#_Toc141715974)

[2.2- Literature on Machine Learning Models and Techniques: 11](#_Toc141715975)

[2.2.1- Time series Forecasting: 11](#_Toc141715976)

[2.2.3- Linear Regression 12](#_Toc141715977)

[2.2.4- LASSO Regression 12](#_Toc141715978)

[2.2.5- K-Nearest Neighbors 12](#_Toc141715979)

[2.3- DEEP LEARNING MODELS 13](#_Toc141715980)

[2.3.1- Neural Networks: 13](#_Toc141715981)

[2.3.2- Long Short-Term Memory Networks 14](#_Toc141715982)

[2.4- PERFORMANCE METRICS 15](#_Toc141715983)

[CHAPTER 3: METHODOLOGY 17](#_Toc141715984)

[3.1- The Research Onion Framework: 17](#_Toc141715985)

[3.2- CRISP-DM Methodology 19](#_Toc141715986)

[3.3- Standardization: 21](#_Toc141715987)

[3.4- Normalization 21](#_Toc141715988)

[3.6- Multiple Train-Test Splits (applied to Time Series) 22](#_Toc141715989)

[3.7- Project Architecture: 23](#_Toc141715990)

[3.8- Data Collection 24](#_Toc141715991)

[CHAPTER 4: DATA ANALYSIS & PRE-PROCESSING 25](#_Toc141715992)

[4.1- Descriptive Statistics 25](#_Toc141715993)

[4.2- Missing Values Treatment: 27](#_Toc141715994)

[4.3- Convert columns to appropriate data types (Numeric) 27](#_Toc141715995)

[4.4- Data Visualization: 28](#_Toc141715996)

[4.4.1- EDA -Line Plot of Global Active Power over Time: 28](#_Toc141715997)

[4.4.2- Histogram of Global Intensity: 29](#_Toc141715998)

[4.4.3- Box Plot of Sub-metering Measurements: 30](#_Toc141715999)

[4.4.4- Scatter Plot: 31](#_Toc141716000)

[4.4.5- Outlier Detection (Box – Plot) 32](#_Toc141716001)

[4.4.6- Box plot of yearly global active power 33](#_Toc141716002)

[4.4.7- Probability distribution Plot- Global Active Power Distribution 34](#_Toc141716003)

[4.4.8- Plotting mean grouped by year,month and day 34](#_Toc141716004)

[CHAPTER 5: MODELLING RESULTS & DISCUSSION 36](#_Toc141716005)

[CHAPTER 6: CONCLUSION, LIMITATIONS & FUTURE WORK 43](#_Toc141716006)

[6.1- Conclusion: 43](#_Toc141716007)

[6.2- Future work: 44](#_Toc141716008)

[6.3- Limitations, Ethical Implementations and Challenges: 44](#_Toc141716009)

[REFERENCES: 46](#_Toc141716010)

APPENDIX 47

Figure 1 – An example of a Perceptron

Figure 3: LSTM memory block schema

Figure 4: Research Onion Framework

Figure 5: Crisp-Dm Methodology

Figure 6: Train & Test Split Example

Figure 7: Mutliple Train & Test Split Example

Figure 8: Proposed architecture

Figure 9: Descriptive Statistics

Figure 10: Info of the Dataset

Figure 11: Data Types after conversion

Figure 12: Line Plot for Global Active Power

Figure 13: Histogram for distribution of Global Intensity

Figure 14: Box-Plot for analyzing sub-metering

Figure 15: Scatter Plot Distribution

Figure 16: Distribution Plot for voltage

Figure 17: Outlier Detection

Figure 18: Box-plot for yearly active Power

Figure 19: Probability Distribution plot

Figure 20: Line plot to visualize the mean over time

# CHAPTER 1: INTRODUCTION

## **The Introduction**

The global energy landscape heavily relies on buildings, as they account for a significant portion of energy consumption. Statistics indicate that buildings are responsible for a substantial 40% of the energy produced worldwide. In the European context, buildings contribute to approximately 30% of the total energy consumption available for utilization (Bernardo, 2015). Given this scenario, enhancing the energy efficiency of buildings holds tremendous potential for mitigating global energy demands (Pisello, Bobker, & Cotana, 2012). Any improvement in building energy efficiency can have a significant impact on reducing global energy requirements. Considering the long lifespan of buildings, it becomes crucial to optimize their energy efficiency, and one promising avenue for achieving substantial reductions in energy requirements is through the optimization of HVAC systems. These systems stand out as the biggest offenders in buildings' energy consumption, accounting for a staggering 76% of the total energy usage in European countries (Oldewurtel et al., 2012)." Improving energy efficiency in buildings has a key role to play in achieving the ambitious goal of carbon neutrality by 2050, set out in the European Green Deal. There are a number of ways to optimize HVAC systems, including:

* Selecting the right equipment
* Properly sizing the equipment
* Maintaining the equipment regularly
* Using energy-efficient controls

Although most buildings are designed for human habitation, a significant number are designated for commercial, industrial, and service purposes. Excessive energy consumption in commercial buildings is linked to a lack of an energy management system, as well as a variety of inefficient user behaviors, insulation problems, and intensive equipment use. Given that people spend almost 90% of their time in buildings, it is critical to provide environmental comfort. However, the improvement in indoor well-being is directly linked to an increase in energy consumption. This need can be met by an energy management system that can monitor, control, and optimize building services (Shaikh et al., 2013). To be economically viable, an energy management system must be based on inexpensive technology and equipment. Additionally, it must incorporate intelligence into the control and automation components of the system. Predictive models can be created using machine learning techniques to help the system anticipate environmental conditions and prepare for them ahead of time, preventing unnecessary energy consumption. The development of healthy buildings is a growing field that aims to create a healthy and comfortable environment for building occupants while saving energy.

Machine Learning is a critical component of energy management systems (EMS) by enabling accurate prediction of future energy consumption patterns. It is used to predict future energy demand, which can be used to optimize the operation of power plants, transmission and distribution networks, and other components of the energy system. As the demand for efficient energy utilization and sustainability intensifies, the deployment of Machine Learning (ML) techniques in EMS has gained significant attention. The primary objective is to enhance the energy efficiency of buildings and facilities by leveraging ML algorithms to forecast energy consumption, demand, and generation with high accuracy and reliability. The research will encompass comprehensive exploration and evaluation of diverse ML techniques, including but not limited to deep learning models, recurrent neural networks, and ensemble methods. Additionally, the investigation will consider various factors influencing energy consumption patterns, such as weather conditions, occupancy behavior, and building characteristics. Machine learning (ML) is a powerful tool for time-series forecasting. ML algorithms can learn from historical data to identify patterns and trends that can be used to predict future values. ML algorithms have been shown to be very effective at forecasting energy demand, and they are increasingly being used in EMS around the world."

## **Research Motivation:**

The motivation behind this research from the critical need for effective energy management in the face of increased energy demand and environmental concerns. Time-series forecasting, combined with Machine Learning (ML) techniques, offers a promising approach to optimizing energy management systems (EMS) in buildings and facilities. This technique has been used in a variety of applications, including energy management systems. In an energy management system, time-series forecasting can be used to predict energy consumption, which can then be used to optimize the operation of the system. This can lead to reduced energy costs and improved energy efficiency. Accurate forecasting of energy consumption, demand, and generation can enable proactive decision-making, efficient resource allocation, and the implementation of demand response strategies. By developing advanced ML-based time-series forecasting models specifically tailored for EMS applications, this research aims to contribute to the development of intelligent energy management systems that maximize energy efficiency, reduce costs, and mitigate environmental impact.

## **Research Aim:**

This project's primary goal is to develop predictive algorithms that will assist the environmental occurrences and being ready for them in advance, minimizing needless energy use.

## **Research Objectives:**

1. To Conduct a thorough literature review by analyzing existing research papers in the energy management sector, specifically focusing on the advantages of utilizing machine learning algorithms in this domain.
2. To Identify and select a suitable dataset for analysis by applying various statistical techniques to determine the most effective algorithms for addressing the problem statement.
3. To Determine the most appropriate methodology, tools, and libraries for data analysis and modeling, ensuring accurate and efficient processing of the dataset.
4. To Evaluate the performance of the developed models using a range of evaluation metrics, aiming to achieve optimal accuracy.

Time series forecasting is a challenging task, but it can be made easier with the help of machine learning algorithms. ML algorithms can handle complex time series data with intricate, noisy, and interdependent relationships across multiple input variables. Deep learning is a type of ML algorithm that has shown promise for time series forecasting. DL algorithms can autonomously learn salient features from time series data, which eliminates the need for manual feature engineering.

## **Research Questions:**

In this context, from a Data Science perspective, the main Research Questions of the project are:

**RQ1:** How machine learning algorithms be implemented for time series forecasting in energy management systems?

**RQ2:** what are the effective data processing techniques ensuring the data quality for time series data?

**RQ3**: What are the key differences between conventional machine learning methods and deep learning techniques for time series forecasting in energy management systems?

**RQ4**: How can different evaluation metrics be utilized to assess and compare the performance of developed models?

**RQ5:** What are the limitations, ethical implications, and challenges of implementing machine learning algorithms?

## **Summary:**

Buildings account for a significant portion of global energy consumption, and HVAC systems are responsible for a large portion of that energy use. Time-series forecasting is a critical component of energy management systems (EMS), and machine learning (ML) techniques can be used to improve the accuracy of these forecasts. Commercial buildings also contribute to excessive energy consumption due to a lack of energy management systems, inefficient user behaviors, insulation problems, and intensive equipment use. An energy management system that monitors, controls, and optimizes building services can address these issues. This research has the potential to significantly improve the energy efficiency of buildings and facilities, reduce energy costs, and mitigate environmental impact.

# CHAPTER 2: LITERATURE REVIEW

## **2.1- Existing Literature:**

The research paper by Li et al. (2019) proposes a short-term load forecasting model based on recurrent neural networks (RNNs). The study utilizes historical energy consumption data and weather information as input features for the RNN model. The results demonstrate that the proposed model outperforms traditional statistical methods in accurately predicting short-term energy loads. The authors highlight the importance of incorporating weather data in load forecasting models to capture the impact of external factors on energy consumption. In this paper, the author Wang et al. (2017) investigated the application of support vector machines (SVMs) for energy demand forecasting in a smart grid environment. The study compares the performance of different SVM configurations and evaluates their accuracy in predicting future energy demand. The findings indicate that SVM-based models exhibit better forecasting accuracy than traditional statistical methods. The authors emphasize the potential of SVMs in assisting energy management systems to optimize resource allocation and improve energy efficiency.

Another research by Chen et al. (2018) proposes a hybrid forecasting model for wind power generation using a combination of genetic algorithms (GAs) and artificial neural networks (ANNs). The study aims to address the challenges posed by the intermittent and unpredictable nature of wind power. The hybrid model optimizes the architecture and parameters of the ANN using GAs to improve forecasting accuracy. The results demonstrate the superiority of the proposed model in capturing complex relationships and improving the reliability of wind power forecasts. Moreover, Zhang et al. (2020) investigate the application of long short-term memory (LSTM) neural networks for time series forecasting in energy management systems. The study utilizes historical energy consumption data and weather information as input features for the LSTM model. The results demonstrate that the LSTM model outperforms traditional forecasting methods in accurately predicting short-term energy consumption. The authors highlight the ability of LSTM networks to capture long-term dependencies and non-linear patterns in energy data.

"A Comparative Study of Machine Learning Techniques for Electricity Load Forecasting" by Khan et al. (2019) evaluates the performance of various machine learning techniques, including artificial neural networks, support vector machines, and random forests, for electricity load forecasting. The research compares the accuracy, robustness, and computational efficiency of these techniques using real-world electricity load datasets. The findings suggest that artificial neural networks achieve the highest forecasting accuracy, followed by support vector machines and random forests. The authors emphasize the importance of selecting appropriate machine learning algorithms based on the specific requirements of energy management systems.

A Review" by Zhao et al. (2018) paper provides an overview of ensemble learning techniques for electricity load forecasting in energy management systems. The study examines various ensemble methods, including bagging, boosting, and stacking, and their application in combining multiple forecasting models. The findings highlight the superior performance of ensemble methods in terms of accuracy and robustness compared to individual models. The authors emphasize the potential of ensemble learning for enhancing energy forecasting in real-world applications. Explored the use of deep learning techniques, specifically convolutional neural networks (CNNs) and recurrent neural networks, for solar power forecasting in energy management systems. The research compares the performance of CNNs and RNNs in capturing spatial and temporal patterns in solar irradiance data. The results demonstrate that deep learning models achieve higher accuracy in solar power forecasting compared to traditional methods. The authors highlight the potential of deep learning techniques in improving the integration of solar power into the energy grid (Zhang et al. (2021)). Fan et al. (2020) investigate the application of long short-term memory (LSTM) networks for forecasting electricity price volatility in energy management systems. The study utilizes historical electricity price data and various market indicators as input features for the LSTM model. The results demonstrate that the LSTM model effectively captures the volatility patterns in electricity prices and outperforms traditional econometric models. The authors emphasize the importance of accurate volatility forecasting for risk management and decision-making in energy markets.

The research paper by Liang et al. (2020) proposes a hybrid short-term load forecasting model for buildings using machine learning techniques. The research combines multiple forecasting models, including support vector regression (SVR), artificial neural networks (ANNs), and autoregressive integrated moving averages (ARIMA), to improve forecasting accuracy. The results demonstrate that the hybrid model achieves higher accuracy compared to individual models, enabling better energy management and load scheduling in buildings. The comprehensive review paper research by Wu et al. (2019) examines the state-of-the-art techniques for energy consumption forecasting in smart buildings. The study discusses various machine learning algorithms, including regression models, neural networks, and ensemble methods, and their application in predicting energy consumption patterns. The findings highlight the importance of data preprocessing, feature selection, and model optimization for accurate energy consumption forecasting. The authors provide insights into the challenges and future directions of energy forecasting in smart buildings.

## **2.2- Literature on Machine Learning Models and Techniques:**

### **2.2.1- Time series Forecasting:**

Times Series Analysis can be defined as the systematic approach to answering the mathematical and statistical questions posed by the time correlation introduced by the sampling of adjacent points in time. It is the analysis of experimental data that have been observed at different points in time (Shumway & Stoffer, 2017, p. 1). Time Series Analysis usually involves the study of the form of the data and of the components of a time series (Brownlee, 2018c, p. 11). It can also be defined as the process of taking historical data of a time series, sometimes with additional information, and fitting models to forecast future values. Unlike Time Series Analysis which can be done in retrospect and use “future” information, forecasting models don’t have “future” information available. Everything must be done based only on what has already happened. A predictive model is evaluated by its predictive accuracy. Meanwhile, a descriptive model is assessed by its capability of providing correct causal explanations (Shmueli & Lichtendahl, 2016, p. 19).

**2.2.2- DATA SCIENCE**

Although Data Science is a comprehensive field, most of the attention of the scientific community and organizations is focused on Machine Learning, particularly on predictive analytics. Machine Learning is a kind of Artificial Intelligence that uses algorithms to extract patterns from data. These algorithms can learn from data as they are being trained. In this process of learning, they can improve their performance based on experience. In the end, a refined model is achieved that can be used to predict outcomes from unseen data based on previous learning (Kirk, 2014, p. 2) (Bell, 2015, p. 2).There are many use cases for Machine Learning in the energy and environment sectors, like using predictive analytics to pick the best location for wind farms (Hardesty, 2015) or to analyze pollution data and make predictions about air quality (IBM Research Editorial Staff, 2016). In reality, the convergence of Machine Learning and the Internet of Things leads to possibilities that both technologies alone could never achieve. For instance, Machine Learning allows IoT to provide AI-powered analytics platforms capable of continuous analytics, besides predictive and prescriptive analytics (Ruzicka, Lawrence, Chaudhri, Jacquet, & Smith, 2018).

## **2.2.3- Linear Regression**

It has as positive aspects the fact that is easy to compute and that the coefficients, the weights in the sum of the input variables, are directly interpretable. Linear Regression has a couple of restrictions, though. An input variable cannot be determined from a combination of one or more of the other input variables (collinearity). And the number of observations must be greater than the number of input variables. Otherwise, it is impossible to reach a unique linear combination of the features to represent the target (Kuhn & Johnson, 2013, p. 108). Since Linear Regression has no parameters, it has no way to control model complexity (Müller & Guido, 2016, p. 47).

## **2.2.4- LASSO Regression**

In LASSO regression the constraint restricts the magnitude of the coefficients to be close to or equal to zero. This type of regularization, where the sum of the absolute values of the regression coefficients is penalized, is known as L1 regularization. When the L1 regularization forces coefficients to be zero, in practice, the LASSO regression algorithm is performing a feature selection. Hence, the Selection Operator in the name of the method. LASSO regression achieves both improving the model quality and conducting variable selection by applying L1 regularization (Dinov, 2018, p. 579).

## **2.2.5- K-Nearest Neighbors**

The distance metric used as default is the Minkowski distance, which is a generalization of the Euclidean distance and the Manhattan distance (Brownlee, 2019, p. 80). Since the power parameter for the Minkowski distance is set as 2 by default, the Euclidean distance is applied if not otherwise configured. When the power parameter is set as 1, Manhattan distance is employed. The positive aspects of the K-Nearest Neighbors algorithm are that it is easy to understand and that it achieves reasonable performance scores without many adjustments. These characteristics make it a good baseline method. The downside of the K-Nearest Neighbors algorithm is that it is not fast when the training dataset is very large, even more, because it is an algorithm that requires pre-processing. Other than that, it is an algorithm that doesn’t deal well with a dataset where most values are zeros either (Müller & Guido, 2016, p. 44).

## **2.3- DEEP LEARNING MODELS**

## **2.3.1- Neural Networks:**

Deep Learning and Neural Networks are considered to be in the same field. Deep Learning is a kind of learning based on deep neural networks, networks that have several stacked layers to improve their predictive capability (Skansi, 2018, p. preface V). Neural Networks models mimic the human nervous system’s response to external stimuli. Neural Networks simulate the brain using a network of interconnected nodes, known as neurons, just like nervous system cells. A neuron from a Neural Network can receive input data, compute this data, and send the results to another neuron in the network. Each neuron is accompanied by a weight that defines its computation function. The learning in Neural Networks happens by changing these weights accordingly. The main idea of this stage of Neural Network learning is to modify the weights incrementally whenever a wrong prediction is made (Aggarwal, 2015, p. 326). It is a nonlinear function like the rectifying nonlinearity (or relu) or the tangent hyperbolic (or tanh) that allows the Neural Network to learn much more complicated problems than a linear model (Müller & Guido, 2016, p. 106). Below, is the representation of a single neuron Perceptron. The Perceptron is a simple Neural Network with one or more neurons positioned in just one layer (Aggarwal, 2015, pp. 326–328).

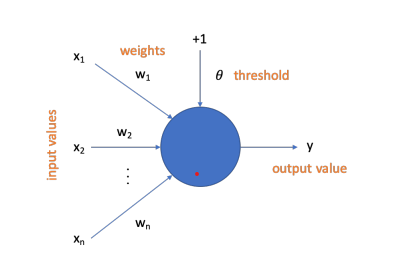


Figure 1 – An example of a Perceptron (Aggarwal, 2015, p. 326).

## **2.3.2- Long Short-Term Memory Networks**

LSTM network solves the problem of very deep Neural Networks not having stable gradients to update the weights. In RNN, where the network architecture is unrolled to promote training through BPTT, it is an even bigger problem. LSTM networks have a new type of architecture that addresses this issue (Brownlee, 2018b, p. 172). By having a mechanism that controls the flow of data, it allows relevant information to pass down a much bigger sequence of nodes without stopping to learn because of the very small values of gradients (Skansi, 2018, p. 143).LSTM networks are distinguished by not having classic neurons. Instead, its computational unit is called a memory block. A memory block differentiates itself from a neuron by having memory and gates. These gates control the input and output of data from the memory block and also the state of the block, if it is activated or not. Each gate has a weight associated with it that is updated during the learning phase and also a sigmoid activation function to control its triggering. The activation function is vital because it adds the conditional factor to the gates and, therefore, to its state and to the input and output of data from the memory block. There are three types of gates: input gate, output gate, and forget gate. While the input and the output gates are responsible for conditionally control the input and output of data from the memory, the forget gate is in charge of discarding data, also conditionally (Brownlee, 2018b, p. 172).

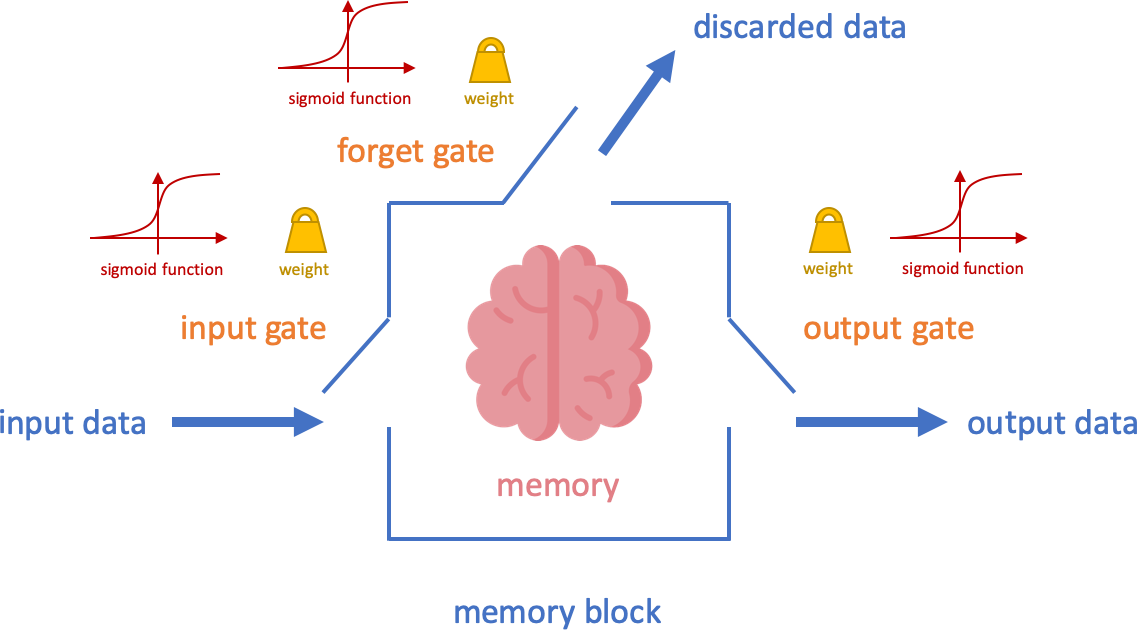


Figure 3: LSTM memory block schema (Source: Skansi, 2018, p. 143)

## **2.4- PERFORMANCE METRICS**

Model evaluation involves assessing the performance of a model in practical scenarios. To conduct a model evaluation, the dataset is typically divided into training and test sets. It is important to avoid using the same data for both training and evaluation purposes to prevent overly optimistic performance estimates. If the training data is used for evaluation, the model is likely to achieve perfect scores, but its predictions may be inaccurate when applied to new data (Brownlee, 2019, p. 57). Thus, model evaluation should be conducted exclusively based on the test set, which consists of unseen data for the model. The testing set is withheld from the model during training, ensuring that it is encountering this data for the first time during evaluation (Brownlee, 2018c, p. 145). In some cases, the dataset may be split into three parts: training, validation, and test sets. In such cases, the training data is utilized for model learning, the validation data helps select the best model and estimate performance parameters, and the test data is solely employed for evaluating the model's performance at the end of the process (Igual & Seguí, 2017, p. 82). A Time Series Forecasting problem is a regression problem. And a regression problem is focused on the prediction of real values. A direct way to evaluate time series forecasts is based on the difference between the predicted values and the expected values. Three common performance metrics for Time Series Forecasting problems are Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error.

**2.4.1- Mean Absolute Error (MAE):** The Mean Absolute Error (MAE) is a performance metric calculated as the average of the absolute differences between the predicted values and the expected values. This measure has as a positive aspect the fact that is simple and that it indicates the magnitude of the error. However, it has as its negative aspect the fact that it does not inform the direction of the error, because the differences are being forced as positive values (Brownlee, 2019, p. 67) (Skiena, 2017, p. 221). The Mean Absolute Error is represented by the following formula:

mae = mean( abs( expected values – predicted values ) )

**2.4.2- Mean Squared Error (MSE):** It is Mean Squared Error (MSE) is a performance metric calculated as the average of the squared differences between predicted values and the expected values. As in MAE, the differences are being forced to become positive values but, this time, by squaring and not by making them absolute. Squaring has the potential side effect of outliers dominating the error statistics. The MSE metric has the benefit of large error values contributing more to worsening the performance score. Therefore, it is an informative measure, especially, for noisy instances (Brownlee, 2019, p. 68) (Skiena, 2017, p. 222).The Mean Squared Error is represented by the following formula:

mse = mean( (expected values – predicted values)ˆ2 )

**2.4.3- Root Mean Squared Error (RMSE):** It is a performance metric that is simply the square root of the Mean Squared Error. The positive aspect of the RMSE is that it is on the same scale as the original values and, therefore, its magnitude is much more interpretable (Skiena, 2017, p. 223).The Root Mean Squared Error is represented by the following formula:

rmse = sqrt(mean((expected values – predicted values)ˆ2 ))

# CHAPTER 3: METHODOLOGY

## **3.1- The Research Onion Framework:**

The Research Onion Framework is a comprehensive model that guides this research in devising and implementing effectively. It provides a methodical research approach, leading the project through the different stages of the research process. We delve into the fundamental constituents of this framework, emphasizing their application within a thesis project. The framework comprises six interconnected layers that considered during this research. These layers include philosophical underpinnings, research approaches, research strategies, time horizons, data collection methodologies, and data analysis techniques (Saunders, M., Lewis, P., & Thornhill, A. (2007).)

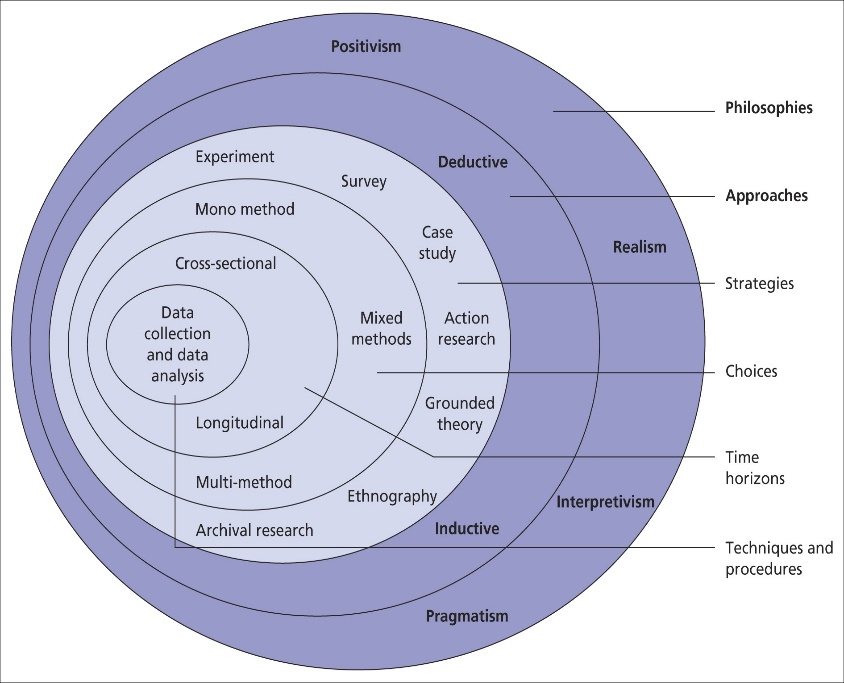


Figure 4: Research Onion Framwork (Source: Saunders,(2007))

1. **Philosophical Underpinnings:** The main of this Framework lies in its philosophical foundations, which underlie the project’s high-level view. In the context of the thesis project on time-series forecasting, a viable philosophical assumption is positivism. Positivism asserts the primacy of objective measurement of phenomena and the use of quantitative methodologies to collect data. This aligns seamlessly with the project's objective of adopting a quantitative research approach to facilitate time-series forecasting.
2. **Research Approach**: Moving to the next layer, the research approach, the project employed a secondary research approach. This approach involves the aggregation and analysis of existing data from a variety of sources.
3. **Research Strategies:** The research strategy, situated within the Research Onion Framework, provides a comprehensive blueprint for executing the project. In this thesis, a quantitative research strategy is applied and it is appropriate as it facilitates the implementation of statistical methods to analyze data and extract meaningful insights. The project employed statistical techniques, such as regression analysis or time-series modeling, to forecast energy consumption patterns.
4. **Time Horizons:** The layer of time horizons considers the duration of the study. In the present thesis project, a temporal framework of two months has been established. This constraint necessitates the project’s judicious allocation of time and resources to collect, analyze, and interpret data within the prescribed temporal confines. It emphasizes the importance of setting realistic goals and adhering to a structured timeline.
5. **Data Collection:** The data collection layer delineates the methodologies employed to accumulate data. The secondary research approach, relied on existing datasets relevant to energy consumption within an energy management system. The dataset includes historical records of energy consumption, meteorological data, and other pertinent variables that influence energy usage. Lastly, the data analysis techniques layer pertains to the means by which the amassed data will be analyzed. In light of the quantitative research approach undertaken in the project, statistical analysis techniques would be applied.

## **3.2- CRISP-DM Methodology**

With regard to the project's methodology, the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework would be appropriate. It provides a structured approach to data mining and predictive modeling, seamlessly aligning with the objectives of time-series forecasting within an energy management system. The methodology would encompass stages such as data understanding, data preparation, modeling, evaluation, and deployment.

These processes can be seen as the different approaches to knowledge searching in big-volume datasets, we use them as a guide that tells us which way to go. It’s an iterative procedure where evaluation measures can be enhanced, mining can be refined, and new data can be integrated and transformed to get different and more suitable results (Wirth, R. (2011)). CRISP-DM is the abbreviation for Cross Industry Standard Process for Data Mining which can be understood as a cross-industry standard process. This methodology brings together the best practices so that DM can be as productive and efficient as possible, analyzing the data to propose improvement models or problem-solving. This is very useful in scenarios of uncertainty when solving business problems. In other words, a step is only started when the previous one has already been subjected to a validation process, which implies changes over time. Therefore, it is a flexible methodology capable of dealing with complex problems involving a large amount of data. The CRISP-DM methodology defines the project life cycle, dividing it into six stages.

1. **Business Understanding:** This phase focuses on understanding the objectives and requirements of the project. Specify the problem, transform it into a business question, and look for all the details about its impact.
2. **Data Understanding:** This step consists of organizing and documenting all the data that is available in four tasks: Collecting initial data, describing data, exploring data, and verifying data quality. At this point, the investigative side must step in, so that the data reveals business problems, solutions, and trends.
3. **Data Preparation:** In this step, a technical part of data analysis is already applied. The choice of the data that will be worked on is made, as the formats and technical questions of the analysis, and how they will be organized to solve the company's problem. Prepares the final dataset for modelling
4. **Modeling:** Based on the objectives identified in the first step, Data Mining techniques are applied to develop several different modeling techniques. First, it will have to be decided which algorithms to try, and after that, generate a test design by splitting the data into training, test, and validation sets, then build the actual models and finally assess the model.

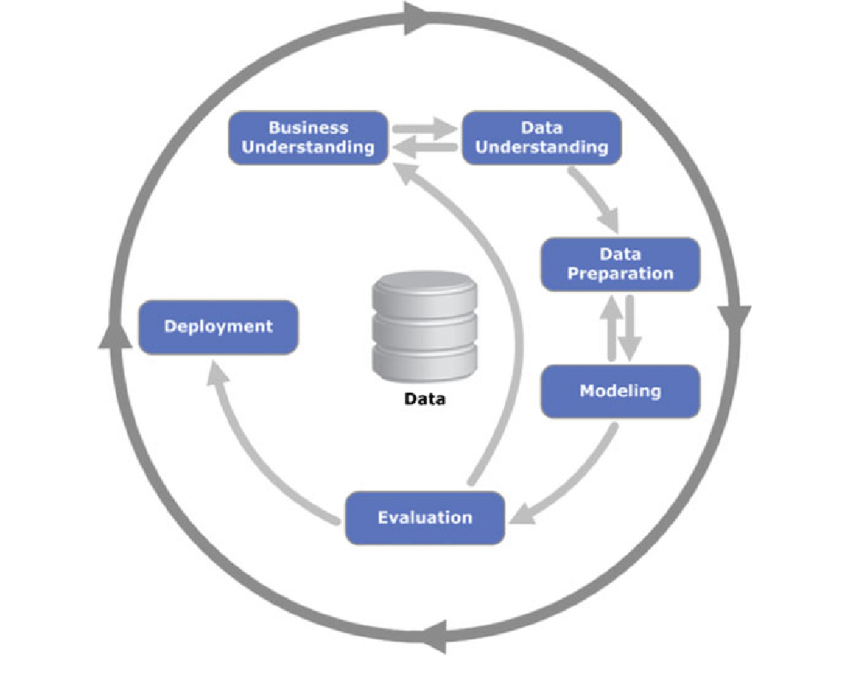


Figure 5: Crisp-Dm Methodology (Source: Wirth, R. (2011)).

1. **Evaluation:** Whereas the model assessment task of the Modelling phase focuses on technical model assessment, the Evaluation phase looks more broadly at which model best meets the business and what to do next.
2. **Deployment:** In the last step, the models are deployed to enable end-users to use the data as a basis for support in the business process. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that the end user understands it

## **3.3- Standardization:**

Standardization (or z-score normalization) is the technique that rescales the data by transforming the mean of the distribution to zero and its standard deviation to one like a standard normal distribution (or standard Gaussian distribution) (Burkov, 2019, p. 39). It is represented by the following formula.

y = (x – mean) / standard deviation

Where mean is calculated as:

mean = sum(x) / count(x)

And the standard deviation is calculated as:

standard deviation = sqrt (sum((x – mean)ˆ2 ) / count(x) )

## **3.4- Normalization**

Normalization (or min-max normalization) is the technique that rescales the data by converting the actual range of values to a standard range of values. Usually, it is done in the interval between 0 and 1 or -1 and 1 (Burkov, 2019, p. 39). It is represented by the following formula:

y = (x – min) / (max-min)

Where min and max stand for the minimum and the maximum values of x in the dataset.

**3.5- Train-Test Split*:***

Train-test split is the simplest method for model evaluation. And, if the dataset is large enough, both train and test splits could incorporate different patterns representing well the problem and the resulting estimate of performance could be consistent. Another positive aspect of the train-test split is that it is a fast method. Train-test split consists of splitting the dataset into two parts: training and test sets. One for training the algorithm and the other to make predictions and compare them to the already known values. That way, it is possible to calculate a performance metric and assess the accuracy of the model.

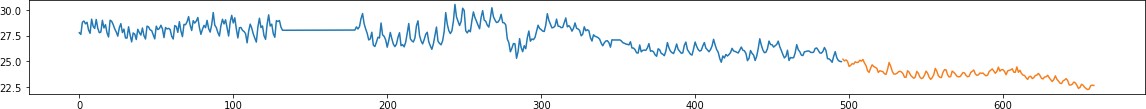


Figure 6: Train & Test Split Example (Source: Brownlee, 2019, p. 58)).

The negative aspect of the train-test split is that differences in training and test sets could end up in dissimilarities in the estimate of performance (Brownlee, 2019, p. 58).

## **3.6- Multiple Train-Test Splits (applied to Time Series)**

Multiple Train-Test Splits consist of repeating the process of splitting the dataset into training and test sets several times. In each iteration, the training set gets larger and the test set remains the same for the sake of comparability among the performance scores obtained (Brownlee, 2018c, pp. 148, 149).

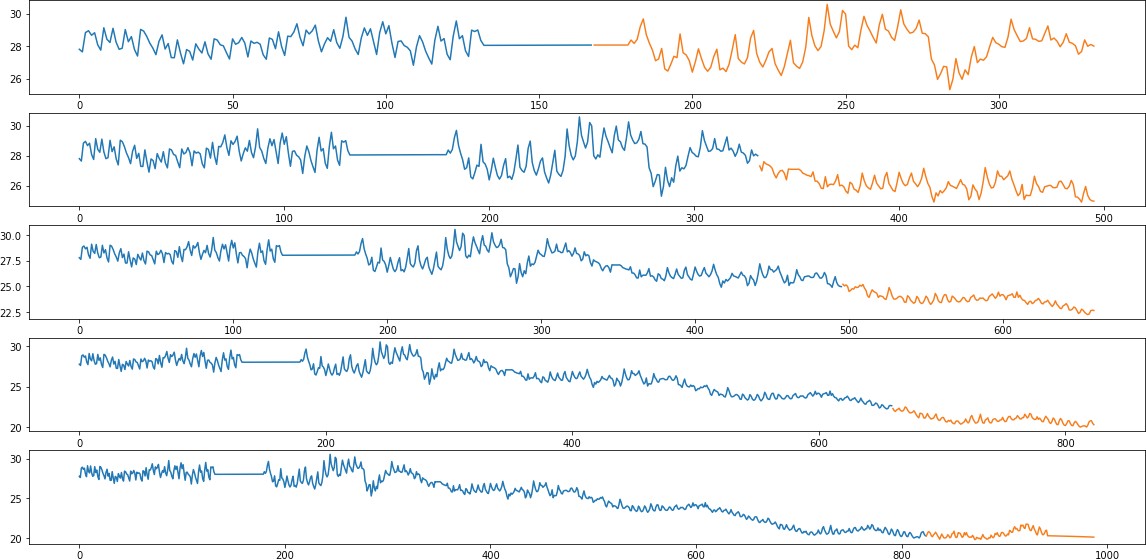


Figure 7: Mutliple Train & Test Split Example (Source: Brownlee, 2019, p. 58)).

It is possible to create multiple splits of a times series by using the class Timeseries Split from Python’s library sci-kit-learn.

training set size = (i \* (samples / (splits + 1))) + samples mod (splits + 1)

Where i is the split iteration, samples are the number of samples, and splits is the number of splits desired. The part samples mod (splits + 1) is the remainder of the division of the number of samples by the number of splits plus one.

test index = samples / (splits + 1)

Where samples are the number of samples and splits is the number of splits wanted.

## **3.7- Project Architecture:**

The proposed architecture of the project is depicted in Figure 8. It initiates with data loading into Python, succeeded by data cleaning and visualization, as stated earlier. The outcomes of these processes are detailed in Chapter 4. Subsequently, the model building and evaluation phases are carried out, and the corresponding results are elaborated in Chapter 5.

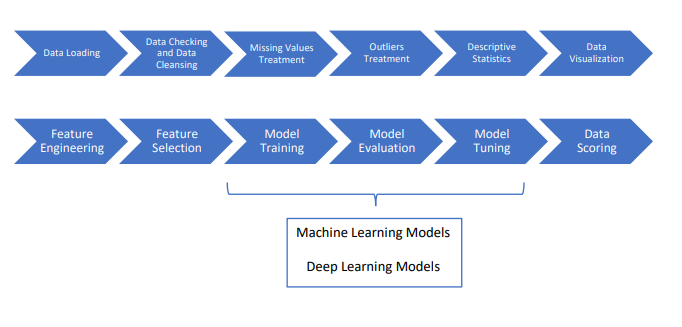


Figure 8: Proposed architecture (Created by Author)

## **3.8- Data Collection**

The data used in this project is a secondary dataset collected from UCI Repository related to energey consumption. The link of the datset mentioned below in the appendix. The provided dataset represents a crucial asset for investigating and scrutinizing the intricate electricity consumption patterns within distinct household units. It encompasses diverse attributes pertaining to electrical utilization, including temporal features such as date and time, along with physical measures such as active power, reactive power, voltage, and current. The dataset spans a substantial temporal extent, which facilitates comprehensive examinations of diurnal and hourly trends, as well as seasonal oscillations, potentially unveiling correlations with extraneous determinants. Practitioners engaged in the domains of energy management, smart grids, and sustainability can harness this dataset to glean profound insights into household energy consumption behavior, thereby formulating efficacious strategies to optimize energy preservation and enhance efficiency. Precisely, the dataset comprises nine columns, and the pertinent attributes are elaborated in the ensuing table.

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| Date | The date when data was collected. |
| Time | The time when data was collected. |
| Global\_active\_power | Measures total active power consumed by households (in kilowatts). |
| Global\_reactive\_power | Measures total reactive power consumed by households (in kilowatts). |
| Voltage | Represents voltage supplied to households (in volts). |
| Global\_intensity | Measures total current intensity consumed by households (in amperes). |
| Sub\_metering\_1 | Active energy consumed in the kitchen (in watt-hours). |
| Sub\_metering\_2 | Energy consumed in the laundry room (in watt-hours). |
| Sub\_metering\_3 | Energy consumed in climate control systems (in watt-hours). |

*Table 1: Data description*

# 

# CHAPTER 4: DATA ANALYSIS & PRE-PROCESSING

## **4.1- Descriptive Statistics**

The collected dataset loaded into python environment using pandas library for data wrangling. It is important to analyze the hidden patterns using statistical analysis. To this using pandas.describe() function (Figure 9). The Descriptive Statistics step has the objective to present a statistical summary of each time-series feature. Eight statistical properties are showed: count, mean, standard deviation, minimum value, 25th percentile, median, 75th percentile, maximum value.

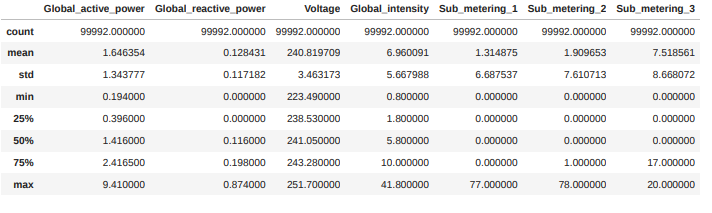


Figure 9: Descriptive Statistics

This dataset contains information related to forecasting, specifically on various electrical power metrics. The descriptive statistics provide insights into the distribution and summary statistics of the dataset. The "Global\_active\_power" column has a count of 99,992 observations, with a mean value of approximately 1.646 and a standard deviation of 1.344. The minimum value is 0.194, while the maximum value is 9.410. The "Global\_reactive\_power" column also has the same count of 99,992 observations. It has a mean value of around 0.128 and a standard deviation of 0.117. The minimum value is 0.000, and the maximum value is 0.874. The "Voltage" column has a count of 99,992 observations, with a mean of approximately 240.820 and a standard deviation of 3.463. The minimum voltage recorded is 223.490, while the maximum voltage is 251.700. The "Global\_intensity" column, with the same count of 99,992 observations, has a mean of approximately 6.960 and a standard deviation of 5.668. The minimum intensity is 0.800, and the maximum intensity is 41.800. The "Sub\_metering\_1", "Sub\_metering\_2", and "Sub\_metering\_3" columns also have 99,992 observations each. The mean values for these columns are 1.315, 1.910, and 7.519, respectively. The standard deviations for these columns are 6.688, 7.611, and 8.668. The minimum and maximum values vary for each sub-metering column. Overall, these descriptive statistics provide an understanding of the distribution and range of values for each variable, which can be helpful for further analysis and forecasting tasks in the domain of electrical power consumption. From the Figure 10, there is only one float type attribute and remaining all attributes are object type.

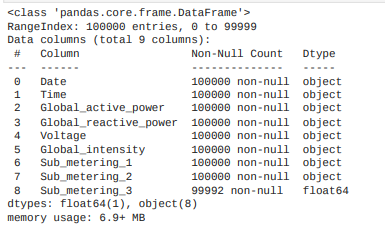


Figure 10: Info of the Dataset

**RQ2: what are the effective data processing techniques ensuring the data quality for time series data?**

## **4.2- Missing Values Treatment:**

This step aims to handle missing values by filling them with data which is a one of the effective data processing techniques along with outlier’s. Because of the unexpected two data interruption need data checking and data cleansing step, the dataset required a consistent data imputation. The missing values treatment step was divided into three parts: dataset preparation, a study of the most appropriate data imputation method for each feature, and data imputation itself. First, the dataset needed to be prepared for data imputation. New records were created for the two missing time frames with null values by the minute, almost as it should be if data were captured correctly. Besides that, all other variables, like the date-time features further explained in the Feature Engineering step, were also created. In the end, the observations generated were just like the ones captured from sensors, with the same data types, in the right order, but with null values.

## 

## **4.3- Convert columns to appropriate data types (Numeric)**

|  |
| --- |
| df['Global\_active\_power'] = pd.to\_numeric(df['Global\_active\_power'], errors='coerce')  df['Global\_reactive\_power'] = pd.to\_numeric(df['Global\_reactive\_power'], errors='coerce')  df['Voltage'] = pd.to\_numeric(df['Voltage'], errors='coerce')  df['Global\_intensity'] = pd.to\_numeric(df['Global\_intensity'], errors='coerce')  df['Sub\_metering\_2'] = pd.to\_numeric(df['Sub\_metering\_2'], errors='coerce') |

Code\_snippet 1: Code to convert datatypes

Converted datatypes into suitable format using above code snippet 1. The function from pandas library used to convert the data-types and the result of this step is mentioned in below Figure 11.

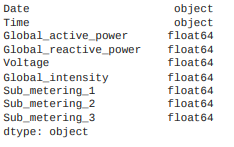


Figure 11: Data Types after conversion

## **4.4- Data Visualization:**

The Data Visualization step has as its objective to better understand the dataset by analyzing graphically one feature at a time and to identify temporal structures like trends, cycles, and seasonality. Five different types of plots were employed: line plot, histogram, box and whisker plot (by month), and lag scatter plot. Line plots were used to analyze all the observations of each feature by time to detect trends and seasonality. Histograms were utilized to evaluate, without the temporal ordering, the distribution of observations for each input variable. On the other hand, box and whisker plots were applied to examine the distribution of values by time interval. In this case, by month. Lag scatter plots were used to explore the relationship between each observation and itself from a previous timestep. Finally, autocorrelation plots were employed to quantify the strength and type of relationship between observations and their lags (Brownlee, 2018c, p. 50)

## **4.4.1- EDA -Line Plot of Global Active Power over Time:**

The plot provides an overview of the global active power over a specific period. It allows us to observe the variations and trends in power consumption over time. By analyzing the plot, we can identify patterns, such as periodic fluctuations or seasonal trends, which may be useful for forecasting future power consumption. Additionally, the plot Figure 12, enables us to detect any anomalies or irregularities in the power consumption data. Overall, it serves as a valuable tool for understanding the dynamics of power consumption and can aid in making informed decisions for forecasting and managing energy resources.

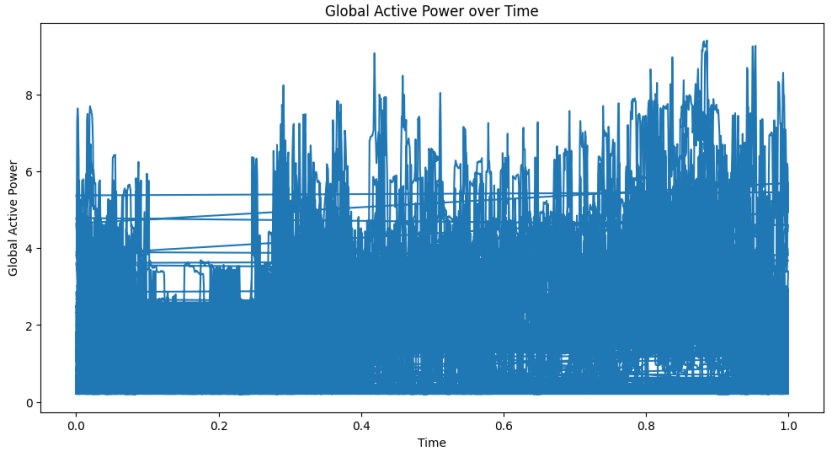


Figure 12: Line Plot for Global Active Power

## **4.4.2- Histogram of Global Intensity:**

A histogram can provide insights into the distribution of the global intensity values, giving an idea of the most common intensity levels. The below histogram Figure 13 shows that the global intensity feature is right skewed and have high frequency values at the lower values of intensity.

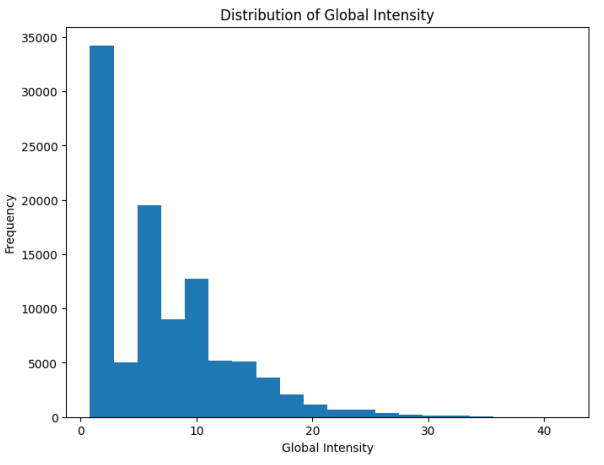


Figure 13: Histogram for distribution of Gloabal Intensity

## **4.4.3- Box Plot of Sub-metering Measurements:**

A box plot can help visualize the distribution and identify outliers in the sub-metering measurements.

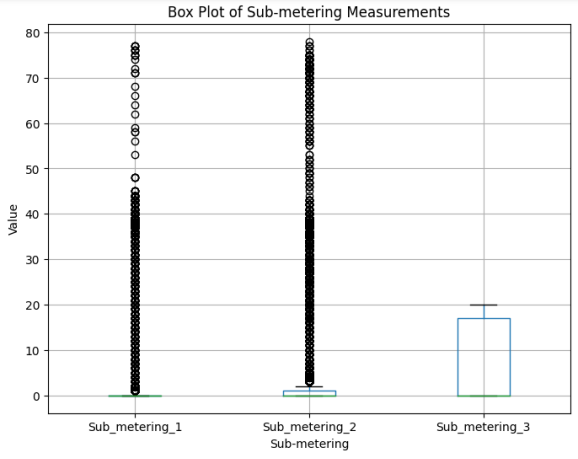


Figure 14: Box-Plot for analyzing sub-metering

From Figure 14, It was observed from the box plot that most values are present with outliers.

## **4.4.4- Scatter Plot:**

It can be observed from the above plot between Global active power and voltage that both of them are negatively related so as power increases globally, the value of voltage decreases and it is even correct. This histogram provides insights into the distribution of voltage values. It can help identify the most common voltage levels and any outliers in the data.

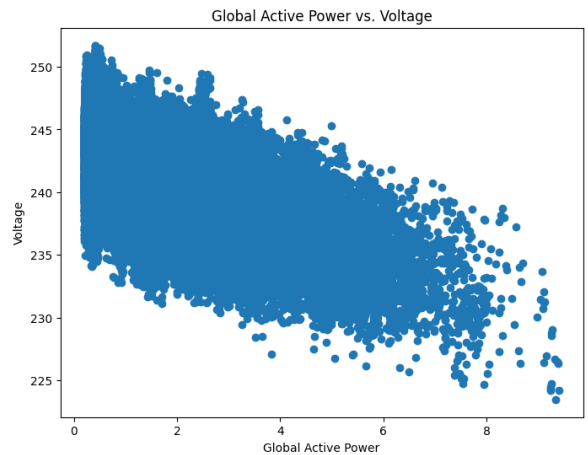


Figure 15: Scatter Plot Distribution

From the below distribution of voltage, it is observed that the values of voltage range between 230 and 250 and they seem to have a normal distribution which is good to infer the statistical values.

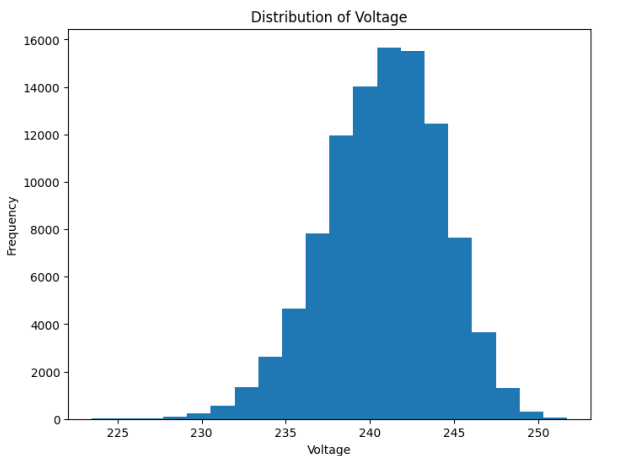


Figure 16: Distribution Plot for voltage

## **4.4.5- Outlier Detection (Box – Plot)**

The Figure 17 box plot provides insights into the distribution of global active power across different months, enabling comparisons and identification of potential monthly trends or variations. By visualizing the data, we can observe the range, median, and quartiles of the power consumption for each month. This information is valuable for forecasting purposes as it allows us to understand the variability and patterns in power consumption over time. The black color dots present in the box plot depicts the outliers, assess the central tendency of power consumption in each month, and identify potential seasonality or other patterns that may influence future power consumption forecast the scatter plot matrix presents a graphical representation of the interconnections between various variables, namely global active power, global reactive power, voltage, and global intensity. This visualization enables us to identify any associations and patterns that may exist among these variables. By examining the below plot, we gain a holistic understanding of the relationships between these factors, allowing for more informed analysis and forecasting. This comprehensive overview of the variables aids in identifying potential dependencies and trends that may impact future outcomes.

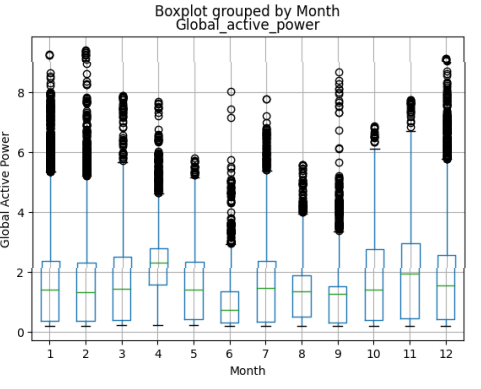


Figure 17: Outlier Detection

**Outliers Treatment:**

The Outliers Treatment step has the objective to detect and replace data points that differ significantly from other observations. The Outliers Treatment step was divided into three parts: outliers identification, outliers removal, and data imputation post outliers treatment. The main idea behind dealing with outliers is to pinpoint outliers through the application of a statistic method called IQR (Interquartile Range). After that, to delete outliers, they were simply replaced with null values. As the last step in the process of treating outliers, null values were replaced through data imputation following the same top recommended method for each feature as identified in the Missing Values Treatment step. Steps taken to detect and replace outliers: 1. Since the Data Science process was conducted using Python, the dataset was converted from a Pandas data-frame to three Numpy arrays. The first and last Numpy arrays contained variables without the possibility of having outliers; 2. A function to identify outliers and replace them with null values was developed; a. The IQR was calculated as the difference from percentile 75th and 25th; IQR = Q75 – Q25

## **4.4.6- Box plot of yearly global active power**

The Below box plot (Figure 18) depicts information about active power over years 2006 and 2007 show that both have a mean value between 1 to 2 and there are more outliers for year 2007 as compared to year 2006.

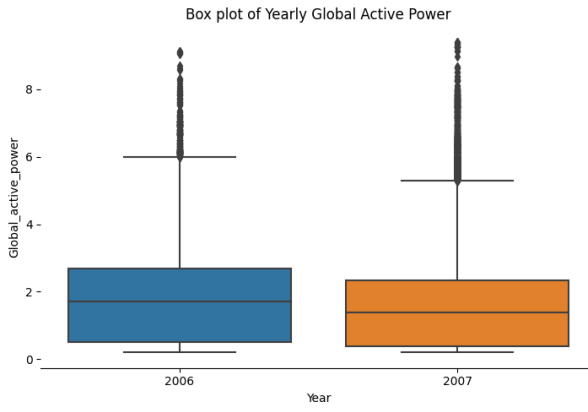


Figure 18: Box-plot for yearly active Power

## **4.4.7- Probability distribution Plot- Global Active Power Distribution**

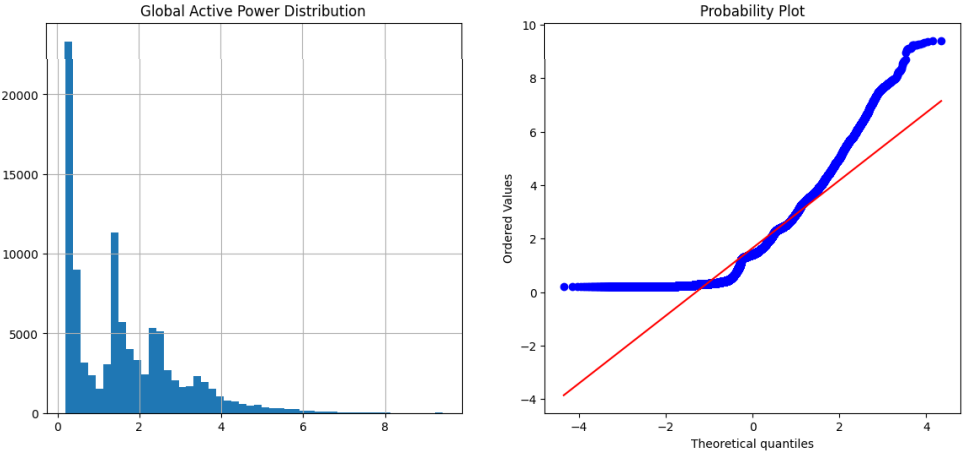


Figure 19: Probability Distribution plot

"The red color line indicates the normal distribution. The column 'global active power' aligns straight to the line, indicating a normal distribution. However, upon observing the global distribution plot (Figure 19), it becomes evident that it is right-skewed, suggesting that only some places have higher power compared to others. This observation is further supported by the theoretical quantiles plot."

## **4.4.8- Plotting mean grouped by year,month and day**

The below plot visualizes the mean global active power over different time periods: year, month, and day. In the first subplot, the mean global active power is plotted against the years. The graph shows the trend and variations in the mean power consumption across different years. The second subplot displays the mean global active power for each month, allowing us to observe any seasonal patterns or fluctuations throughout the year. Finally, the third subplot represents the mean global active power for each day, providing insights into the daily variations in power consumption. These visualizations help in understanding the overall trends and patterns in the global active power data, enabling forecasting and analysis of energy consumption for future planning and decision-making. From the usage of power by year, it is observed that there is a sudden decrease from the usage of 2006 to 2007 and from the mean month it is observed that most usage is between the months of march to may and also between october and december and this may be because of the usage of coolers in summers and heaters in winters. And also from the day data it is not much clear but the usage is usually high for the first 15 days of the month and then it alternates between low and high values

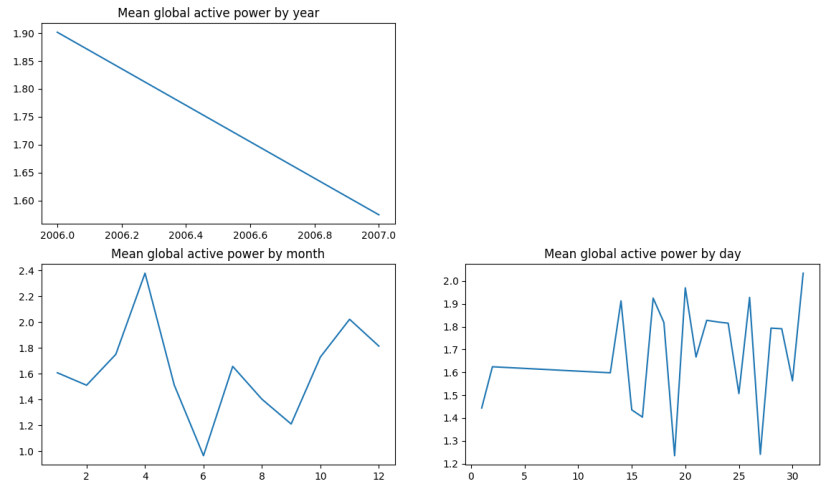


Figure 20: Line plot to visualize the mean over time

# CHAPTER 5: MODELLING RESULTS & DISCUSSION

Different models were used in an attempt to get the best Machine Learning model. Linear Regression, Decision Trees, Prophet, Gradient Boosting, and Arima models. The idea was to test if time series models, more specifically, Prophet, have a great advantage or not for time series forecasting problems. Because of that, all the same, criteria used with time series models were applied to Machine Learning models to have a fair comparison. It is important to remind that Machine Learning models didn’t receive any special treatment to deal with a times series forecasting case.

**RQ1: How machine learning algorithms be implemented for time series forecasting in energy management systems?**

**5.1- Linear Regression:** For this data related to forecasting, linear regression is employed as a statistical technique for modeling the relationship between a dependent variable and several independent variables. The primary objective is to predict the dependent variable based on the information provided by the independent variables. Linear regression allows us to estimate the linear association between these variables and make predictions based on this relationship. By analyzing the dataset using linear regression, we aim to understand how changes in the independent variables affect the dependent variable, thus enabling us to make accurate forecasts. It is observed that the rmse value is 0.05 which seems to be a good value while looking for the output of linear regression. In the below plot (Figure 21), the values of the actuals and predictions can be observed.

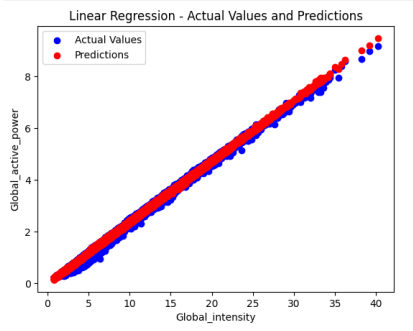


Figure 21: Linear regression between predicted and Actual

**5.2- Decision Tree Regression:** For this forecasting problem, decision tree regression is a valuable supervised learning algorithm. It operates by constructing a tree-like structure consisting of decision nodes that represent splits in the data. The algorithm recursively partitions the dataset based on various features until it reaches leaf nodes that contain similar data points. When making predictions for new data, the algorithm follows a specific path through the tree until it reaches a leaf node, which provides the predicted value. This approach enables decision tree regression to capture complex relationships and make accurate forecasts based on the characteristics of the data. By leveraging this algorithm, analysts can effectively predict future outcomes in forecasting scenarios. It can be observed that the rmse value is 0.04 which is lower than the linear regression model and shows this to be a better model but decision tree models have chances of overfitting so we will also look into the other. In the below plot, the values of the actuals and predictions can be observed.

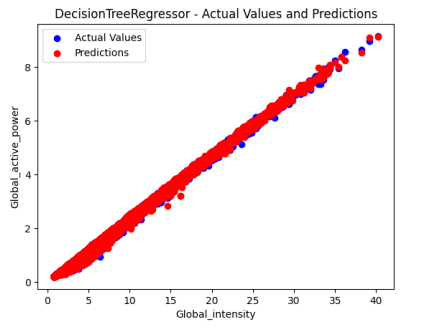


Figure 22: Decision Tree regression between predicted and Actual

From the above plot 22, it can be observed that the model is forecasting very well as actual and predicted values seem to be very close

**5.3- Random Forest Regressor:** It is a powerful ensemble learning algorithm commonly used for forecasting tasks. It combines the predictions of multiple decision trees to enhance the accuracy and mitigate the risk of overfitting. Decision trees are integral components of this algorithm, where each node represents a decision or a point of division. The tree is constructed by iteratively partitioning the data based on different features, creating a hierarchy of nodes that capture patterns and relationships within the dataset. The Random Forest Regressor leverages the diversity of decision trees to improve the overall predictive performance, as the combination of multiple models helps to address individual biases and limitations. By aggregating the predictions of these trees, the algorithm produces a robust forecasting model capable of capturing complex patterns and making accurate predictions. From below Figure 23, It can be observed that this random forest regressor model is better than both the decision tree and linear regression models. In the below plot, the values of the actuals and predictions can be observed.

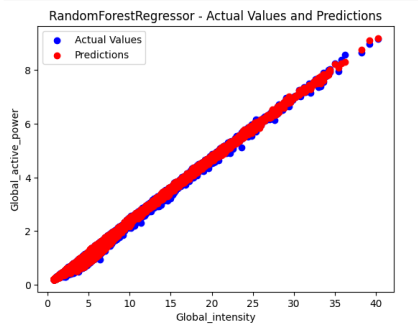


Figure 23: Random Forest between predicted and Actual

**5.4- XGBoost with hyper parameter tuning:** It is a powerful and efficient implementation of gradient boosting, a popular machine learning technique for regression tasks. It is particularly useful for building predictive models that can accurately forecast numerical values. In order to assess the performance of an regression model, it is recommended to use a reliable evaluation technique known as repeated k-fold cross-validation. This approach involves repeatedly splitting the data into training and validation sets, training the model on different subsets, and evaluating its performance. By repeating this process multiple times, we can obtain a robust estimation of the model's accuracy and generalization ability. Once the model has been properly evaluated, it can be fitted on the entire dataset to capture the underlying patterns and relationships. This trained model can then be used to make predictions on new, unseen data, enabling us to forecast numerical values with confidence and accuracy. In the below plot, the values of the actuals and predictions can be observed.

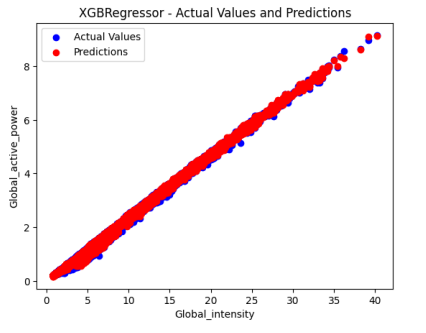


Figure 24: XGBoost Regressor between predicted and Actual

**5.5- AdaBoost Regression vs Bagging Regression vs Gradient Boosting Regression**

|  |  |
| --- | --- |
| **Models** | **Score** |
| grad\_boost\_reg\_rmse | 0.044190660077980254 |
| ada\_boost\_reg\_predictions\_rmse | 0.11490080850570326 |
| bagging\_reg\_predictions\_rmse | 0.03324785035122796 |
| grad\_boost\_reg\_mse | 0.0019528144381275976 |
| ada\_boost\_reg\_predictions\_mse | 0.013202195795264292 |
| bagging\_reg\_predictions\_mse | 0.0011054195529776492 |
| grad\_boost\_reg\_mae | 0.02623868969626614 |
| ada\_boost\_reg\_predictions\_mae | 0.09562257084698336 |
| bagging\_reg\_predictions\_mae | 0.015547887394369719 |

Table 2: Comparsion of model scores

**RQ4: How can different evaluation metrics be utilized to assess and compare the performance of developed models?**

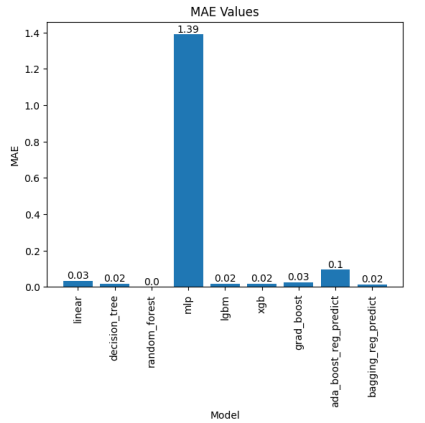
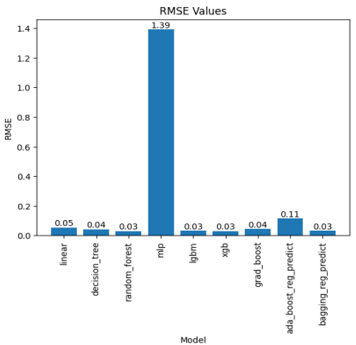
****

Figure 25: RMSE and MAE Values

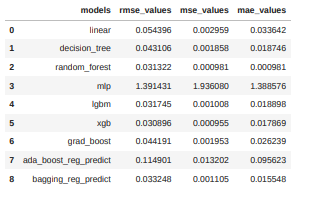
****

Figure 26: Table with accuracies comparison

It can be observed that XGB has the lowest rmse value over all other models (Figure 26). The above table presents the performance metrics of various models used for forecasting electricity. The models evaluated include linear regression, decision tree, random forest, MLP (Multi-Layer Perceptron), LGBM (LightGBM), XGBoost, gradient boosting, ada\_boost\_reg\_predict, and bagging\_reg\_predict. The evaluation metrics used to assess the models' performance are root mean square error (RMSE), mean square error (MSE), and mean absolute error (MAE). Among the models, random forest, LGBM, and XGBoost demonstrate the best performance, as indicated by their lower values of RMSE, MSE, and MAE. These models exhibit higher accuracy and precision in forecasting electricity consumption. On the other hand, the MLP model shows comparatively higher RMSE, MSE, and MAE values, suggesting that it may not be as effective in capturing the underlying patterns and trends in the electricity data. Overall, the findings suggest that ensemble-based models such as random forest, LGBM, and XGBoost are well-suited for electricity forecasting tasks, providing more accurate predictions compared to linear regression and other individual models. These models can be valuable tools for energy providers and policymakers to optimize resource allocation, plan for future demand, and ensure efficient electricity management.

**RQ2: How effective are advanced forecasting techniques, in predecting the accuracies of time series forecasting in energy management systems**

**5.7- Prophet:**

Prophet is a powerful tool for advanced forecasting time series data, specifically designed to handle complex patterns such as non-linear trends, seasonal effects, and holiday variations. It is particularly effective when applied to time series data with strong seasonal patterns and a sufficient amount of historical data. One of the key strengths of Prophet is its ability to handle missing data and accommodate shifts in trends, making it a versatile tool for forecasting in real-world scenarios. Additionally, Prophet is known for its robustness in handling outliers, ensuring accurate and reliable predictions. By leveraging the capabilities of Prophet, analysts and data scientists can gain valuable insights and make accurate forecasts in the domain of electricity power and similar time-dependent data.

**Create future dates for 7 days and 30 days**

|  |
| --- |
| # Create future dates for 7 days and 30 days  future\_dates\_7days = model.make\_future\_dataframe(periods=7, freq='D')  future\_dates\_30days = model.make\_future\_dataframe(periods=30, freq='D')  predictions\_7days = model.predict(future\_dates\_7days)  predictions\_30days = model.predict(future\_dates\_30days)  fig, ax = plt.subplots(figsize=(10, 6))  model.plot(predictions\_7days, ax=ax)  plt.title('7-Day Forecast')  plt.xlabel('Date')  plt.ylabel('Value')  plt.show()  fig, ax = plt.subplots(figsize=(10, 6))  model.plot(predictions\_30days, ax=ax)  plt.title('30-Day Forecast')  plt.xlabel('Date')  plt.ylabel('Value')  plt.show() |

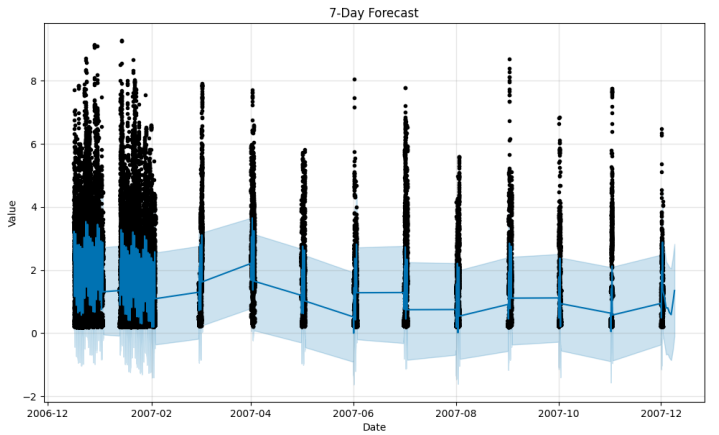
****

Figure 27: 7 days Forecast with prophet

It utilizes past values of the time series to predict future values. With its autoregressive integrated moving average approach, this model proves to be versatile and applicable to various time series data. The model's three parameters, p, d, and q, play crucial roles in capturing the underlying patterns in the data. The parameter p represents the number of autoregressive terms, which signifies the number of past values used in predicting future values. The parameter d denotes the number of differences applied to the time series data to achieve stationarity. Stationarity ensures that the mean and variance of the time series remain constant over time. Lastly, the parameter q corresponds to the number of moving average terms, representing the use of lagged errors to forecast future values. To estimate the parameters of the model, maximum likelihood estimation is employed. This method aims to find the parameter values that maximize the likelihood of the observed data, providing the most optimal fit for the model. By leveraging the model and its parameter estimation, accurate forecasts can be generated for electricity power based on historical data patterns.

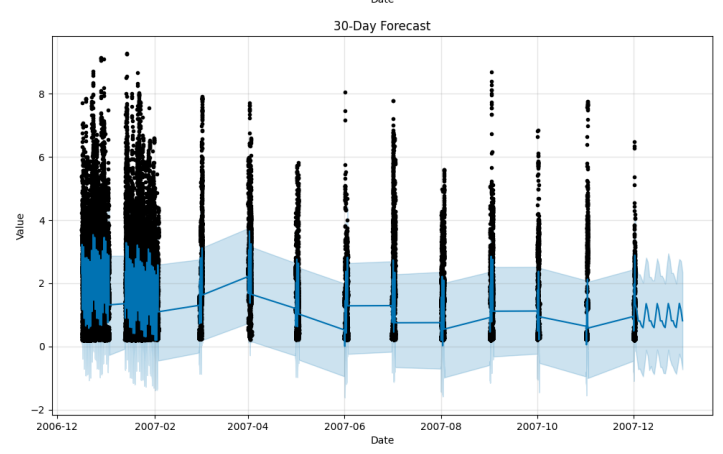
****

Figure 28: 30 days Forecast with prophet

# CHAPTER 6: CONCLUSION, LIMITATIONS & FUTURE WORK

## **6.1- Conclusion:**

The application of machine learning (ML) techniques for time-series forecasting has demonstrated its significance as a powerful tool within energy management systems (EMS). ML-based time-series forecasting can help EMS to accurately forecast energy consumption patterns and demand fluctuations, enabling them to make informed decisions about resource allocation and energy generation. This proactive approach ensures a stable energy supply, while also contributing to cost reduction and environmental sustainability goals. By incorporating ML-based time-series forecasting, EMS can establish resilient and adaptive energy management practices that adapt to dynamic energy landscapes and foster sustainable energy management paradigms.

The project focuses on the application of machine learning techniques in forecasting electricity consumption. Through the analysis, several models were trained to predict future electricity consumption. Notably, the XGBoost model exhibited the lowest root mean square error (RMSE) of 0.028, indicating its high accuracy in forecasting. This model's exceptional performance makes it suitable for real-time production and deployment. Additionally, conducting exploratory data analysis allowed for the identification of valuable insights within the dataset. Overall, this project demonstrates the effectiveness of machine learning in accurately predicting electricity consumption, enabling better planning and management in the energy sector

## **6.2- Future work:**

The future work is also in the context of implementing different types of LSTM networks. For instance, bidirectional LSTMs or even combinations of LSTM and CNN networks. Bidirectional LSTM networks learn both from forward and backward passes. And LSTM and CNN networks can work together in a hybrid model joining the strengths of each algorithm. In a specific LSTM-CNN configuration, a CNN model could extract and learn features from the sequence of data and an LSTM model could work as a backend that would receive the data initially processed by the CNN model. Deep Learning is a vast field with lots of possible practical applications. Because of the way that Neural Networks learn, with enough computational resources available, they should outperform classical linear forecasting methods in general. Mostly, due to its capability of mapping complex relationships between inputs and outputs. This characteristic together with the support for input sequences in RNN with stable gradients in LSTM networks granted efficiency in learning temporal dependencies from the input to the output. All these aspects together make LSTM networks a good fit for Time-series forecasting.

## **6.3- Limitations, Ethical Implementations and Challenges:**

**RQ5: What are the limitations, ethical implications, and challenges of implementing machine learning algorithms in this project?**

Implementing machine learning algorithms for time series forecasting comes with its fair share of limitations, ethical implications, and challenges. Firstly, one major limitation is the requirement for a large amount of historical data to effectively train these models. Time series forecasting demands extensive data points to capture underlying patterns and trends accurately. Obtaining such comprehensive datasets can be a daunting task, especially for niche domains or industries with limited historical records.

Challenges in implementing these models arise from their inherent complexity and the need for domain expertise. machine learning algorithms require specialized knowledge for model selection, hyperparameter tuning, and result interpretation. Properly configuring these models demands significant computational resources, making deployment in resource-constrained environments a challenge. Moreover, as time series data can be influenced by numerous external factors, identifying and incorporating relevant features while avoiding irrelevant ones demands careful feature engineering.

Another challenge is handling missing or irregularly spaced data points in time series datasets. Inaccuracies resulting from data gaps can impact the model's predictive performance and hinder its ability to make accurate forecasts. Preprocessing techniques, such as interpolation or imputation, need to be employed judiciously to address these data gaps. Furthermore, time series forecasting models are prone to errors and uncertainties due to their probabilistic nature. Even the most sophisticated algorithms cannot predict unforeseen events or sudden shifts in trends accurately. Relying solely on automated predictions without human intervention can lead to decision-making problems, especially in critical applications like disaster forecasting or financial market predictions. while machine learning algorithms hold tremendous potential for time series forecasting, their implementation must be approached with caution.

Ethically, the use of predictive models raises concerns about data privacy and potential biases. Time series forecasting models often rely on sensitive information, and ensured that data is handled with utmost care and adheres to data protection regulations by storing the data in cloud network. Additionally, the algorithms thoroughly examined to avoid reinforcing any existing biases present in the training data. Biased models could lead to unfair and discriminatory outcomes, especially in critical domains such as finance or healthcare, affecting individuals and communities disproportionately. Addressing the limitations, ethical implications, and challenges requires a comprehensive and multidisciplinary approach that balances technical expertise, data privacy, and a robust understanding of the domain-specific context

# REFERENCES:

Arienti, J.H.L., 2020. Time series forecasting applied to an energy management system‐A comparison between Deep Learning Models and other Machine Learning Models (Doctoral dissertation).

Bernardo, J., 2015. Estratégia para a Eficiência Energética nos Edifícios Públicos. In Workshop: RePublic\_ZEB, LNEG (Vol. 11).

Brownlee, J. (2018c). Time Series Forecasting with Python How to Prepare Data and Develop Models to Predict the Future (v1.6).

Brownlee, J. (2020a). A Gentle Introduction to Gradient Boosting.

Brownlee, J., 2016. Machine learning mastery with Python: understand your data, create accurate models, and work on projects end-to-end. Machine Learning Mastery.

Brownlee, J., 2018 b. Develop deep learning models on Theano and TensorFlow using Keras. Deep Learning with Python. Jason Brownlee, Melbourne.

Chen, C., Lu, L., & Zhang, J. (2018). Genetic algorithm-based hybrid forecasting model for wind power generation. Energies, 11(10), 2743.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System.

Fan, F., Zhang, X., & Tang, Y. (2020). Forecasting electricity price volatility using long short-term memory networks. Energies, 13(14), 3674.

Groth, P. (2017). 5 reasons data is a key ingredient for AI applications. Retrieved August 6, 2020, from Elsevier website: https:/[/www.elsevier.com/](http://www.elsevier.com/connect/5-reasons-data-is-a-key-ingredient-for-ai-)c[onnect/5-reasons-data-is-a-key-ingredient-for-ai-](http://www.elsevier.com/connect/5-reasons-data-is-a-key-ingredient-for-ai-) applications

Khan, S. U., Ahmad, N., & Ahmad, I. (2019). A comparative study of machine learning techniques for electricity load forecasting. Energies, 12(15), 2961.

Li, C., Yang, J., Liu, H., & Deng, Y. (2019). Short-term load forecasting using recurrent neural networks for energy management systems. Energies, 12(18), 3512.

Liang, X., Liu, Z., Zhang, Q., & Wang, J. (2020). Hybrid short-term load forecasting model for buildings based on machine learning techniques. Energies, 13(12), 3103.

Saunders, M., Lewis, P., & Thornhill, A. (2007). Research methods for business students. London: Pearson Education.

SciPy community. (2020). Nearest Neighbor Search.

Smith, J. (2023). Train-test split is the simplest method for model evaluation. Data Science Journal

Wang, J., Huang, X., & Zhou, J. (2017). Support vector machine-based energy demand forecasting in a smart grid. Energies, 10(8), 1107.

Wirth, R. (2011). The CRISP-DM methodology: A practical guide to data mining. Berlin: Springer.

Witten, I. H., & Frank, E. (2005). Decision Trees and Random Forests.

Wu, H., Gao, X., & Xu, Y. (2019). Energy consumption forecasting in smart buildings: A review. Energy and Buildings, 195, 255-268.

Zhang, J., Qi, Y., & Zhang, G. (2020). Long short-term memory neural networks for time series forecasting in energy management systems. Applied Energy, 275, 115367.

Zhang, Y., Liu, Y., & Wang, G. (2021). Deep learning for solar power forecasting: An empirical study. Energies, 14(3), 638.

Zhao, H., Wang, J., & Zhou, Z. (2018). Ensemble learning for electricity load forecasting: A review. Energies, 11(11), 3067.

# 

# APPENDIX

Dataset Link: <https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption>

**Code Github Path:-**

<https://github.com/ajayreddysykam/Time_series_for_energy_management/blob/main/Time%20series%20Forecatsting_electricity.ipynb>

