FORECASTING ENERGY CONSUMPTION USING MACHINE LEARNING

A DISSERTATION SUBMITTED TO MANCHESTER METROPOLITAN UNIVERSITY
FOR THE DEGREE OF MASTER OF SCIENCE
IN THE FACULTY OF SCIENCE AND ENGINEERING

2024

By
Anthony Anayo Eze
Department of Computing and Mathematics

Contents

Abstract **x**

Declaration **xi**

Acknowledgements **xii**

Abbreviations **xiii**

1 Introduction **1**

1.1 Project Overview 1
1.2 Problem Statement 2
1.3 Objectives 3
1.4 Research Questions 3
1.5 Scope Of Study 4
1.6 Project Timeline 5
1.7 Dissertation Structure 6

2 Literature Review **7**

2.1 Introduction 7
2.2 Introduction To Energy Consumption Forecasting 8
2.2.1 Background Information on Energy Usage Forecasting 8
2.2.2 Importance of Accurate Energy Forecasts 8
2.3 Time Series Analysis In Energy Forecasting 9
2.3.1 Components of Time Series Data 9
2.4 Traditional Forecasting Methods 12
2.4.1 Historical Development and Key Principles 12
2.4.2 Applications in Energy Consumption Forecasting 13
2.5 Machine Learning In Energy Forecasting 15

2.5.1 Supervised Learning . . . 16
2.5.2 Linear Regression . . . 18
2.5.3 Decision Tree . . . 19
2.5.4 Random Forest Regressor (RFR) . . . 20
2.5.5 Gradient Boosting Machine (GBM) . . . 20
2.5.6 Extreme Gradient Boosting (XGBoost) . . . 21
2.5.7 Support Vector Regression (SVR) . . . 22
2.5.8 Ensemble Learning . . . 22
2.5.9 Neural Networks . . . 24
2.6 Key Challenges In Energy Consumption Forecasting . . . 28
2.7 Summary . . . 29

3 Methodology **31**
3.1 Introduction . . . 31
3.2 Data Collection . . . 32
3.3 Data Preprocessing . . . 34
3.3.1 Transforming Data to a Time-Series Format . . . 34
3.3.2 Handling Missing Values and Outliers . . . 35
3.3.3 Feature Engineering . . . 36
3.3.4 Data Normalization . . . 39
3.4 Model Building . . . 40
3.4.1 Introduction to Model Building . . . 40
3.4.2 Linear Regression . . . 41
3.4.3 Decision Tree . . . 42
3.4.4 Random Forest Regressor . . . 42
3.4.5 Gradient Boosting Regressor . . . 43
3.4.6 Extreme Gradient Boosting (XGBoost) . . . 43
3.4.7 Support Vector Regressor (SVR) . . . 44
3.4.8 Multilayer Perceptrons (MLPs) . . . 44
3.4.9 Convolutional Neural Networks (CNNs) . . . 45
3.4.10 Long Short-Term Memory networks (LSTMs) . . . 45
3.4.11 Hyperparameter Estimation . . . 45
3.4.12 Ensemble Learning . . . 51
3.5 Result Collection . . . 52
3.6 Result Analysis . . . 53
3.7 Summary . . . 54

4 Result and Discussion **55**
4.1 Introduction . . . 55
4.2 Data Preprocessing . . . 55
4.3 Exploratory Data Analysis (EDA) . . . 56
4.3.1 Summary Statistics . . . 56
4.3.2 Correlation Analysis Results . . . 57
4.3.3 Datetime features Analysis Results . . . 58
4.3.4 Energy Consumption vs Temperature Analysis Results . . . 59
4.4 Individual Model Results . . . 60
4.4.1 Linear Regression Results . . . 60
4.4.2 Decision Tree Results . . . 62
4.4.3 Random Forest Results . . . 63
4.4.4 Gradient Boosting Results . . . 65
4.4.5 Extreme Gradient Boosting Results . . . 67
4.4.6 Support Vector Regressor (SVR) Results . . . 68
4.4.7 Neural Network (MLP, CNN, LSTM) Results . . . 70
4.4.8 Ensemble Learning Results . . . 74
4.5 Comparative Analysis . . . 76
4.5.1 Comparison of All Models . . . 76
4.5.2 Discussion of Best Model . . . 77
4.5.3 Effect of Feature Engineering . . . 77
4.5.4 Trade-offs Between Model Complexity and Accuracy . . . 77
4.6 Discussion of Key Findings . . . 78
4.6.1 Model explainability with SHAP . . . 78
4.6.2 Impact of Hyperparameter Tuning . . . 80
4.6.3 Limitations . . . 81
4.6.4 Practical Implications . . . 81
4.6.5 Comparison with Literature . . . 82
4.7 Summary . . . 83

5 Case Study **84**
5.1 Introduction to the Case Study . . . 84
5.2 Description of the Case Study Data . . . 84
5.3 Application of the Developed Models . . . 85
5.4 Results of the Case Study . . . 85
5.5 Discussion . . . 85

5.6 Summary 88

6 Further Work **89**

6.1 Introduction 89

6.2 Data Collection and Preprocessing 89

6.3 Model Development and Experimentation 89

6.4 Managing Uncertainty and Enhancing Model Explainability 90

6.5 Broader Applications and Future Case Studies 91

6.6 Summary 91

7 Conclusion **92**

References **94**

Appendices **96**

A Terms of Reference and Ethics **97**

B Dataset **113**

C All Experimental code **114**

D Experimental results **115**

List of Tables

1.1 Project Timeline 5

3.1 Hyperparameter tuning for Linear and Polynomial Regression models 46
3.2 Hyperparameter tuning for Decision Tree Regressor 47
3.3 Hyperparameter tuning for Random Forest Regressor 47
3.4 Hyperparameter tuning for Gradient Boosting Regressor 48
3.5 Hyperparameter tuning for XGBoost 48
3.6 Hyperparameter tuning for Multilayer Perceptrons (MLPs) using Keras Tuner 49
3.7 Hyperparameter tuning for Convolutional Neural Networks (CNNs) using Keras Tuner 50
3.8 Hyperparameter tuning for Long Short-Term Memory Networks (LSTMs) using Keras Tuner 51
3.9 Result collection format 52

4.1 Model Performance Comparison 76

5.1 Model Performance Comparison 85

List of Figures

1.1 Gantt Chart Project Timeline. 5

2.1 Showing the various forms of trends. 10
2.2 The various forms of Seasonality. 11
2.3 Typical workflow of supervised machine learning algorithms. 16
2.4 Types of supervised machine learning. 17
2.5 Examples of Classification and Regression. 17
2.6 Linear Regression. 18
2.7 Example of a Decision Tree relating to Energy Forecasting. 19
2.8 Example of an Ensemble Learning. 23
2.9 A typical Neural Network. 25
2.10 A typical Multilayer Perceptrons (MLPs). 25
2.11 A typical Convolutional Neural Networks (CNNs). 26
2.12 A typical Long Short-Term Memory networks (LSTMs). 27

3.1 Experimental Methodology Framework. 32
3.2 Pjm Zones. 33
3.3 PJM Dominion Energy(DOM) Data table. 34
3.4 Python Code for Processing and Concatenating Data. 35
3.5 PJM DOM consumption 2006 to 2024 in MW. 36
3.6 Python Function for Getting Holiday Features. 38
3.7 Python Function for Getting Weather Features. 39
3.8 Calling all Python functions and Merging Features. 39
3.9 Ensemble learning. 52

4.1 Data Preprocessing Result. 56
4.2 Correlation Analysis. 57
4.3 Plots of Energy consumption vs Hour, Hour by Day of the week, Month and Year. 58

4.4 Energy Consumption vs Average Temperature 59
4.5 Linear regression plot: Actual vs Prediction from January 2016 to December 2019 61
4.6 Linear regression plot: Scatter-plot Actual vs Predicted 61
4.7 Linear regression plot: Feature Importance 61
4.8 Decision Tree regression plot: Actual vs Prediction from January 2016 to December 2019 63
4.9 Decision Tree regression plot: Scatter-plot Actual vs Predicted 63
4.10 Decision Tree regression plot: Feature Importance 63
4.11 Random Forest regression plot: Actual vs Prediction from January 2016 to December 2019 64
4.12 Random Forest regression plot: Scatter-plot Actual vs Predicted . . . 64
4.13 Random Forest regression plot: Feature Importance 64
4.14 Gradient Boosting regression plot: Actual vs Prediction from January 2016 to December 2019 66
4.15 Gradient Boosting regression plot: Scatter-plot Actual vs Predicted . . 66
4.16 Gradient Boosting regression plot: Feature Importance 66
4.17 Extreme Gradient Boosting regression plot: Actual vs Prediction from January 2016 to December 2019 68
4.18 Extreme Gradient Boosting regression plot: Scatter-plot Actual vs Predicted 68
4.19 Extreme Gradient Boosting regression plot: Feature Importance . . . 68
4.20 Support Vector regression plot: Actual vs Prediction from January 2016 to December 2019 69
4.21 Support Vector regression plot: Scatter-plot Actual vs Predicted . . . 69
4.22 MLP Plot: Residual Error Distribution 71
4.23 MLP Plot: Training and Validation Loss Curves 71
4.24 MLP Plot: Scatter Plot of Actual vs Predicted Values 71
4.25 MLP Plot: Actual vs Prediction from January 2016 to December 2019 71
4.26 CNN Plot: Residual Error Distribution 72
4.27 CNN Plot: Training and Validation Loss Curves 72
4.28 CNN Plot: Scatter Plot of Actual vs Predicted Values 72
4.29 CNN Plot: Actual vs Prediction from January 2016 to December 2019 72
4.30 LSTM Plot: Residual Error Distribution 73
4.31 LSTM Plot: Training and Validation Loss Curves 73

4.32 LSTM Plot: Scatter Plot of Actual vs Predicted Values 73
4.33 LSTM Plot: Actual vs Prediction from January 2016 to December 2019 73
4.34 Ensemble Plot: Actual vs. Predicted for Ensemble_XGB_RFR_GBR . 75
4.35 Ensemble Plot: Scatter Plot for Ensemble_XGB_RFR_GBR 75
4.36 Ensemble Plot: Actual vs. Predicted for Ensemble_XGB_GBR 75
4.37 Ensemble Plot: Scatter Plot for Ensemble_XGB_GBR 75
4.38 SHAP Summary Plots for Random Forest Regression Model (Hyperparameter Tuned 2) 78
4.39 SHAP Summary Plots for Gradient Boosting Regression Model (Hyperparameter Tuned 2) 79
4.40 SHAP Summary Plots for XGBoost Regression Model (Hyperparameter Tuned 2) 80

5.1 LSTM Case Study Plot: Actual vs Prediction from July 2023 to July 2024 86
5.2 LSTM Case Study Plot: Scatter Plot of Actual vs Predicted Values . . 86
5.3 LSTM Case Study Plot: Training and Validation Loss Curves 86
5.4 Hyperparameter Tuned Xgboost 2 Case Study plot: Actual vs Prediction from July 2023 to July 2024 87
5.5 Hyperparameter Tuned Xgboost 2 Case Study plot: Scatter Plot of Actual vs Predicted Values 87
5.6 Hyperparameter Tuned Xgboost 2 Case Study plot: Feature Importance 87

Abstract

This study focuses on developing machine learning models, which would increase the accuracy of both short-term and long-term forecast energy consumption, which has been challenged by increasing demands in energy and complexity in energy usage patterns. Traditional approaches to energy consumption forecasting have failed to understand nonlinear and dynamic natures, leading to significant errors in energy consumption forecasting. The work seeks to override the shortcomings of previous traditional approaches by leveraging recent machine learning techniques in developing better energy management and supporting transition to net zero.

To address this problem, historical energy consumption data from the PJM Interconnection(Dominion Energy) was preprocessed and used to develop several models, including Linear Regression, Decision Tree, Random Forest, Gradient Boosting, Extreme Gradient Boosting (XGBoost), Ensemble learning and Neural Networks (MLP, CNN, LSTM). The models were evaluated using the following metrics which are RMSE, MAE, and MAPE. Hyperparameter tuning and feature engineering were applied to enhance model performance, and feature importance were utilized for model explainability. The SHAP plots was also used to confirm the model explainablity which showed that Temperature and hour played a big part in forecasting energy. A case study using up to date data from PJM Interconnection(Dominion Energy) was conducted to validate the models' practical applicability.

The results demonstrate that machine learning models significantly improve the accuracy of energy consumption forecasts compared to traditional methods. The top 3 performing models were LSTM, Ensemble(XGBoost, RFR, GBR) and Hyperparameter tuned XGBoost 2. LSTM models provided the best performance, reducing forecast errors and offering practical insights into energy consumption patterns. These findings highlight the potential of machine learning in enhancing energy efficiency and supporting sustainable energy systems.

Declaration

No part of this project has been submitted in support of an application for any other degree or qualification at this or any other institute of learning. Apart from those parts of the project containing citations to the work of others, this project is my own unaided work. This work has been carried out in accordance with the Manchester Metropolitan University research ethics procedures, and has received ethical approval number 57973.

Signed: 

Date: 24/09/2024

Acknowledgements

I would like to thank everyone who have supported me through this dissertation journey. This work is dedicated, firstly , to the loving memory of my late mother, Eunice Eze. Her strength and guidance continue to inspire me every day. My father, Princewill Eze, and my siblings, Michael Eze, Ada Eze, and Onyebuchi Eze, your unwavering support has been invaluable. I am forever thankful for your encouragement.

To my fiancée, Omotolu Oyewole, thank you for your love, patience, and support over the past 12 months. Your presence has been my constant source of motivation.

I hereby want to express my deepest appreciation to the following individuals at Scottish Power: my supervisors, Ainsley Meechan and Peter Brown, for the advice and mentorship that helped me while I was working with the GEM – Portfolio Optimisation team. Your insight and experience have greatly contributed to my learning and success regarding this project.

Finally, I want to say a big thank you to my dissertation supervisor, Anthony Bukowski for reviewing my work and giving me valuable feedback, especially on the structure. Your guidance helped me navigate the challenges of this project, and I deeply appreciate it.

Above all, I thank God for His strength and blessings throughout this journey.

Abbreviations

ARIMA	AutoRegressive Integrated Moving Average
CNNs	Convolutional Neural Networks
CSV	Comma Separated Values
CV	Cross Validation
DOM	Dominion Energy
DT	Decision Trees
ELT	Extract, Load and Transform
ERM	Empirical Risk Minimization
EPT	Eastern Prevailing Time
GBM	Gradient Boosting Machine
HVAC	Heating, Ventilation and Air Conditioning
k-NN	k-Nearest Neighbors
LR	Linear Regression
LSTMs	Long Short-Term Memory networks
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLPs	Multilayer Perceptrons
MW	Megawatt
PJM	Pennsylvania, New Jersey, Maryland Interconnection
RFR	Random Forest Regressor
RMSE	Root Mean Squared Error
RTO	Regional Transmission Organization
SMA	Simple Moving Average
SVR	Support Vector Regression
UCT	Coordinated Universal Time
XGBoost	Extreme Gradient Boosting

Chapter 1

Introduction

1.1 Project Overview

Energy consumption has seen a rise of 100% increase over the past 40 years, as reported by a study (Zeyu Wang et al. 2018). This increase in energy demand, combined with the consequent energy crisis, has made energy consumers to find ways of efficient energy usage. Notably, commercial buildings have emerged as the primary user of energy globally for example people spend more than 90% of their lives indoors (Zeyu Wang et al. 2018). Therefore, addressing energy consumption has become critical in the development of sustainable energy techniques.

The issue of short-term energy consumption forecasting and its function in energy control systems has been thoroughly researched in recent decades. For electric utility companies like Scottish power, accurate short-term load forecasts hold great promise since they can provide more control over operational choices, including dispatch, unit commitment, fuel allocation, and maintenance. Buildings consume the most energy out of the three main core economic sectors that are showing high energy consumption: transportation, industry, and buildings. Building HVAC systems in the United States account for about 44% of household energy consumption, whereas in the countries of the European Union, buildings account for about 40% of energy use (Jozi et al. 2019).

The relatively high energy consumption of these facilities, energy consumption forecasting may be considered important to get the best use out of energy systems, further control over an energy distribution network, and energy consumption reduction. The structures range in size from modest rooms to expansive estates, catering to a variety of clientele including residential, commercial, business, and technological

(Jozi et al. 2019). If we want to achieve net zero energy targets forecasting energy usage is vital. it will require buildings and facilities to produce just as much energy as they consume. Accurate forecasting improves the deployment of renewable energy sources and promotes a supply-demad balanced. forecasting energy consumption patterns properly, energy systems will coordinate production and consumption properly which improves the energy grid's stability and efficiency.

Machine learning, provides an opportunity to improve the accuracy and reliability of energy consumption projections. These improvements will help encourage better and longer-lasting energy management practices, which will add to the overall aims of net zero.

1.2 Problem Statement

For energy systems to function as efficiently and dependably as possible, accurate forecasting of energy usage is essential. However, there are challenges with the current forecasting techniques, especially in managing complex and unpredictable energy consumption patterns. Traditional models do not catch the nonlinear and dynamic nature of energy demand, resulting in significant forecast errors and less efficiency in energy management.

All energy consumers which consist of industrial and comercial carry its own forecasting challenges due to their different consumption patterns and how multiple variables such as customer count, weather condition, and fluctuating energy needs. This differences makes it difficult to predict energy use accurately, affects the energy distribution network, higher operating costs, and challenges in adding renewable energy sources into the grid.

Moreover, the shift to net zero energy targets and the requirement for energy network balance bring to light the shortcomings of existing forecasting methods. Achieving sustainability and energy efficiency is hampered by the inability to generate accurate and trustworthy projections, which also makes energy planning ineffective.

The creation of more reliable and accurate energy consumption forecasting models is the primary challenge this study attempts to solve. This study intends to increase the accuracy of energy consumption forecasts by utilising cutting-edge machine learning techniques. This will improve the sustainability and efficiency of energy systems and facilitate the incorporation of renewable energy sources.

1.3 Objectives

The general aim of this research is to develop and evaluate machine learning models in energy consumption prediction. It will be accomplished by setting the following specific objectives:

1. Development of machine learning models for energy forecasting: Regression, Time Series, Neural Networks, and ensemble methods will be developed solely for energy consumption prediction.

2. Performance evaluation of the model: Estimating accuracy, reliability, and robustness by appropriate evaluation metrics of ML models and Neural Network models.

3. Identifying Factors Driving Energy Consumption: To develop the ability for analyzing and pinpointing major variables that affect the energy consumption pattern for example environmental factors.

4. Real-World Case Study Using Developed Models: Apply developed models in a real-world dataset to validate its performance, quality, and practical applicability for Energy Consumption prediction.

1.4 Research Questions

This research will address the following key questions:

1. What are the most effective machine learning models for forecasting energy consumption?

2. How do machine learning models compare in terms of accuracy and reliability?

3. What are the primary factors influencing energy consumption and how can they be incorporated into forecasting models?

4. Can machine learning models developed in this study be effectively applied to real-world data for accurate energy consumption forecasting?

1.5 Scope Of Study

The scope of this study includes:

1. Data Sources: We will use the historical energy consumption data from the PJM Interconnection LLC, streamline to data from the DOM local area. The Dataset contains energy consumption from 2006 to 2024. The data is gotten through ELT process from PJM's website and is in megawatts (MW).

2. Geographical Focus: The main data used in this research comes from the DOM local area of the PJM Interconnection region of the United States.

3. Model Development: The study will develop and evaluate several machines learning models, including regression models, time series models, neural networks, and ensemble methods.

4. Evaluation Metrics: Metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R-squared (R^2), and Mean Average Percentage Error (MAPE) will be used to evaluate the performance of the model.

5. Case Study Application: To confirm the generated models' practical applicability, a case study in real time will be conducted.

1.6 Project Timeline

Figure 1.1 and Table 1.1 shows the project timeline through the various stages of the project. It aids in setting expectations, tracking progress, and ensuring that the project stays on schedule.

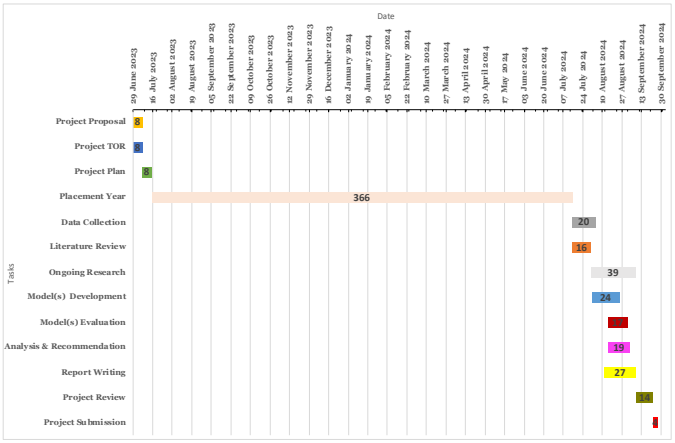


Figure 1.1: Gantt Chart Project Timeline.

Tasks	Start Date	Duration (Days)	End Date
Project Proposal	29/06/2023	8	07/07/2023
Project TOR	29/06/2023	8	07/07/2023
Project Plan	07/07/2023	8	15/07/2023
Placement Year	15/07/2023	366	14/07/2024
Data Collection	15/07/2024	20	04/08/2024
Literature Review	15/07/2024	16	31/07/2024
Ongoing Research	31/07/2024	39	08/09/2024
Model(s) Development	01/08/2024	24	25/08/2024
Model(s) Evaluation	15/08/2024	17	01/09/2024
Analysis & Recommendation	15/08/2024	19	03/09/2024
Report Writing	12/08/2024	27	08/09/2024
Project Review	09/09/2024	14	23/09/2024
Project Submission	23/09/2024	4	27/09/2024

Table 1.1: Project Timeline

1.7 Dissertation Structure

The structured of the Dissertation is shown below:

1. Introduction: This chapter contains an introduction to the subject, problem statement, aims, objectives, research questions, scope of the study, and project timeline

2. Literature review which discusses the time series, traditional and machine learning methods for energy consumption forecasting, and key challenges facing energy consumption.

3. Methodology, focusing on the methodology used at the different stages of the experimentation which are data collection, preprocessing, model development, and validation.

4. Results and Discussion: The performance of different models are analysed. Also compared their strengths, limitations, and the impact of feature engineering and hyperparameter tuning.

5. Case study: The top three models from the result and discussion chapter were applied the to real-world/Current data to evaluate their performance and practical applicability.

6. Futher Work: This chapter focuses on potential areas for future research and suggests improvements that could improve forecasting accuracy.

7. Conclusion: Summarizes the findings made within the dissertation and discusses if the objectives set within the study have been achieved

.

Chapter 2

Literature Review

2.1 Introduction

Chapter 2 begins with a thorough review of the literature to shed light on the application of machine learning to energy consumption forecasting. This critical analysis of academic literature reveals a variety of viewpoints and approaches, shedding light on the intricacies, developments, and possible solutions in the field of energy forecasting.

The combination of many academic perspectives and practical information provides the foundation for the study's later stages. This chapter seeks to gather important insights that will guide the creation of a reliable energy consumption forecast model through a thorough investigation. The purpose of such a model is to provide relevant parties with the means to confidently and in advance navigate the ever-changing terrain of energy demand.

Using a wide variety of research materials, the following topics will be explored in this chapter:

- Introduction to Energy Consumption Forecasting
- Time Series Analysis in Energy Forecasting
- Traditional Forecasting Methods
- Machine Learning in Energy Forecasting
- Key Challenges in Energy Consumption Forecasting
- Summary

2.2 Introduction To Energy Consumption Forecasting

Energy consumption forecasting is a very important for predicting future energy needs. To this effect, accurate energy consumption is useful to all stakeholders, including end-users, system operators, policy framers, and economic planners. It helps in the effective management of the electrical grid, optimization of energy use, and mitigation of environment-related effects.

2.2.1 Background Information on Energy Usage Forecasting

Energy use forecasting refers to the act of establishing how much energy will be consumed in the future based on trends, weather, economic activity, and technology. This helps in planning for any future needs in the generation and distribution of energy. The various methods and models used to forecast energy consumption vary from statistical techniques to machine learning models, each with its strengths and limitations.

For instance, the stochastic approach, as discussed by Arghira et al. 2013, involves forecasting the consumption of electrical appliances in residential settings. This method is particularly challenging due to the variability and unpredictability of individual appliance usage. By segmenting and aggregating data, the precision of energy forecasts can be enhanced, making it a vital tool for managing power flow in the electrical grid.

Similarly, time series forecasting techniques, as reviewed in Deb et al. 2017, are extensively used for predicting building energy consumption. These models are crucial for real-time energy monitoring and efficient building operation, highlighting their importance in ensuring energy efficiency and fault detection in building systems.

2.2.2 Importance of Accurate Energy Forecasts

Accurate energy forecasts are indispensable for several reasons:

1. Policy-making: Accurate forecasts help governments and other regulatory bodies in making effective and sound energy policies, realistic targets relating to the adoption of renewable energy, and enable them to take due measures for reducing greenhouse gas emissions.

2. Economic Planning: Energy forecasting is very important in terms of economic planning and investment decisions. The forecast assists in estimating the trend in

energy demand, which then guides decisions on energy production, infrastructure development, and market strategies.

3. Environmental Implication: Energy forecasting can protect the environment through better management of the available energy. With proper forecasts, there is a limited need to apply fossil fuel energy sources, leading to lower carbon emissions and reduced effects of climate change.

U. F. Akpan and G. E. Akpan 2012 emphasizes the link between energy consumption and climate change, illustrating how the rise in fossil fuel use has led to increased CO2 emissions. Accurate energy forecasting can aid in implementing mitigation mechanisms such as improved energy efficiency, carbon emission taxes, and investment in renewable energy technologies.

2.3 Time Series Analysis In Energy Forecasting

A time series consists of data points collected or recorded at consistent intervals, representing the evolution of a variable over time (Cryer and Kellet 1991). An example is energy consumption, with intervals/settlement periods ranging from 30 mins to 1 hour. Typically, these data points are visualized through line graphs or time-series plots. Time series data is crucial across various industries such as economics, finance, Energy and operations management, where it aids in identifying trends, understanding patterns, and making predictions.

2.3.1 Components of Time Series Data

Time series data comprises several key components that reflect its inherent patterns or structures. Understanding these components is essential for accurate analysis and forecasting. These components include trends, seasonality, and noise (Cryer and Kellet 1991).

Trend

The trend refers to a long-term movement within the data showing the general increase or decrease over a period. In fact, identification and modeling of a trend are among the most important issues in fundamental time series analysis, as a trend reveals the

direction, and the quantitative size of changes initiated. Trends can manifest in various forms:

- Upward Trend: Indicates a general increase in data values over time.
- Downward Trend: Indicates a general decrease in data values over time.
- Horizontal Trend: Shows little to no change in data values over time.
- Non-linear Trend: Exhibits complex patterns with changes in direction or rate.
- Damped Trend: Shows a decreasing rate of change over time.

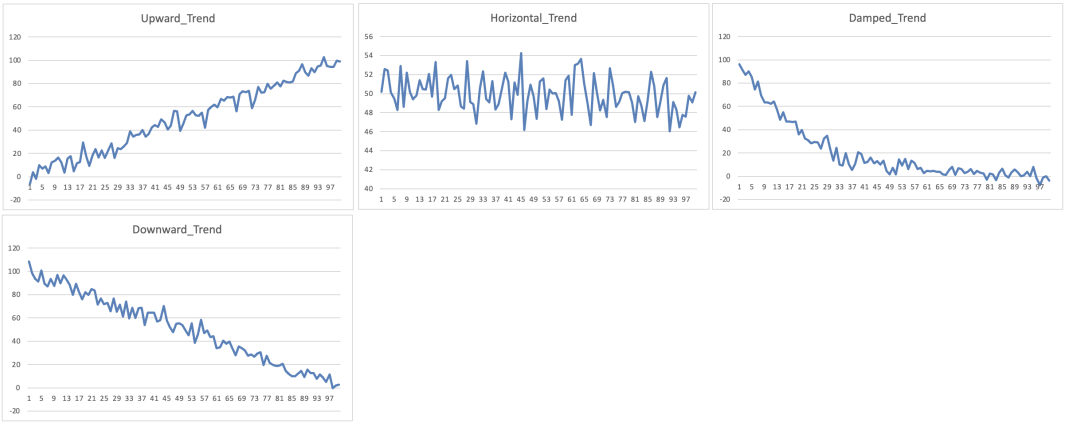


Figure 2.1: Showing the various forms of trends.

It is important to know that one can have multiple trends in a time series. Precise detection and modeling of these trends immensely enhance the accuracy of the predictions and correct interpretation of the patterns.

Seasonality

Seasonality refers to periodic patterns that repeat over periods of fixed length, for example, daily, weekly, monthly, or yearly. In general, it is driven by events occurring at regular intervals or by some cycles of nature. In energy demand forecasting, seasonality appears in the periodic regularity of demand:

- Weekly Seasonality: Patterns that repeat every seven days, commonly seen in energy consumption or sales data.

- Monthly Seasonality: Patterns that repeat every 30 or 31 days, often observed in sales or weather-related data.
- Patterns that repeat every year, relevant to agricultural practices, travel trends, or sales cycles.

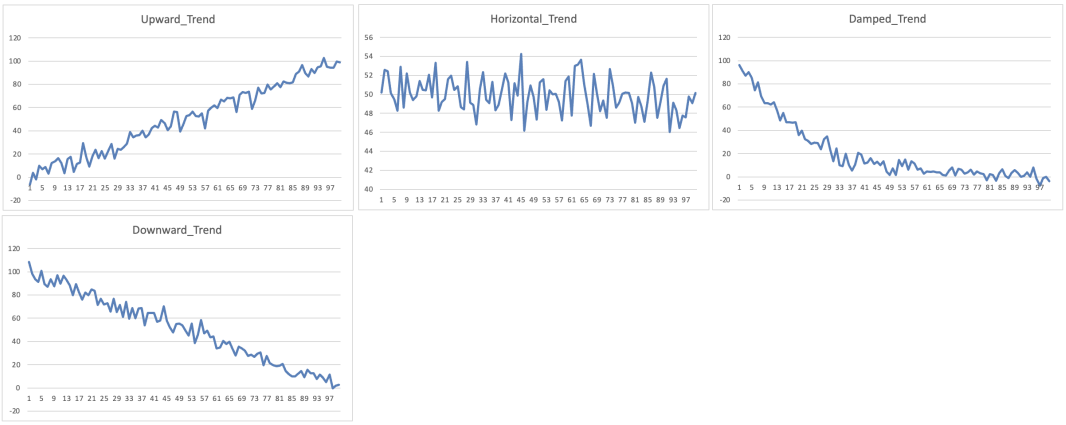


Figure 2.2: The various forms of Seasonality.

Noise

In time series data, noise refers to any form of random fluctuation or variation that cannot be attributed to a form of underlying pattern or trend. Irregularities can result from measurement errors, data processing errors, or simply plain randomness in what is measured. These noises hide the trends and patterns of the data, hence making analysis harder. Therefore, noise reduction or removal is very essential in time series analysis, in order to obtain sharper and more accurate forecasts.

Basically, time series data analysis involves understanding and modeling its core components: trends, seasonality, and noise. Such recognition will provide a greater accuracy of predictions, through an improved understanding of the basic patterns and relationships within the data for better decision-making. It provides the means to focus and conduct analysis on these core components, bettering understanding of the data and leading to better forecasting in energy and other sectors.

2.4 Traditional Forecasting Methods

Traditional approaches to time series analysis and forecasting have been based on systematic methods developed from very well-established statistical theories. Such methods are used to predict future values using historical variables. Some of the most popular traditional models used in forecasting include ARIMA (AutoRegressive Integrated Moving Average), linear regression models, etc (Pourahmadi 2001). These models have been applied to a huge number of areas, including energy use forecasts, proving their efficiency in catching temporal patterns and trends. This section reviews the history of evolution, major principles, and applications of these traditional methods of forecasting in an energy consumption context.

2.4.1 Historical Development and Key Principles

ARIMA Models

According to Ljung, Ledolter, and Abraham 2014 the ARIMA technique was popularized by Box and Jenkins in the 1970s and is among the most popular techniques for time series forecasting. This technique combines three components: autoregression, differencing to achieve stationarity, and moving average. The general form of an ARIMA model which is inspired from Ospina et al. 2023 is denoted as ARIMA(p,d,q), where:

- p represents the number of autoregressive terms,
- d represents the number of non-seasonal differences needed for stationarity
- q represents the number of lagged forecast errors in the prediction equation.

The mathematical representation of an ARIMA model is given by:

$$Y_t = \mu + \sum_{i=1}^{p} \alpha_i Y_{t-i} + \sum_{j=1}^{q} \theta_j e_{t-j} + e_t \tag{2.1}$$

where Y_t is the time series value (after differencing if $d > 0$)), μ is the mean, α_i and θ_j are coefficients of autoregressive and moving average terms, respectively, and e_t is the error term assumed as white noise.

where Y_t is the time series value (after differencing if $d > 0$)), μ is the mean, α_i and θ_j are coefficients of autoregressive and moving average terms, respectively, and e_t is the error term assumed as white noise.

Linear Regression

Another core element of traditional approaches to prediction is linear regression. It is one method of predicting a dependent variable from one or more given independent variables (Kumar et al. 2023). The simple linear regression equation takes the following form:

$$Y_t = \beta_0 + \beta_1 X_t + \epsilon_t \tag{2.2}$$

where Y_t is the dependent variable, X_t is the independent variable, β_0 and β_1 are the coefficients, and ϵ_t is the error term.

Linear regression is most effective in those cases where it is presumed that there is a linear relationship between variables.

Simple Moving Average (SMA)

A simple Moving Average is a basic, yet effective, tool for forecasting time series data. It smoothes out data fluctuations, highlighting longer-term patterns by averaging throughout a set number of past observations. The formula for an n-period SMA is given by the following formula:

$$\text{SMA}_t = \frac{1}{n} \sum i = 0^{n-1} Y_{t-i} \tag{2.3}$$

where SMA_t is the moving average at time t, $Yt - i$ are the past observations, and n is the number of periods.

SMA is particularly appropriate for showing trends over some predefined period; however, it lags behind the sudden changes in the data.

2.4.2 Applications in Energy Consumption Forecasting

ARIMA models have been used in energy consumption forecasting because of their capacity to capture the autocorrelation structure of time series data. For example, the

ARIMA approach was used in one of the studies to forecast energy consumption in Turkey by generating future values with regard to coal, oil, natural gas, renewable energy, and total energy consumption. Specifically, the ARIMA models selected were ARIMA (1,1,1) for coal consumption and ARIMA (0,1,0) for oil consumption, since these minimized the Akaike Information Criterion, as can be seen in S. Ozturk and F. Ozturk 2018. These forecasts obtained from the study pointed to a continuous increase in energy consumption in Turkey, hence increasing dependency on imported energy, thus confirming the need for strategic energy planning.

ARIMA was also used in forecasting South Africa's energy consumption. The results indicated high reliability, with the ARIMA model providing accurate predictions. The study projected a 7.49% annual growth rate in energy consumption over the next 14 years, offering critical insights for policy adjustments in energy supply and demand management (Ma and Zhuangzhuang Wang 2019).

Comparative studies have demonstrated the effectiveness of ARIMA models relative to other traditional forecasting methods. For example, a study on China's primary energy consumption used both ARIMA and GM (1,1) models. The ARIMA model provided forecasts that were less responsive to short-term fluctuations, adhering more closely to long-term trends. Conversely, the GM (1,1) model, which relies on the most recent data points, was more sensitive to recent changes. A hybrid model combining ARIMA and GM (1,1) outperformed both individual models, offering more accurate forecasts with lower Mean Absolute Percentage Error (MAPE) (Yuan, Liu, and Fang 2016).

Other traditional models effectively applied in energy consumption forecasting include exponential smoothing and Holt-Winters. In Universiti Tun Hussein Onn Malaysia, various techniques of time series forecasting, like Simple Moving Average, Simple Exponential Smoothing, and Holt-Winters, were applied to forecast monthly electricity consumption. It accounted for both trend and seasonal patterns; it therefore showed, in this context, very low forecasting errors and represented the decreasing trend of electricity consumption very well due to faculty relocations to other places, as shown in Y. Lee, Tay, and Choy 2018.

It is for these reasons that conventional methods of forecasting, like ARIMA, SMA, and linear regression, become very important in time series analysis and forecasting. Applications to energy consumption forecasting shows their strength in capturing temporal patterns and trends. These models give an exact forecast and more importantly give insight into the underlying dynamics, which helps greatly in strategic planning

and policy formulation. Enough evidence of the relevance and efficiency to deal with very complex problems of energy consumption forecasting is represented by the use of these models in many studies in this field.

2.5 Machine Learning In Energy Forecasting

The introduction of machine learning into energy forecasting methods has been an important step in evolution regarding how humanity addresses the challenge of predicting and managing energy consumption. Machine learning, defined by Arthur Samuel as that domain of the art which gives computers the ability to learn without being explicitly programmed, has made a turn from a conceptual innovation to a practical tool commonly applied in various industries. ML relies on a host of algorithms that tend to be suited for various types of data problems, making it highly adaptable to the complex, varied nature of energy use patterns. Such algorithms can extract meaningful information from large data sets, which would have otherwise been impracticable, if not impossible, to do with traditional methods (Mahesh 2020).

A machine is said to be learning from past experiences (i.e., data fed into it) if its performance in each task improves over time (Mahesh 2020). This principle supports the application of ML in energy forecasting, where algorithms learn from historical energy consumption data to predict future trends. For instance, just as a machine might predict a customer's purchase behaviour based on previous data, ML models in energy forecasting predict consumption patterns based on historical energy usage. The ability of these models to continuously improve as they are exposed to more data makes them especially valuable in this field.

The choice of the right machine learning algorithms is very influential since the scenarios of energy consumption may change from residential to the industrial sectors. There is, hence, no one-size-fits-all solution. The choice of an algorithm depends upon the general nature of the forecasting task, such as the characteristics of data and desired accuracy. Linear regression, decision trees, support vector machines, and neural networks are some of the commonly used algorithms in energy use. Each of these algorithms offers different strengths depending on the context, such as the simplicity of linear regression for basic tasks or the sophistication of neural networks for handling complex, nonlinear data relationships.

2.5.1 Supervised Learning

Supervised machine learning is a widely utilized technique in various fields, including energy forecasting, where it plays a crucial role in predicting energy consumption patterns. At its core, supervised learning involves the task of learning a function that maps inputs to outputs based on labelled data (Cunningham, Cord, and Delany 2008). This method relies on example input-output pairs, where the input dataset is divided into training and testing sets (Mahesh 2020). The algorithm learns from the training dataset, which includes the correct output for each input, and then applies this learned function to make predictions or classifications on the test dataset.

Workflow of Supervised Machine Learning

Labelled datasets are at the core of supervised learning, where for each example in the dataset, its related correct output is known. Input data are fed to the algorithm during the training exercise, with the underlying patterns and relationships between the entries learned by the algorithm. Once trained, the model is tested on a separate dataset to evaluate its performance (Jiang, Gradus, and Rosellini 2020). Inspired by Mahesh 2020 Figure 2.3 illustrates a typical workflow of supervised machine learning algorithms, from data preparation to model evaluation.

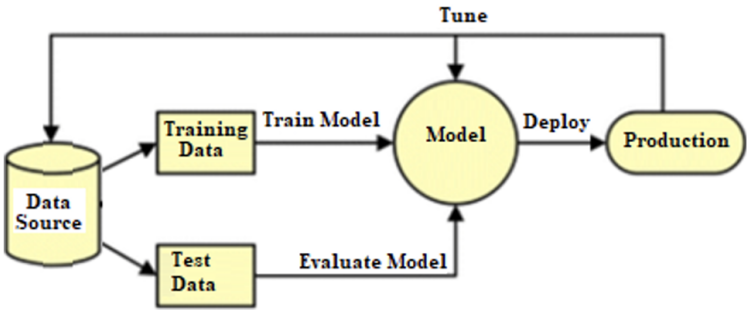


Figure 2.3: Typical workflow of supervised machine learning algorithms.

Types of Supervised Learning Algorithms

Supervised learning algorithms are typically divided into two main categories: regression and classification. These categories determine the type of output the model is designed to predict.

Figure 2.4 showing Regression and classification are key supervised learning techniques used in machine learning. Regression predicts continuous values by identifying

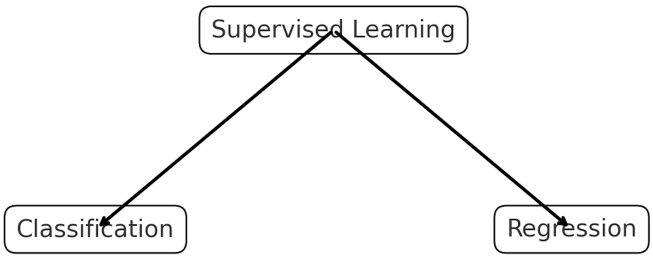


Figure 2.4: Types of supervised machine learning.

relationships between input features and output variables, commonly applied in energy forecasting to estimate future consumption. Classification predicts categorical outcomes by assigning data to predefined classes, such as categorizing energy usage patterns. Both methods use algorithms such as Linear Regression, Decision Trees, and Neural Networks. Ensuring high-quality and accurate predictions means requiring large volumes of corresponding data.

The most important factors in training a model through supervised learning are the amount and quality of the labelled data used. Any trained model's accuracy thus directly depends on the quality of the dataset used for training. Any regression task or classification task involving large, well-represented datasets can be very useful to machine learning models in capturing underlying data complexities (Jiang, Gradus, and Rosellini 2020).

The following Figure 2.5 illustrates some of the common supervised machine learning algorithms and their applications for regression and classification.

Avg. Energy Consumption (kWh)	Time of Day	Season	User Type	Usage Category
5.2	Morning	Summer	Residential	High
3.8	Afternoon	Winter	Commercial	Low
7.5	Evening	Fall	Industrial	High

Classification

Time of Day	Season	User Type	Avg. Energy Consumption (kWh)
Morning	Summer	Residential	5.2
Afternoon	Winter	Commercial	3.8
Evening	Fall	Industrial	7.5

Regression

Figure 2.5: Examples of Classification and Regression.

The Figure 2.5 above explains how classification tasks differ from regression because they predict a categorical outcome based on input features. A classification task is designed to basically assign a record to one of a set of predefined categories or classes. For instance, using features such as average energy consumption, time of day, season, and user type, a classification model could put energy usage into "High" or "Low" categories. The model predicts the appropriate class label for each record based on the input features.

On the other hand, regression focuses on predicting a continuous numerical value

rather than a category. For example, using the same features—time of day, season, and user type—a regression model would predict the exact average energy consumption (in kWh) rather than categorizing it. The predicted output would be continuous in nature and would tell exactly how much energy is to be used under certain specified input conditions.

Basically, classification labels each record, like ”High” or ”Low” energy consumption, while regression predicts a continuous value, such as the exact amount of energy consumed.

2.5.2 Linear Regression

As already explored in traditional forecasting, linear regression is a straightforward yet powerful tool in supervised machine learning for predicting continuous output values. The model establishes a linear relationship between input features and the target variable, making it ideal for scenarios where such a relationship is assumed or known. While its simplicity makes it easy to interpret and apply, linear regression may have limitations in handling complex or nonlinear datasets.Tunç, Çamdali, and Parmaksizoğlu 2006 used linear regression model to predict electricity consumption in Turkey based on population and per capita consumption rates. It, therefore, proved the efficiency of linear regression in energy forecasting, especially in situations where there is a linear relationship between variables and when the relationship is simple.

Simple linear regression is visually represented in the figure below, where the relationship between a single input feature and the output variable is graphically depicted.

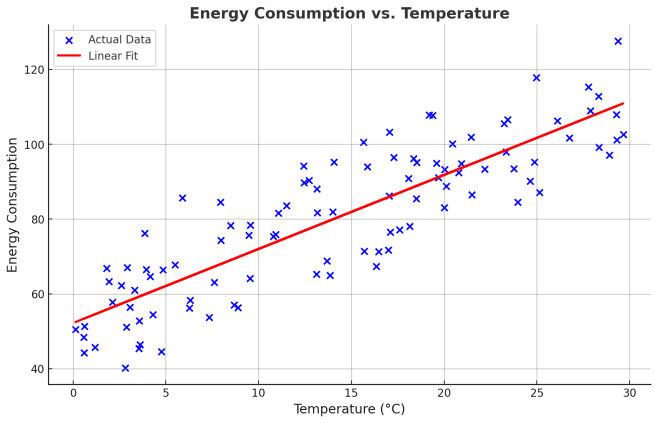


Figure 2.6: Linear Regression.

It plots the line as shown in Figure 2.6 that the model has fitted through the data points to show the strength and direction of the relationship between the variables.

2.5.3 Decision Tree

Decision Trees are a popular supervised learning algorithm known for their simplicity and interpretability. The model segments data through a series of rule-based splits, creating an empirical tree that serves as a predictive tool (Tso and Yau 2007). One of the main advantages of decision trees is their ability to produce interpretable rules, making the logic behind predictions transparent. This interpretability is particularly valuable when handling both continuous and categorical variables, as the model provides clear insights into the importance of each factor (Tso and Yau 2007). However, decision trees have limitations, particularly in dealing with non-linear data and noisy inputs, where they may underperform compared to more complex models like neural networks. Additionally, they are generally better suited for categorical outcomes and less effective for time series data without clear patterns (Tso and Yau 2007).

In practical applications, decision trees have proven useful in tasks such as model selection for intelligent maintenance systems. By incorporating relevant variables, decision trees have significantly reduced forecasting errors, demonstrating their practical value (Shcherbakov, Kamaev, and Shcherbakova 2013). Decision Tree Regression, which is like its classification counterpart, is designed for continuous data. It uses criteria like Mean Absolute Error (MAE) to guide the splits, and careful management of tree depth is crucial to balance bias and variance, optimizing the model's accuracy (Gupta, Bansal, Roy, et al. 2021). The Figure 2.7 below shows a typical DT

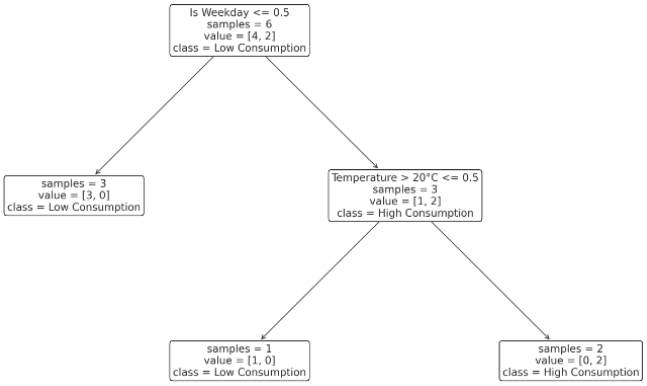


Figure 2.7: Example of a Decision Tree relating to Energy Forecasting.

Figure 2.7 shows a possible decision tree in graphical form for the purpose of

energy forecasting. The tree splits data according to different features such as temperature if it's a weekday, and occupancy. Each path from this tree forms a conclusion on whether the energy consumption will be high or low.

- Temperature $>$ 20° C: The first decision point is taken if the temperature is greater than 20° C.
- Is Weekday: Based on the temperature, the next decision is whether it is a weekday.
- Occupancy $>$ 50%: Finally, the tree considers occupancy level.

Each leaf node at the bottom denotes the predicted energy consumption, either high or low, based on the conditions met as you move down the tree.

2.5.4 Random Forest Regressor (RFR)

The Random Forest Regressor (RFR) is an ensemble learning method that builds upon the simplicity of decision trees by aggregating the results of multiple trees to enhance predictive accuracy and robustness. The model requires tuning of key parameters such as the nodesize, which influences the depth of the trees, and the number of trees (ntree) in the forest (Zeyu Wang et al. 2018). While Random Forest is highly effective in handling both continuous and categorical variables, its primary strength lies in its ability to reduce overfitting by averaging multiple trees. However, this averaging process may lead to a loss of sensitivity to intricate patterns in the data, particularly when compared to boosting methods (Zeyu Wang et al. 2018). The versatility of Random Forest makes it applicable across various fields, including biology, medicine, and business, where it is valued for its interpretability and ease of use (Shi et al. 2018).

2.5.5 Gradient Boosting Machine (GBM)

Gradient Boosting Machine (GBM) builds upon the principles of boosting by constructing models sequentially, where each new model attempts to correct the errors made by its predecessor. This iterative approach allows GBM to capture complex patterns in the data, making it more powerful than Random Forest in many scenarios (Touzani, Granderson, and Fernandes 2018). In GBM, weak learners—often decision trees—are incrementally added to minimize a loss function, such as Mean Squared

Error (MSE) in regression tasks (Touzani, Granderson, and Fernandes 2018). A crucial aspect of GBM is the learning rate (shrinkage parameter α), which controls the contribution of each tree to the overall model. Although a smaller learning rate can lead to higher accuracy, it also requires more iterations, increasing computational demands and the risk of overfitting (Touzani, Granderson, and Fernandes 2018). GBM has proven particularly effective in applications like time series forecasting, where it outperforms traditional statistical models by capturing intricate patterns that might be missed otherwise (Di Persio and Fraccarolo 2023).

2.5.6 Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) is an advanced implementation of the gradient boosting framework, designed to enhance computational efficiency and model performance. Among others, XGBoost innovated with approximate grid search and parallel learning, which significantly enhanced both its computational speed and scalability relative to conventional GBM (Sauer et al. 2022). This algorithm is particularly well-suited for tasks that require high computational efficiency and scalability, such as large datasets or environments where speed is critical, like Kaggle competitions (Bassi et al. 2021). XGBoost supports a wide range of objective functions, including regression and classification, and excels at handling complex, non-linear data relationships, as well as anomalies (Sauer et al. 2022). When applied to specific tasks, such as energy consumption predictions, XGBoost has been shown to outperform other gradient boosting variants like LightGBM and CatBoost, demonstrating its superior ability to generalize from complex datasets (Bassi et al. 2021).

When comparing these ensemble methods, each has distinct strengths and limitations. Random Forest Regressor is known for its robustness, ease of interpretation, and relatively simple tuning process, making it ideal for situations where data is noisy or where model interpretability is crucial (Shi et al. 2018). However, it may not capture complex patterns as effectively as boosting methods. Gradient Boosting Machine, with its iterative approach, is more powerful for scenarios requiring the modelling of complex dependencies, but it demands careful parameter tuning, particularly of the learning rate, and can be computationally intensive (Touzani, Granderson, and Fernandes 2018). XGBoost provides better computational efficiency and scalability compared to GBM; hence, it is the best option where speed and accuracy are needed in large-scale applications (Sauer et al. 2022). Each algorithm serves different needs, and choice among them should be guided by the requirements of the task in terms of balancing

accuracy, interpretability, and computational resources.

2.5.7 Support Vector Regression (SVR)

Support Vector Regression is an extension of the Support Vector Machine technique tailored for regression tasks. The principle behind Structural Risk Minimization is to reduce an upper limit on generalization error, which is a combination of both training error and a confident level. This approach contrasts with the Empirical Risk Minimization (ERM) principle, commonly employed by traditional neural networks, which focuses solely on minimizing the training error (Dong, Cao, and S. E. Lee 2005). As a result, SVR often achieves higher generalization performance compared to these traditional methods, making it a robust choice for a wide range of regression problems (Dong, Cao, and S. E. Lee 2005).

A distinctive feature of SVR is that its training process is equivalent to solving a linearly constrained quadratic programming problem. This ensures that the solution is always unique and globally optimal, unlike other methods that might get trapped in local minima during the training process (Dong, Cao, and S. E. Lee 2005). The model's efficiency is further enhanced by its reliance on support vectors—a subset of the training data—rather than the entire dataset, allowing it to achieve the same solution with less computational overhead. However, one of the drawbacks of SVR is its computational complexity, as the training time scales between quadratic and cubic with respect to the number of training samples. This can be a significant limitation when dealing with large datasets (Dong, Cao, and S. E. Lee 2005). Nonetheless, in scenarios where the dataset size is manageable, SVR remains an effective and precise tool for regression analysis.

SVR is a powerful technique that balances the trade-offs between model complexity and generalization ability, offering a globally optimal solution through its unique training process. However, its application is best suited to problems where computational resources can accommodate its more demanding training requirements, particularly when the dataset is not excessively large.

2.5.8 Ensemble Learning

Ensemble learning represents a sophisticated approach within machine learning, where multiple models are combined to achieve superior predictive performance compared to individual standalone models. This technique leverages the strengths of various

algorithms, mitigating the weaknesses that any single model might have, and is widely recognized for its ability to improve both classification and regression outcomes (Khan et al. 2024). One of the most common ensemble methods is the voting ensemble, which can be applied to both classification and regression tasks. In classification, this method is known as the voting classifier, and in regression, it is referred to as the voting regressor (Khan et al. 2024).

The voting ensemble works by aggregating the predictions from multiple base models. For classification tasks, the final prediction is determined by a voting mechanism, which can be either hard or soft voting. Hard voting relies on majority voting, where the class label that receives the most votes from the individual classifiers is selected as the final prediction. On the other hand, soft voting involves averaging the predicted probabilities from each classifier, and the class with the highest average probability is chosen (Khan et al. 2024). Khan et al. 2024 shows in Figure 2.8 a typical Ensemble learning

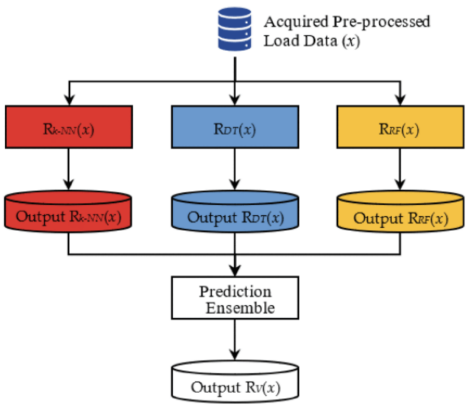


Figure 2.8: Example of an Ensemble Learning.

This approach allows the ensemble to capitalize on the strengths of each model, resulting in more robust and accurate predictions.

For regression tasks, the voting ensemble uses an averaging method to combine the predictions from multiple models. This can be done through either a simple average or a weighted average. In a simple average, the predictions of all models are averaged equally for each instance in the test dataset. Conversely, the weighted average assigns different weights to the predictions from each model, with the aim of giving more importance to models that perform better, thus enhancing the overall prediction accuracy (Khan et al. 2024). This weighted approach is particularly useful when the base models have varying levels of performance, as it allows the ensemble to produce more reliable predictions by emphasizing the contributions of the stronger models.

In practical applications, ensemble learning has been implemented using various combinations of models, such as Decision Trees (DT), Random Forest (RF), and k-Nearest Neighbors (k-NN). These models are combined to form a cohesive ensemble that benefits from the diverse strengths of each individual algorithm. The layout of such an ensemble technique, as applied in specific research contexts, typically involves a systematic approach to combining these models to maximize predictive accuracy, as depicted in the relevant figures of the research work (Khan et al. 2024).

As we can see ensemble learning is a powerful strategy in machine learning that enhances model performance by combining multiple models into a single, more robust predictor. The voting ensemble method, with its applications in regression, exemplifies the effectiveness of this approach, particularly when leveraging different voting and averaging strategies to optimize predictions

2.5.9 Neural Networks

Neural networks have been strongly influenced, above all, by the structure of the human brain and are, therefore, quite arguably a cornerstone in modern machine learning, especially for tasks requiring modeling complex, nonlinear relationships. A neural network is basically an interconnection of layers of artificial neurons processing inputs to obtain an understanding of the underlying patterns and make a prediction. The key advantage of neural networks lies in their ability to adapt to changing inputs, making them versatile across a wide range of applications, including trading systems and energy management (Mahesh 2020). For instance, in smart grids, neural network-based models have demonstrated high performance in load forecasting, particularly when the appropriate hyperparameters, such as the type of activation function and the number

of hidden layers, are carefully selected. Recent studies show that neural networks with scaled exponential linear units and five hidden layers provide superior accuracy in energy consumption forecasting (Moon et al. 2019). Mahesh 2020 shows in Figure 2.9 a typical neural network

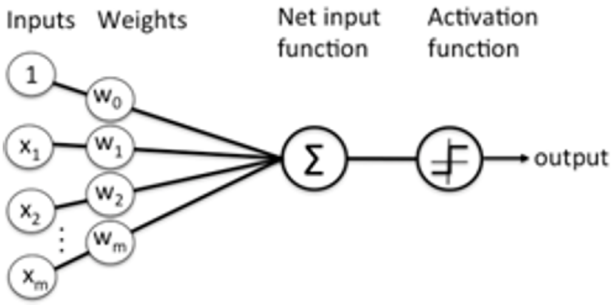


Figure 2.9: A typical Neural Network.

Multilayer Perceptrons (MLPs)

Multilayer Perceptrons are the simplest and most powerful family of artificial neural networks that excel in numerous tasks. A typical MLP will have an input layer, one or a few hidden layers, and an output layer. In such an architecture, every neuron of one layer is fully connected to the neurons of the next layer (Kuo and Huang 2018). The MLP is particularly effective in energy load forecasting, where the input layer receives past energy load data, and the output layer predicts future loads. Despite its relatively simple structure, the MLP has proven effective in numerous applications, largely due to its ability to model complex patterns through its hidden layers (Kuo and Huang 2018). Kuo and Huang 2018 in Figure 2.10 explained with a diagram a typical multilayer perception (MLPs).

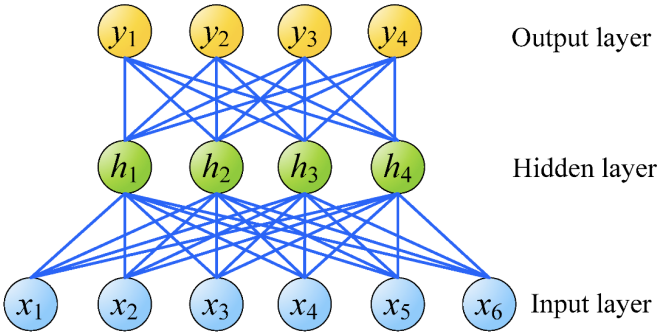


Figure 2.10: A typical Multilayer Perceptrons (MLPs).

Convolutional Neural Networks (CNNs)

While MLPs are effective, Convolutional Neural Networks (CNNs) offer enhanced performance, particularly in tasks involving highly non-linear data, such as image recognition and energy load forecasting. CNNs introduce the concept of weight sharing through convolutional layers, which reduces the number of parameters and improves computational efficiency (Kuo and Huang 2018). For instance, in energy load forecasting, CNNs are used to extract important features through convolution and pooling operations, leading to more accurate predictions. The ability of CNNs to capture spatial hierarchies in data makes them particularly powerful for complex prediction tasks (Kuo and Huang 2018). This can be graphically explained by Figure 2.11 which is inspired by Kuo and Huang 2018.

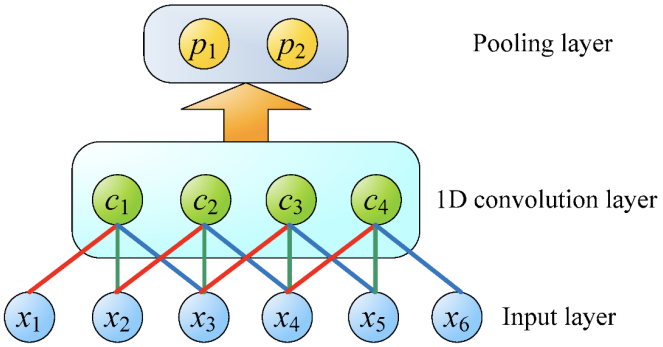


Figure 2.11: A typical Convolutional Neural Networks (CNNs).

Recurrent Neural Networks (RNNs) and LSTMs

RNNs, especially LSTMs, are specialized for sequential data and time series forecasting. LSTMs design out the vanishing gradient problem in traditional RNNs so they can handle the learning of very long-term dependencies (Kuo and Huang 2018).The structure of LSTMs includes memory cells and gating mechanisms that control the flow of information, allowing the network to retain important information over extended sequences. This makes LSTMs particularly effective for forecasting energy consumption, where long-term trends and seasonal variations are crucial (Kuo and Huang 2018). Kuo and Huang 2018 shows a typical Long Short-Term Memory networks (LSTMs) in Figure 2.12

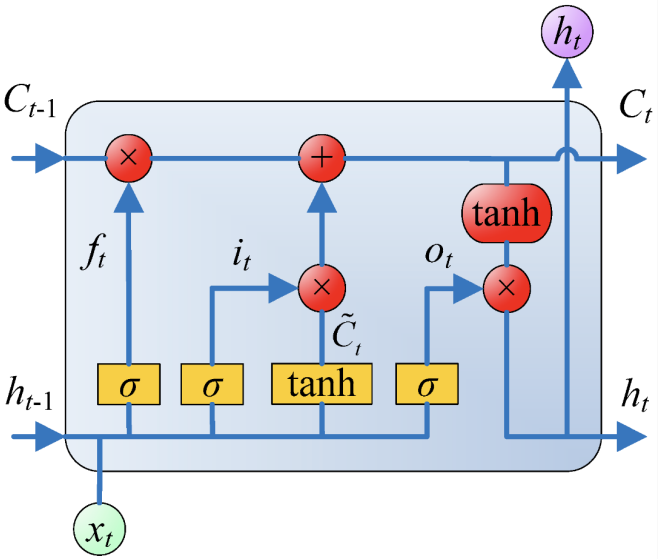


Figure 2.12: A typical Long Short-Term Memory networks (LSTMs).

Summary

In this section, we explored various neural network architectures and their significance in energy consumption forecasting. Multilayer Perceptrons (MLPs) serve as foundational models, offering simplicity and effectiveness in pattern recognition through their fully connected layers. Convolutional Neural Networks (CNNs) provide further advancements by introducing convolutional layers to capture spatial hierarchies, making them particularly adept at handling complex, non-linear data. Meanwhile, Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) excel in modeling temporal dependencies, crucial for tasks like time series forecasting in energy systems. Each of these types has its own advantages, and their results are heavily dependent on the task and hyperparameter tuning involved. The emerging new neural networks architecture has greatly improved the accuracies for energy consumption forecasting, showing their importance in current predictive analytics.

2.6 Key Challenges In Energy Consumption Forecasting

Predicting energy consumption, especially in an industrial context is challenging because of a plethora of problems that exert impact on the trueness and correctness properties of prediction models. These are multi-faceted problems, encompassing everything from data quality to the uncertainty in modeling and the intrinsic limitations of existing forecasting algorithms.

One of the foremost challenges in this domain is data fusion, a process that critically impacts the precision of energy consumption forecasting models. In industrial environments, data quality is frequently compromised due to the disparate levels of technological advancement across enterprises. This variation results in inconsistent data collection frequencies and sensitivities, often exacerbated by the harsh conditions of industrial production environments. For instance, data collection devices might be affected by factors such as temperature fluctuations, vibrations, and other environmental interferences, leading to high data confusion and a low signal-to-noise ratio. These conditions make it challenging to develop a universal data-cleaning model that can be effectively applied across different industrial sectors. Consequently, extensive preprocessing is required before data can be used for model development, which is both time-consuming and resource intensive. Moreover, even with thorough preprocessing, discrepancies in data processing standards between enterprises can lead to divergent forecasting results, thereby complicating efforts to enhance forecasting accuracy (Hu and Man 2023).

Uncertainty in forecasting represents another critical challenge, emerging from both model-related and data-related factors. Industrial processes are inherently complex, involving numerous parameters that can vary significantly over time. This complexity introduces substantial uncertainty into the data, which in turn affects the reliability of forecasting models. While there has been work in developing probabilistic models to handle model uncertainty, these methods are often complex and demanding of intricate training mechanisms. However, the complexity of handling these models is a major hurdle to their broad release. Furthermore, non-interpretable ML models make this even harder. This may imply that interpretable models, which give visibility into the relationship between inputs and predictions can be used to better understand and navigate through these uncertainties, pave a way for more robust forecasts (Hu and Man 2023).

The core algorithms employed in energy consumption forecasting also present considerable challenges. Most of the models in the literature are based on data-driven methods, which have the advantage of not necessarily being based on an in-depth understanding of the underlying industrial process. Most of them have been heavily dependent on the presence of huge and high-quality datasets. Their accuracy often deteriorates as the number of forecasting steps increases. This decline is mostly due to the fact that there exist some levels of limitations to data collection, in which key features of the data are not well captured, hence leading to incomplete or inaccurate prediction. Deep learning methods, despite their advanced capabilities, have not yet shown a significant improvement in forecasting accuracy over traditional hybrid intelligence algorithms. This shortfall is largely attributed to the difficulties in extracting sufficient data features and the resultant high uncertainty in the models' predictions (Hu and Man 2023).

The development of accurate and reliable energy consumption forecasting models is impeded by several key challenges, particularly concerning data quality, uncertainty in modeling, and the limitations of current algorithms. Addressing these challenges requires a comprehensive approach that integrates improved data collection methods, more effective uncertainty management strategies, and the development of robust core algorithms capable of autonomous learning and adaptation. By focusing on these areas, it is possible to enhance the precision and reliability of energy consumption forecasts, thereby supporting more informed decision-making in industrial settings (Hu and Man 2023).

2.7 Summary

The paper provides a comprehensive review of methodologies and challenges in energy consumption forecasting, focusing on applications of machine learning methodologies. This review outlines the evolution of energy forecasting-from simple traditional approaches like ARIMA and linear regression to sophisticated machine learning models such as neural networks and ensemble methods.

Energy consumption has traditionally been forecasted using time series analysis and other statistical models. Although robust, these models are often very weak in capturing those complex, nonlinear relationships within the data that machine learning models thrive on. The machine learning approaches to this, involving especially

supervised learning, are decision trees, random forests, and neural networks that considerably improve the predictive accuracy by learning patterns from large datasets and adapting to new ones.

Some of the major challenges in energy consumption forecasting, such as poor data quality, uncertainty in modeling, and limitations of the algorithms available so far, are also focused on in the review. These are pointed out as very critical obstacles that need resolution to improve forecasting accuracy.

In summary, literature suggested that, though traditional methods served as a foundation, the introduction of machine learning techniques offers a promising way of achieving higher forecast accuracy and reliability in energy consumption. However, the challenges identified are not going to allow the full development of their potential in practical applications without further developments in data processing, algorithms, and uncertainty management.

Chapter 3

Methodology

3.1 Introduction

This chapter is set out to discuss the approaches for energy consumption forecasting with different machine learning models.It will be shaped in a way that aligns with the stated goals of creating accurate forecast models of energy consumption while taking into account the complexity of the patterns of energy usage. Modern machine learning models come with this strategy built in, which gives it strength and flexibility in completing the forecasting task.

Following an experimental approach, the research design is represented in Figure 3.1, and it is based on the empirical investigation of the machine learning models. It involves starting with data collection and then going straight to elaborate data preprocessing, which includes cleaning steps, feature engineering, and exploratory data analysis probably being the most basic and essential steps in ensuring data quality and enhancing model performances.

Development for the decision trees, random forests, and neural networks-MLP, CNN, and LSTM architecture-is performed along with fine-tuning in the model development phase. Also, hyperparameter tuning was done in an effort to try and improve the accuracy, using methods like GridSearchCV among other optimization methods. Ensemble learning and hybrid models have been developed, further enhancing the forecasting performance through the exploitation of strengths from different algorithms.

The methodology is designed in such a way that the process for energy consumption forecasting will be comprehensively taken into consideration systematically and in detail, where each stage starting from data collection down to model evaluation meaningfully contributes to the development of reliable and valid forecasting models.

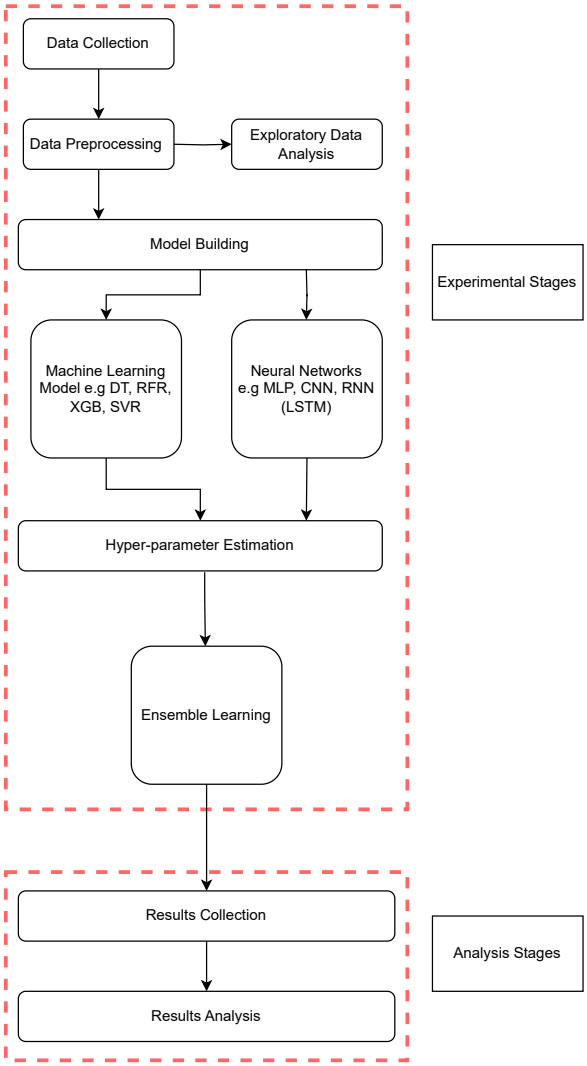


Figure 3.1: Experimental Methodology Framework.

3.2 Data Collection

For this research, the primary data source utilized was the PJM Interconnection (PJM) database, which provides extensive hourly energy consumption data across various service territories within the PJM RTO region. PJM, a regional transmission organization, ensures reliable electricity distribution and manages the high-voltage grid across several states in the United States, including North Carolina, Ohio, and Virginia, which

are serviced by Dominion Energy (DOM). The data provided by PJM includes both company-verified metered values and PJM-generated estimates, ensuring the availability of comprehensive and high-quality datasets. Figure 3.2 shows a map of Pjm Zones with Dominion highlighted in Purple

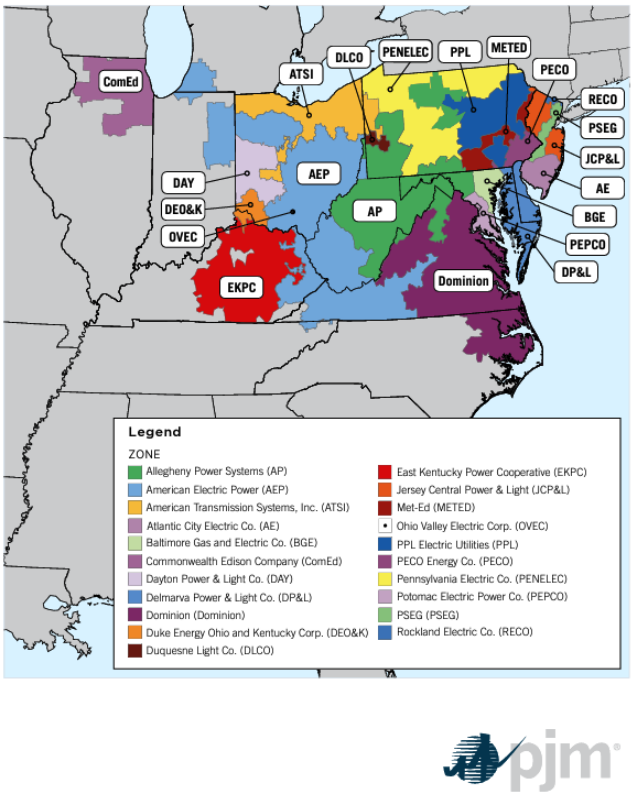


Figure 3.2: Pjm Zones.

The data spans from January 1, 2006, to July 31, 2024, and was collected manually by filtering the PJM database for the Dominion Energy (DOM) region. This hourly load data represents the net energy consumed in megawatt-hours (MW), covering over 17 years of historical data. Each year's dataset was downloaded individually from the PJM database and subsequently uploaded to a GitHub repository for remote access and streamlined processing during model development.

The data gathered includes various timestamps, energy consumption readings, and other relevant features required for accurate energy consumption forecasting. The collection process ensures that the datasets are consistent and reliable, forming a solid foundation for subsequent machine learning model training and evaluation.

The list of file paths for each year's data was created in a python list and pandas was used to read each file. Figure 3.3 shows the table loaded with python pandas for

further preprocessing and analysis

1 to 25 of 400 entries Filter

index	datetime_beginning_utc	datetime_beginning_ept	nerc_region	mkt_region	zone	load_area	mw	is_verified
0	1/1/2008 5:00:00 AM	1/1/2008 12:00:00 AM	SERC	SOUTH	DOM	DOM	9948.0	true
1	1/1/2008 6:00:00 AM	1/1/2008 1:00:00 AM	SERC	SOUTH	DOM	DOM	9744.0	true
2	1/1/2008 7:00:00 AM	1/1/2008 2:00:00 AM	SERC	SOUTH	DOM	DOM	9508.0	true
3	1/1/2008 8:00:00 AM	1/1/2008 3:00:00 AM	SERC	SOUTH	DOM	DOM	9392.0	true
4	1/1/2008 9:00:00 AM	1/1/2008 4:00:00 AM	SERC	SOUTH	DOM	DOM	9417.0	true
5	1/1/2008 10:00:00 AM	1/1/2008 5:00:00 AM	SERC	SOUTH	DOM	DOM	9611.0	true
6	1/1/2008 11:00:00 AM	1/1/2008 6:00:00 AM	SERC	SOUTH	DOM	DOM	9912.0	true
7	1/1/2008 12:00:00 PM	1/1/2008 7:00:00 AM	SERC	SOUTH	DOM	DOM	9999.0	true
8	1/1/2008 1:00:00 PM	1/1/2008 8:00:00 AM	SERC	SOUTH	DOM	DOM	10043.0	true
9	1/1/2008 2:00:00 PM	1/1/2008 9:00:00 AM	SERC	SOUTH	DOM	DOM	10111.0	true
10	1/1/2008 3:00:00 PM	1/1/2008 10:00:00 AM	SERC	SOUTH	DOM	DOM	10131.0	true
11	1/1/2008 4:00:00 PM	1/1/2008 11:00:00 AM	SERC	SOUTH	DOM	DOM	10094.0	true
12	1/1/2008 5:00:00 PM	1/1/2008 12:00:00 PM	SERC	SOUTH	DOM	DOM	9920.0	true
13	1/1/2008 6:00:00 PM	1/1/2008 1:00:00 PM	SERC	SOUTH	DOM	DOM	9766.0	true
14	1/1/2008 7:00:00 PM	1/1/2008 2:00:00 PM	SERC	SOUTH	DOM	DOM	9673.0	true
15	1/1/2008 8:00:00 PM	1/1/2008 3:00:00 PM	SERC	SOUTH	DOM	DOM	9729.0	true
16	1/1/2008 9:00:00 PM	1/1/2008 4:00:00 PM	SERC	SOUTH	DOM	DOM	10241.0	true
17	1/1/2008 10:00:00 PM	1/1/2008 5:00:00 PM	SERC	SOUTH	DOM	DOM	11704.0	true
18	1/1/2008 11:00:00 PM	1/1/2008 6:00:00 PM	SERC	SOUTH	DOM	DOM	12154.0	true
19	1/2/2008 12:00:00 AM	1/1/2008 7:00:00 PM	SERC	SOUTH	DOM	DOM	12210.0	true
20	1/2/2008 1:00:00 AM	1/1/2008 8:00:00 PM	SERC	SOUTH	DOM	DOM	12117.0	true
21	1/2/2008 2:00:00 AM	1/1/2008 9:00:00 PM	SERC	SOUTH	DOM	DOM	11778.0	true
22	1/2/2008 3:00:00 AM	1/1/2008 10:00:00 PM	SERC	SOUTH	DOM	DOM	11116.0	true
23	1/2/2008 4:00:00 AM	1/1/2008 11:00:00 PM	SERC	SOUTH	DOM	DOM	10579.0	true
24	1/2/2008 5:00:00 AM	1/2/2008 12:00:00 AM	SERC	SOUTH	DOM	DOM	10299.0	true

Figure 3.3: PJM Dominion Energy(DOM) Data table.

3.3 Data Preprocessing

The preprocessing of data is an important prerequisite, which gets the raw data ready to feed the machine learning models with structured and cleaned data for model training. This covers the transformation of data into time-series format, handling missing values, and feature engineering. Each of these preprocessing tasks contributes to the accuracy and speed of machine learning models in forecasting energy consumption.

3.3.1 Transforming Data to a Time-Series Format

Since this research focuses on energy consumption forecasting, the dataset must be converted into a time-series format to capture temporal dependencies. The data collected from PJM is initially in a raw format with multiple columns and timestamps. The following steps were implemented to transform the data into time-series format:

1. **Loading Data:** The hourly energy consumptions are downloaded for the years ranging from 2006 to 2024. Each year's data is loaded into a separate file.

2. **Removing Unnecessary Columns:** All columns, which aren't needed for any meaningful modeling tasks, have been removed, for example, datetime_beginning_utc, nerc_region, zone, load_area, is_verified,mkt_region.

3. **Datetime Conversion:** The timestamp datetime_beginning_ept was converted to a proper datetime format, ensuring that the time-series structure is preserved.

4. **Merging Data:** Data from different years was concatenated into a single dataframe to make continuous data for time-series analysis.

```
def process_file(file_path):
    df = pd.read_csv(file_path)
    df = df.drop(columns=['datetime_beginning_utc', '
        nerc_region', 'zone', 'load_area', 'is_verified', '
        mkt_region'])
    df['datetime_beginning_ept'] = pd.to_datetime(df['
        datetime_beginning_ept'], format='%m/%d/%Y %I:%M:%S %p
        ')
    return df

# Processing and concatenating the data
dataframes = [process_file(fp) for fp in file_paths]
combined_df = pd.concat(dataframes, ignore_index=True)
combined_df = combined_df.rename(columns={'
    datetime_beginning_ept': 'datetime', 'mw': '
    energy_consumption_mw'})
```

Figure 3.4: Python Code for Processing and Concatenating Data.

This ensures that the data, when prepared in this manner, is representative of a continuous dataset for which time-series analysis can be performed. It enables the models to learn the different trends, seasonality, and temporal patterns that might exist in the energy consumption data. The preprocessed dataset is then ready for the next steps of handling missing values, scaling, and feature engineering.

3.3.2 Handling Missing Values and Outliers

Handling Missing Values

The next task was to check for missing values during the data exploration phase. It was determined that there are no missing values in the dataset, which simplifies the preprocessing workflow. Both the datetime and energy_consumption_mw columns were found to be complete therefore no imputation or handling techniques were necessary for this step.

Identifying and Addressing Outliers

While there are no missing values, the analysis revealed a significant change in energy consumption patterns starting from 2020. When plotting the time series data, Figure 3.5 shows that energy consumption increased substantially between 2020 and 2024.

This may be attributed to several factors such as changes in economic activity, population growth, increased use of electrical devices due to remote work (post-COVID-19 pandemic effects), or regional infrastructure developments.

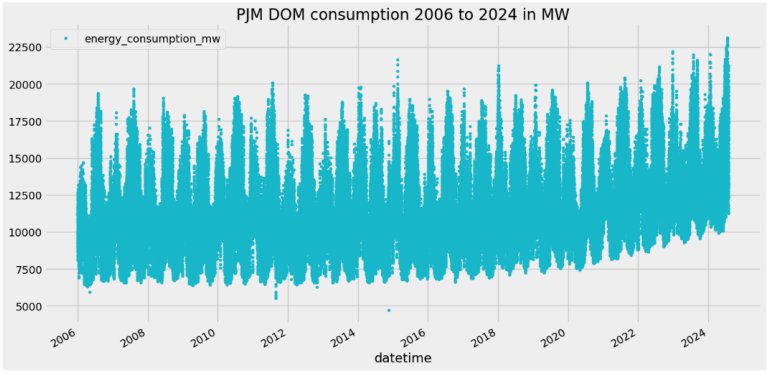


Figure 3.5: PJM DOM consumption 2006 to 2024 in MW.

Given this observed shift, the dataset was split into two periods:

1. Training Period (2006–2019): This period reflects the stable energy consumption pattern before the major change in 2020, and will be used to train the machine learning models.

2. Case Study Period (2020–2024): This period, where the consumption patterns show a marked increase, will be reserved for case study analysis to examine how the models perform on this more recent, distinct dataset.

By separating the data into these two periods, the analysis can focus on evaluating model performance on historical data while also testing its adaptability to the recent surge in energy consumption.

Outliers were not explicitly removed from the dataset since the spike in consumption post-2020 is likely representative of real-world changes and not data anomalies. This strategic decision allows the model to capture these consumption trends without losing valuable information.

3.3.3 Feature Engineering

Feature engineering involves creating new variables from existing data to help models capture more intricate patterns within the dataset. For this study, I performed three key feature engineering tasks: creating datetime features, generating holiday features, and obtaining weather-related features.

Creating Datetime Features

Datetime features are important for time-series forecasting as they allow the model to capture periodic patterns such as daily, weekly, and yearly time intervals. Using the datetime index, several features that represent different aspects of the time component, such as the hour, day, month, and year.The following features were extracted below

- **Hour:** Represents the hour of the day, capturing daily patterns in energy consumption.
- **Day of the week:** Captures the variations in energy usage based on weekdays versus weekends.
- **Quarter:** Represents the business quarter, helping to capture seasonal effects.
- **Month and year:** Important for identifying long-term trends.
- **Day of the year and day of the month:** Captures finer details in daily and yearly cycles.
- **Week of the year:** Helps the model capture weekly consumption patterns.

Get Holiday Features

Public holidays can have a large impact on energy consumption due to disruption in the usual patterns of work and home life. To account for these change, holiday features were created using the holidays library fixed on major U.S. holidays. These holiday labels were merged into the dataset, allowing the model to know the difference between regular days and holidays.As shown in Figure 3.6 the holidays were obtained and added to the data using the following function:

```python
def get_holiday_features(df, country_code='US'):
    year_range = list(range(min(df.index.year), max(df.index.
        year) + 1))
    country_holidays = holidays.country_holidays(
        country_code,
        years=year_range,
        observed=False
    )
    holiday_df = pd.DataFrame(country_holidays.items())
    holiday_df.columns = ['date', 'holiday']
    holiday_df['date'] = pd.to_datetime(holiday_df['date'])
    return holiday_df
```

Figure 3.6: Python Function for Getting Holiday Features.

The holiday feature was merged with the main dataset, and missing values were filled with empty strings to handle non-holiday dates.

Get Weather Features

Weather is a key determinant of energy consumption, especially for our data which is from PJM Dominion Energy which supplies energy to Virginia. To capture the effects of weather on energy usage, historical weather data based on the geographical location of Richmond, Virginia was obtained. The following weather-related features were added:

- **Minimum temperature absolute difference from room temperature:** This captures how far the daily minimum temperature is from a standard comfortable room temperature (20°C), highlighting heating or cooling needs.
- **Maximum temperature absolute difference from room temperature:** Similar to the above, this feature captures the cooling demand during peak temperatures.

These features were extracted using the following function in Figure 3.7:

```python
def get_weather_features(df, lat, lon):
    room_temperature = 20  # Celsius
    start = min(df.index)
    end = max(df.index)
    stations = Stations().nearby(lat, lon).fetch(1)
    weather_data = Daily(stations['wmo'][0], start, end).
        fetch()
    weather_data['tmin_abs_diff_from_room_temperature'] = abs
        (weather_data['tmin'] - room_temperature)
    weather_data['tmax_abs_diff_from_room_temperature'] = abs
        (weather_data['tmax'] - room_temperature)
    weather_data = weather_data.rename(columns={'time': 'date
        '})
    return weather_data
```

Figure 3.7: Python Function for Getting Weather Features.

By incorporating these weather features, the model gains an understanding of how temperature variations influence energy consumption patterns.

Merging Features

Figure 3.8 shows the final step in the feature engineering process which involved merging the datetime, holiday, and weather features into the main dataset

```python
data = create_datetime_features(data_main)
holiday_features = get_holiday_features(data)
weather_features = get_weather_features(data, 37.5407,
    -77.4360) # Richmond Capital of Virginia
data = data.merge(holiday_features, how='left', on='date')
data['holiday'] = data['holiday'].fillna('')
data = pd.get_dummies(data)
data = data.merge(weather_features, how='left', on='date')
```

Figure 3.8: Calling all Python functions and Merging Features.

These engineered features play a critical role in improving the model's predictive power by allowing it to learn from time, holiday effects, and weather fluctuations, all of which have significant influences on energy consumption patterns.

3.3.4 Data Normalization

Min-max normalization is used to scale data within a specific range, typically [0, 1], to ensure that all features e.g `hour`, `day` contribute equally to the model's learning process, especially in models like LSTMs that are sensitive to the magnitude of input

values. By preserving the relative relationships between the original feature values, min-max normalization helps balance features that may have different units or scales. This approach is particularly useful when working with time-series data and neural networks, as it improves the convergence and predictive performance of models. Kim et al. 2024 showed that min-max scaling, combined with LSTM algorithms, yields superior results during evaluations compared to other normalization techniques like z-score and mean normalization. The formula for min-max normalization is:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.1}$$

where x is the original value, x' is the normalized value, $\min(x)$ is the minimum value in the feature, and $\max(x)$ is the maximum value in the feature.

3.4 Model Building

3.4.1 Introduction to Model Building

This sections focuses on transforming the timeseries data into a format that machine learning algorithms can interpret and generalize from. For this study, both supervised learning models and neural networks were used to forecast energy consumption. The choice of models was motivated by their capacity to capture both linear and non-linear dependencies in time series data, with specific attention given to hyperparameter tuning and model validation.

Data Splitting and Preprocessing

A critical step in the model-building process is the splitting of data into training and testing sets. The goal is to ensure that models are trained on a portion of historical data while being evaluated on unseen data to simulate real-world forecasting conditions.

To maintain the temporal integrity of the time series data, the split was made based on a specific date: January 1, 2016. Data before this date was allocated to the training set, while data afterward (January 1, 2016 to December 31, 2019) was reserved for testing. This approach aligns with the principles of time series forecasting, where data is not shuffled to avoid leakage of future information into the training phase. Also note

that Data From January 1, 2020 to July 31, 2024 was reserved for the case study as explained when we were identifying and dealing with outlier earlier in the chapter.

In both supervised learning models and neural networks, the Min-Max Scaling technique was applied to normalize the features. For neural networks, particularly Long Short-Term Memory (LSTM) models, this step is essential, as LSTMs perform optimally when input data is scaled to a consistent range.

The core steps for supervised learning models, like XGBoost and Random Forest, involved the following:

1. Dropping irrelevant columns (date, datetime, energy_consumption_mw) to isolate the feature set.

2. Normalizing the features using the `MinMaxScaler`, which scales each feature between 0 and 1, ensuring uniformity across input features.

3. Splitting the normalized dataset into training and testing sets.

For neural networks, specifically LSTMs, additional preprocessing was required:

1. The same `MinMaxScaler` was applied to both the feature set (X) and the target variable (y) to ensure consistent scaling across inputs and outputs.

2. The input data was reshaped into a 3D structure, which is a requirement for LSTM models, with the format [samples, time steps, features]. This allows the LSTM model to capture temporal dependencies in the data by passing sequences of observations over time.

3.4.2 Linear Regression

Linear Regression has been a cornerstone of forecasting models due to its simplicity and interpretability. In energy consumption forecasting, it is particularly useful for modeling relationships between dependent variables (energy consumption) and one or more independent features such as temperature, time of day, or occupancy rates. The literature review in Chapter 2 emphasizes the importance of capturing linear trends in historical data as a foundation for more complex models. Although linear regression may not perform well in capturing non-linear patterns, it serves as an essential baseline model for understanding and interpreting energy consumption patterns, aligning with studies such as those by Tunç, Çamdali, and Parmaksizoğlu 2006 on electricity consumption prediction.

The Linear Regression model was built using the `LinearRegression` class from the sklearn library. The model was trained on the scaled features from the training dataset, ensuring that all input variables were normalized to fall within a uniform range. The target variable was not scaled, given that linear regression models require interpretable output in the original unit. After training, predictions were generated on the scaled test data. The global `random_seed` of 22 was used for repeatability and fitting the model to the preprocessed data.

Also, note the result collection and result analysis for Linear regression will be discussed further down the chapter.

3.4.3 Decision Tree

The Decision Tree algorithm is another well-established model reviewed in Chapter 2. It provides an interpretable, non-parametric method for predicting energy consumption by creating rule-based splits in the data. In energy forecasting, Decision Trees are often used to capture non-linear relationships between features such as time, temperature, and energy consumption. As noted in Chapter 2, Decision Trees can struggle with overfitting but provide significant value due to their interpretability.

The Decision Tree model was built using the `DecisionTreeRegressor` class, with the `random_state` parameter set to 22 for repeatability. After preprocessing the data and ensuring all features were scaled, the model was trained on the scaled training data. Predictions were then made on the scaled test data. This method helps to capture more complex patterns compared to linear regression.

Also, note the result collection and result analysis for Decision Tree will be discussed further down the chapter.

3.4.4 Random Forest Regressor

The Random Forest Regressor is an ensemble method that forms multiple Decision Trees and averages predictions to increase accuracy and reduce overfitting of singular trees. According to the literature in Chapter 2, Random Forests are widely used in energy consumption forecasting due to their robustness in handling both continuous and categorical variables. This method was selected because it improves upon the accuracy and generalization of the Decision Tree model by aggregating results from multiple trees.

The Random Forest model was constructed using the `RandomForestRegressor`

class. As with other models, the data was preprocessed by scaling the features, and a global random seed (22) was set for consistency. The model was then trained on the scaled training data and evaluated on the scaled test set.

Also, note that result collection and analysis for the Random Forest Regressor will be discussed further in this chapter.

3.4.5 Gradient Boosting Regressor

Gradient Boosting builds models sequentially, where each new model corrects the errors made by the previous one. This model works particularly well when modeling complex, nonlinear relationships in time series and is referenced in Chapter 2 for use in energy consumption forecasting. The literature review discusses how Gradient Boosting improves prediction accuracy by focusing on areas where earlier models performed poorly.

The Gradient Boosting model was implemented using the `GradientBoostingRegressor` class. The model was trained on the scaled features of the training data, with a random seed (22) set to ensure consistency in results. After training, predictions were generated on the scaled test data.

Also, note that result collection and analysis for Gradient Boosting will be discussed further in this chapter.

3.4.6 Extreme Gradient Boosting (XGBoost)

XGBoost is the optimized version of Gradient Boosting. It is well-known for its computational efficiency and the accuracy it provides when handling large datasets. As emphasized in Chapter 2, in energy consumption forecasting, XGBoost's effectiveness includes its ability to handle missing data and model complex relationships. This model was selected for its performance in producing highly accurate predictions, as demonstrated in several forecasting studies.

The XGBoost model was built using the `XGBRegressor` class. As with other models, the training data was scaled, and a global random seed (22) was set for repeatability. The model was then trained on the preprocessed data, and predictions were made on the test set.

Also, note that result collection and analysis for XGBoost will be discussed further in this chapter.

3.4.7 Support Vector Regressor (SVR)

The SVR has been selected for this study because it is better suited to handle nonlinearities in the data, which is a key factor for energy consumption forecasting. As indicated in Chapter 2, linear regression seldom captures the complexity of energy consumption patterns. SVR, with its capacity to map inputs into high-dimensional feature spaces, allows for more flexible modeling of non-linear dependencies. Furthermore in Chapter 2, SVR's strong generalization capabilities are ideal for handling the diverse and fluctuating nature of energy demand, ensuring minimal overfitting.

The SVR model was built using the `SVR()` function from the scikit-learn library. After normalizing the input data using `MinMaxScaler`, the SVR model was trained on the scaled training data (`X_train_scaled`) and the corresponding target values (`y_train`). The default hyperparameters were used, focusing on the radial basis function (RBF) kernel, which is particularly effective for capturing non-linear relationships. The trained model was then used to make predictions on the test set.

3.4.8 Multilayer Perceptrons (MLPs)

It was discussed in Chapter 2 that the reasons for the choice of MLPs include their efficacy in understanding intricate nonlinear relationships in energy consumption data. The usefulness of MLPs in modeling such interactions between multiple factors, such as weather and time, which are crucial to forecast energy consumption. Given the versatility and strong predictive power, two versions of MLP models were built with different batch sizes to assess how model performance changes.

Afterward, two MLP models were implemented. Both had a similar structure, where the architecture consisted of three hidden layers, each with 128, 64, and 32 units, respectively. In addition, Dropout was considered after each for preventing overfitting. The ReLU activation function was used throughout the hidden layers, whereas the final layer was providing the output for regression. Among them, the first was trained with a batch size of 1000, and the batch size used to train the second was 365. These were trained on 150 epochs using the Adam optimizer for testing the models.

Both the MLP models comprise one input layer, three fully connected hidden layers, and one output layer. The Dropout layers were used to avoid overfitting.

3.4.9 Convolutional Neural Networks (CNNs)

The selected CNN architectures capture the spatial and temporal hierarchies in data. This has made the CNN competent in modeling time series data through feature extraction using convolutional layers as explained in the literature review; hence, they are appropriate for energy consumption forecasts. To assess the impact of batch size on model performance, two versions of CNN models were built.

Accordingly, two equal architecture models of CNN were created; each used one convolutional layer with 64 filters, followed by a MaxPooling layer and a fully connected layer. Each was trained, in this case, where the first model was trained at a batch size of 1000 and the second was at a batch size of 365. Then, fitting for the two models on 100 and 150 epochs, respectively, on the test data was carried out.

So, in a nutshell, CNN models are composed by a convolutional layer, followed by a MaxPooling to reduce the dimensions and finally by a fully-connected which returns the final regression result.

3.4.10 Long Short-Term Memory networks (LSTMs)

Long Short-Term Memory Networks were chosen because since these neurons have the potential in modeling time series data, especially in those cases with long-term dependencies. LSTMs have been used since they become very efficient in capturing day-to-day, weekly, and seasonal patterns of energy consumption forecasting. Two variants, LSTM models were created to compare performances running different batch sizes.

This means each model in the series had three LSTM layers consisting of 100 units, then followed by a Dropout layer overfitting. The first was trained with the batch size 1,000, while the second was with a batch size of 365. The first model was trained for 100 epochs, while this one was trained for 150 epochs; in the testing part, this model was evaluated on the test set with metrics such as RMSE, MAPE and MAE.

All LSTM models used here have an identical architecture, using stacked LSTM layers to model long-term dependencies. After each layer, there is a Dropout layer to avoid overfitting and a regression output layer.

3.4.11 Hyperparameter Estimation

In this section, we build and optimize various supervised Machine Learning models, including Linear Regression, Polynomial Regression, Decision Tree, Random Forest,

Gradient Boosting, XGBoost, Multilayer Perceptrons (MLPs), Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks using built-in functions of sklearn, such as GridSearchCV, Keras Tuner, Keras, and TensorFlow. These hyperparameter-adjusted models were employed to predict the arrival time of the train, with a particular focus on minimizing the mean squared error for traditional models and the validation loss for the models with neural components. More specially, the key hyperparameters, such as the number of units, dropout rates, and learning rates, of these architectures are being optimized under computational resources constraints, such as limiting the number of n-folds for the GridSearchCV, the number of trials of the KerasTuner, the number of epochs for the neural networks, etc.

Linear Regression

Two versions of Linear Regression were implemented. In the first implementation, GridSearchCV was used to search for the best combination of two hyperparameters while the second version involved a pipeline combining Polynomial Features with Linear Regression. Table 3.1 showing the hyperparameters that are tuned:

Hyperparameter Tuning Model	**Version**	**Values Tested**
Hyperparameter Tuned LR 1	Version 1	fit_intercept: True, False
		positive: True, False
Hyperparameter Tuned LR 2	Version 2	poly__degree: 1, 2, 3
		poly__interaction_only: True, False
		poly__include_bias: True, False
		linear__fit_intercept: True, False
		linear__positive: True, False

Table 3.1: Hyperparameter tuning for Linear and Polynomial Regression models

Decision Tree

A Decision Tree Regressor was trained using GridSearchCV to optimize the following hyperparameters in Table 3.2

Hyperparameter Tuning Model	Version	Values Tested
Hyperparameter Tuned DTR	Version 1	max_depth: 5, 10, 15, 20
		min_samples_split: 2, 10, 20, 25
		min_samples_leaf: 1, 5, 10, 15
		max_features: None, sqrt, log2

Table 3.2: Hyperparameter tuning for Decision Tree Regressor

Random Forest Regressor

Two versions of the Random Forest Regressor were implemented, both using GridSearchCV, The first and second versions(Extension of the first version) of the hyperparameters are shown in Table 3.3:

Hyperparameter Tuning Model	Version	Values Tested
Hyperparameter Tuned RFR 1	Version 1	n_estimators: 50, 100
		max_depth: 10, 20, None
		min_samples_split: 2, 5
		min_samples_leaf: 1, 2
Hyperparameter Tuned RFR 2	Version 2	n_estimators: 50, 100, 200
		max_depth: 10, 20, None
		min_samples_split: 2, 5, 7
		min_samples_leaf: 1, 2, 3

Table 3.3: Hyperparameter tuning for Random Forest Regressor

Gradient Boosting Regressor

Two implementations of Gradient Boosting Regressor were tuned with GridSearchCV, The first and second versions(Extension of the first version) of the hyperparameters are shown in Table 3.4:

Hyperparameter Tuning Model	Version	Values Tested
Hyperparameter Tuned GBR 1	Version 1	n_estimators: 50, 100, 150
		max_depth: 3, 5
		learning_rate: 0.05, 0.1
		subsample: 0.8
		min_samples_split: 2, 5
		min_samples_leaf: 1, 2
Hyperparameter Tuned GBR 2	Version 2	n_estimators: 50, 100, 150
		max_depth: 3, 5, 7
		learning_rate: 0.05, 0.1
		subsample: 0.8, 1.0
		min_samples_split: 2, 5, 7
		min_samples_leaf: 1, 2, 3

Table 3.4: Hyperparameter tuning for Gradient Boosting Regressor

Extreme Gradient Boosting (XGBoost)

Two XGBoost models were optimized, The first version tuned the following hyperparameters while The second version had a finer grid. Table 3.5 shows the parameters:

Hyperparameter Tuning Model	Version	Values Tested
Hyperparameter Tuned XGB 1	Version 1	n_estimators: 50, 100
		max_depth: 3, 5
		learning_rate: 0.05, 0.1
		subsample: 0.8
		colsample_bytree: 0.8
		gamma: 0, 1
Hyperparameter Tuned XGB 2	Version 2	n_estimators: 100, 200
		max_depth: 3, 5, 7
		learning_rate: 0.01, 0.05, 0.1
		subsample: 0.8, 1.0
		colsample_bytree: 0.8, 1.0
		gamma: 0, 0.5, 1
		reg_lambda: 1, 1.5
		reg_alpha: 0, 0.5

Table 3.5: Hyperparameter tuning for XGBoost

Support Vector Regressor (SVR)

No hyperparameter tuning was performed for Support Vector Regressor due to its high computational demands and poor performance in initial experiments.

Multilayer Perceptrons (MLPs)

For the MLP models, Keras Tuner was used with Bayesian Optimization to search for the best hyperparameters. Two models were built with the following steps:

- Defined the model with two hidden layers using Dense and Dropout layers, allowing the hyperparameter search for the number of units (64, 128), dropout rates (0.2, 0.3), and learning rates (1e-4 to 1e-3).
- Initialized the tuner with BayesianOptimization, limiting the number of trials to 2 for faster tuning.
- Performed the hyperparameter search using reduced epochs (50), fixed batch size (128), and early stopping criteria.
- Retrieved the best hyperparameters and trained the model with the best configuration.

The second model followed a similar process but with slightly different configurations, such as using a batch size of 1000 and training for 150 epochs.Table 3.6 showing the parameters in tabular form.

Hyperparameter Tuning Model	**Version**	**Values Tested**	**Optimization**
Hyperparameter Tuned MLP 1	Version 1	units: 64, 128	Bayesian Optimization
		dropout: 0.2, 0.3	
		learning_rate: 1e-4 to 1e-3	
Hyperparameter Tuned MLP 2	Version 2	batch_size: 1000, epochs: 150	Bayesian Optimization

Table 3.6: Hyperparameter tuning for Multilayer Perceptrons (MLPs) using Keras Tuner

Convolutional Neural Networks (CNNs)

Two CNN models were built and optimized using Keras Tuner with the following steps:

- Defined the CNN model with tunable Conv1D filters (32-128), kernel sizes (2-5), and Dense layers with adjustable units (30-100). The dropout rate was also tuned within the range of 0.0 to 0.5.

- Initialized the tuner with RandomSearch, setting a maximum of 5 trials and one execution per trial for efficient search.
- Added early stopping to monitor validation loss, allowing the model to stop training early if the validation loss did not improve for 3 consecutive epochs.
- Retrieved the best hyperparameters and built the final model with optimal configurations.

In the second CNN model, a similar procedure was followed but with 100 epochs instead of 50, to explore deeper architectures and more complex patterns. Table 3.7 showing the parameters in tabular form.

Hyperparameter Tuning Model	**Version**	**Values Tested**	**Optimization**
Hyperparameter Tuned CNN 1	Version 1	Conv1D filters: 32-128	Random Search
		kernel_sizes: 2-5	
		Dense units: 30-100	
		dropout: 0.0-0.5	
Hyperparameter Tuned CNN 2	Version 2	epochs: 100	Random Search

Table 3.7: Hyperparameter tuning for Convolutional Neural Networks (CNNs) using Keras Tuner

Long Short-Term Memory networks (LSTMs)

For LSTMs, Random Search with Keras Tuner was used to optimize three LSTM layers and the following steps were taken:

- Defined the LSTM model with three layers, where the units for each LSTM layer were tuned between 50 and 150, along with dropout rates (0.1 to 0.4).
- Initialized the tuner, limiting trials to 2 for faster tuning.
- Performed the hyperparameter search using reduced epochs (50), with a fixed batch size of 128.
- Retrieved the best hyperparameters and trained the final LSTM model using the optimal configuration.

The second and third LSTM models were built with similar steps but extended the training to 100 epochs and tuned the batch size as well. Table 3.8 showing the parameters in tabular form.

Hyperparameter Tuning Model	Version	Values Tested	Optimization
Hyperparameter Tuned LSTM 1	Version 1	LSTM units: 50-150	Random Search
		dropout: 0.1-0.4	
		batch_size: 128, epochs: 50	
Hyperparameter Tuned LSTM 2	Version 2	epochs: 100, batch_size: tuned	Random Search

Table 3.8: Hyperparameter tuning for Long Short-Term Memory Networks (LSTMs) using Keras Tuner

3.4.12 Ensemble Learning

The choice of ensemble learning techniques, particularly XGBoost, Random Forest Regressor, and Gradient Boosting, is well-supported by the literature on energy consumption forecasting. As discussed in Chapter 2, ensemble methods are very good at handling complex, nonlinear relationships in data while reducing prediction variance. These models have been successfully applied in past research for energy forecasting because they combine the strengths of multiple algorithms, providing a balance between bias and variance. The superior performance of ensemble models, especially in high-dimensional data contexts such as energy forecasting, justifies their inclusion in this study. Incorporating ensemble models such as Voting Regressors is a natural progression from standalone machine learning techniques. These ensemble methods address the key challenge of balancing predictive accuracy and model interpretability, making them ideal for forecasting tasks where large volumes of data and multiple influencing factors, such as weather conditions, must be considered.

Ensemble Learning Implementation

In this methodology, two ensemble learning models were constructed using pre-trained XGBoost, Gradient Boosting Regressor, and Random Forest Regressor models, emphasizing the strength of blending diverse algorithms. In Ensemble 1, the Voting Regressor combines XGBoost and Gradient Boosting. The code achieves this by loading the optimized models from prior GridSearchCV runs and creating a combined model using a VotingRegressor. The ensemble model is then trained on scaled training data, and its performance is evaluated on test data. This process leverages the strengths of each individual model to improve prediction accuracy by averaging their outputs.

Ensemble 2 extends the first ensemble by adding a Random Forest Regressor to the Voting Regressor. This addition provides more diversity in the learning algorithms, further boosting the generalization ability of the ensemble. The models are combined through a voting mechanism that averages the predictions, thereby reducing overfitting and improving accuracy on unseen data. After training the ensemble, predictions were

generated, and model performance was evaluated on the test data. Figure 3.9 showing both Voting regressors that was implemented.

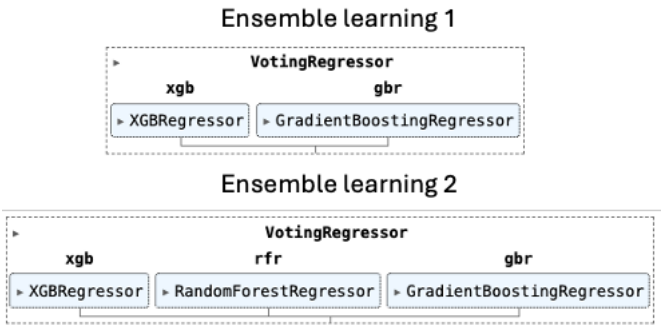


Figure 3.9: Ensemble learning.

3.5 Result Collection

In this section, the results obtained from the machine learning models trained for energy consumption forecasting are collected and summarized. Each model was saved upon completion of the GridSearchCV or manual optimization process to ensure reproducibility. Each of the models was loaded and fitted to the test dataset, starting from 1st January 2016 to 31st December 2019. The forecasted results of each model were exported for further analysis with their evaluation metrics. Table 3.5 below summarizes the performance comparison of each model in terms of MAPE, MAE, and RMSE.

Model	**Parameter Setting**	**MAPE (%)**	**MAE**	**RMSE**	**Rank**
Random Forest Regressor	None	x.x	x.x	x.x	x
XGBoost	Tuned settings	x.x	x.x	x.x	x
LSTM	Batch Size: 1000	x.x	x.x	x.x	x

Table 3.9: Result collection format

Table 3.9 above provides a summary of 3 machine learning models applied to energy consumption forecasting. The table is organised with five columns: model type, parameter setting and the key evaluation metrics. The Model column specify the type of machine learning algorithm used, and Parameter setting column specifying configuration or tuning the model. The MAPE (%) column indicates the Mean Absolute Percentage Error of the prediction, which is an average percentage difference between prediction and actual values. The MAE column shows the Mean Absolute Error, which

is the measure of the average magnitude of the prediction errors; and RMSE in the last column indicating the Root Mean Squared Error. In general terms, lower values in each column would imply a better model in the prediction range.

3.6 Result Analysis

To evaluate the performance of the models, three key metrics were employed: MAPE, RMSE, and MAE, as these are widely used in energy consumption forecasting. A study of energy prediction models indicates that 29% of reviewed studies utilize MAPE, while 16% use RMSE (Amasyali and El-Gohary 2018). MAPE provides a percentage-based measure of forecast accuracy, making it interpretable across datasets, but it can be sensitive to low values. RMSE gives more weight to larger errors, making it useful for understanding the variance in model predictions. MAE, on the other hand, provides an average of absolute errors, offering a simpler and more intuitive evaluation of model accuracy.

The formulas for these metrics are as follows:

$$\text{MAPE} = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right| \tag{3.2}$$

where Y_t is the actual value, $\hat{Y}_t$ is the predicted value, and n is the number of observations.

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^{n} \left| Y_t - \hat{Y}_t \right| \tag{3.3}$$

where Y_t is the actual value, $\hat{Y}_t$ is the predicted value, and n is the number of observations.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^{n} \left(Y_t - \hat{Y}_t \right)^2} \tag{3.4}$$

where Y_t is the actual value, $\hat{Y}_t$ is the predicted value, and n is the number of observations.

These metrics in the equations above will be used to evaluate the model performance from January 1, 2016, to December 31, 2019, and further analysis will focus on comparing the actual and predicted results over the entire period to gain deeper insights into the model's forecasting abilities.

3.7 Summary

The methodology chapter highlights the planned approach for energy consumption forecasting, underlining the emphasis on machine learning models. Initiating from data gathering of an hourly consumption dataset between the years 2006-2024 from the PJM Interconnection database, exhaustive preprocessing was given to ensuring quality, converting it into time-series, handling the outliers, and feature engineering tasks such as adding weather and holiday features that will eventually enhance predictive accuracy. These steps lay the perfect foundation for model development and evaluation.

The models that will be used throughout this study are supervised learning algorithms: Linear Regression, Decision Trees, Random Forest; neural networks: MLP, CNN, LSTM. The models have been tuned to hyperparameters with hyperparameter optimization techniques such as GridSearchCV and Keras Tuner for increased accuracy in forecasting. Ensemble learning techniques, at the same time, also help improve the performance of the combined model further. In the end, each model will be evaluated using key performance indicators such as MAPE, RMSE, and MAE to come up with a reliable and valid forecast. The methodology represents a balance in technical complexity and interpretability to meet the forecasting objectives of this study.

Chapter 4

Result and Discussion

4.1 Introduction

The experiments covered in Chapter 3 are implemented in this chapter. Figure 3.1 provides an illustration of the main sections of this chapter. Keeping the process in mind, each of those stages is covered in detail in the sections below.

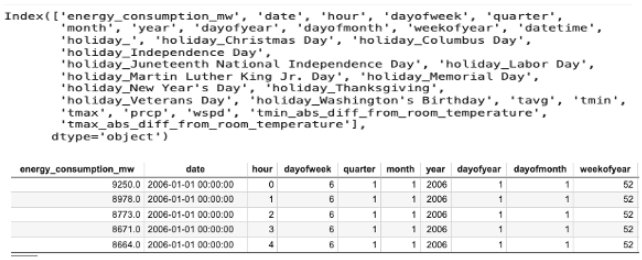
4.2 Data Preprocessing

As outlined in the methodology, the preprocessing phase began by structuring the dataset for forecasting of energy consumption. The data was first transformed into a time-series format, preserving the temporal dependencies essential for energy usage analysis. This transformation involved organizing the data chronologically while extracting key time-based features, such as hour, day, month, and year. These features helped the models capture variations in energy consumption across different time intervals. Furthermore, the dataset is from 2006 to 2024, which allowed the identification of long-term trends and seasonal patterns crucial for accurate predictions.

Addressing missing values and outliers was relatively straightforward. Since no missing values were present, imputation techniques were not required. However, an analysis of outliers revealed a significant shift in energy consumption patterns beginning in 2020, likely influenced by the post-pandemic recovery and increased remote work. Rather than treating this shift as an anomaly, it was incorporated into the models as a real-world phenomenon. The data was then split into two distinct periods which are pre-2020 for training and post-2020 as a case study which enabled a more detailed evaluation of how well the models could handle both stable and dynamic consumption

trends.

In terms of feature engineering, additional information, such as holiday indicators and weather-related features, was introduced to improve predictive performance. Features like holiday labels accounted for irregular energy usage during public holidays, while weather data, such as temperature deviations from room temperature, captured environmental influences on consumption. The final dataset, with these engineered features and the application of min-max normalization for consistent scaling, formed a robust foundation for training the machine learning models aimed at forecasting energy consumption accurately. Figure 4.1 showing the results from the data preprocessing.



```
Index(['energy_consumption_mw', 'date', 'hour', 'dayofweek', 'quarter',
       'month', 'year', 'dayofyear', 'dayofmonth', 'weekofyear', 'datetime',
       'holiday_', 'holiday_Christmas Day', 'holiday_Columbus Day',
       'holiday_Independence Day',
       'holiday_Juneteenth National Independence Day', 'holiday_Labor Day',
       'holiday_Martin Luther King Jr. Day', 'holiday_Memorial Day',
       'holiday_New Year's Day', 'holiday_Thanksgiving',
       'holiday_Veterans Day', 'holiday_Washington's Birthday', 'tavg', 'tmin',
       'tmax', 'prcp', 'wspd', 'tmin_abs_diff_from_room_temperature',
       'tmax_abs_diff_from_room_temperature'],
      dtype='object')
```

energy_consumption_mw	date	hour	dayofweek	quarter	month	year	dayofyear	dayofmonth	weekofyear
9250.0	2006-01-01 00:00:00	0	6	1	1	2006	1	1	52
8978.0	2006-01-01 00:00:00	1	6	1	1	2006	1	1	52
8773.0	2006-01-01 00:00:00	2	6	1	1	2006	1	1	52
8671.0	2006-01-01 00:00:00	3	6	1	1	2006	1	1	52
8664.0	2006-01-01 00:00:00	4	6	1	1	2006	1	1	52

Figure 4.1: Data Preprocessing Result.

4.3 Exploratory Data Analysis (EDA)

4.3.1 Summary Statistics

The dataset contains 122,713 entries with 30 columns. The summary statistics for energy consumption, average temperature (tavg), and the absolute difference between maximum temperature and room temperature (tmax_abs_diff_from_room_temperature) reveal critical trends within the dataset. Energy consumption averages approximately 11,027 MW, with a standard deviation of 2,411 MW, reflecting moderate fluctuations in usage. The minimum observed energy consumption is 4,724 MW, while the maximum recorded value reaches 20,728 MW.

As regards to temperature, the average (tavg) stands at 15.47 °C, with a notable range spanning from -11.7°C to 36.7°C, signifying considerable temperature variability over the dataset's timeline. Additionally, the absolute difference between maximum temperature and room temperature averages 8.37°C, with a standard deviation of 4.95°C. This indicates that deviations from room temperature are common, likely leading to increased energy usage for both heating and cooling purposes.

4.3.2 Correlation Analysis Results

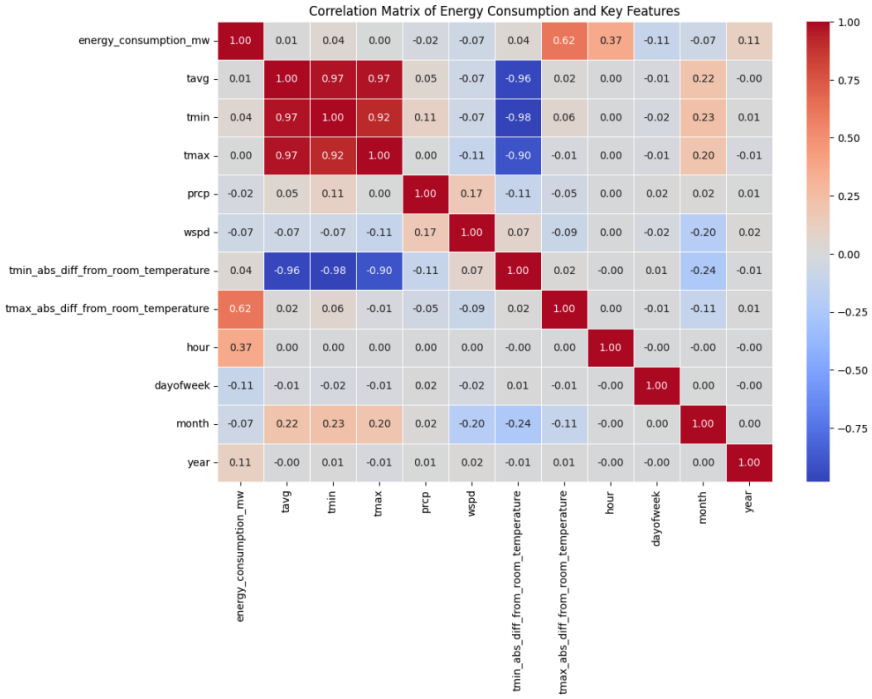


Figure 4.2: Correlation Analysis.

Following the correlation analysis in Figure 4.2, key relationships between energy consumption and the dataset's features were identified. The strongest positive correlation is between energy consumption and the absolute difference between maximum temperature and room temperature (tmax_abs_diff_from_room_temperature), with a correlation coefficient of 0.62. This suggests that when outdoor temperatures deviate further from a comfortable room temperature, energy consumption increases, likely driven by the need for additional heating or cooling.

The hour variable also has a moderate positive correlation of 0.37, showing the variation that occurs during the day, while higher consumption may fall during morning and evening peaks. In contrast, the year feature is weakly positively correlated at 0.11.

Other feature variables like minimum temperature (tmin), average temperature (tavg), and maximum temperature (tmax) are very weakly correlated, having values close to zero, hence causing minimal direct impact on energy consumption. Precipitation, wind speed, month of the year, and day of the week are some of those features which show negative correlations; these correlations are rather weak. Of these, the strongest is the day of the week, with a negative correlation of -0.11.

This analysis highlights that temperature deviations and time of day play the most significant roles in driving energy consumption patterns.

4.3.3 Datetime features Analysis Results

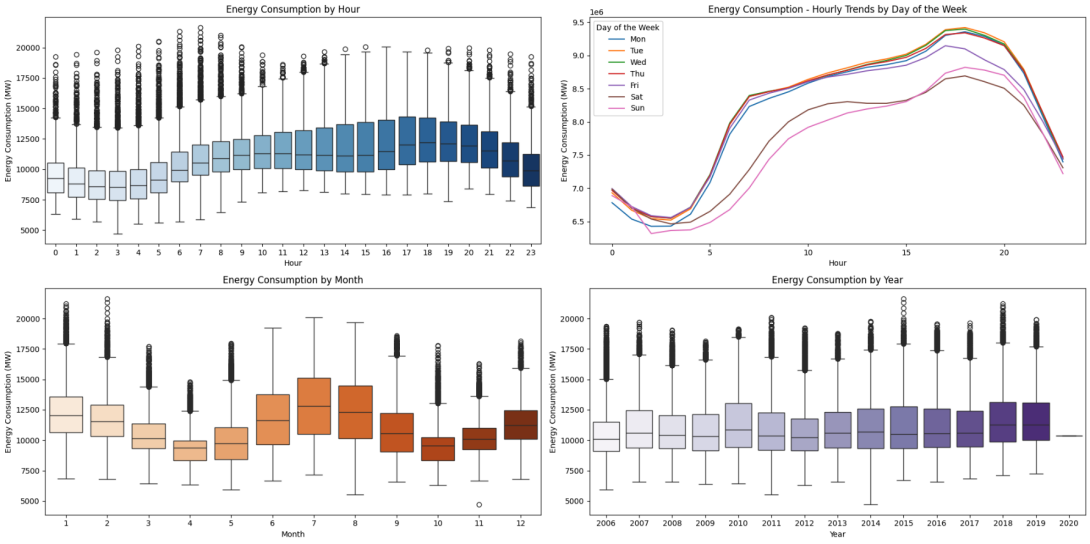


Figure 4.3: Plots of Energy consumption vs Hour, Hour by Day of the week, Month and Year.

By examining Figure 4.3 the four plots uncover key insights into energy consumption patterns. The first plot, a box plot by hour, revealed distinct peaks during specific hours of the day, likely linked to increased residential or commercial activity, such as in the morning and evening, while late-night hours showed lower energy consumption. Next, a line plot comparing hour and day of the week highlighted a consistent trend of higher energy usage on weekdays compared to weekends, suggesting the influence of industrial and commercial activity on energy demand.

Further exploration using a box plot by month exposed a clear seasonal pattern, with energy consumption spiking during the summer and winter months, likely driven by the increased need for cooling and heating, respectively. Finally, the box plot by year indicated relatively stable energy consumption over the years, with minor fluctuations possibly reflecting changes in population, infrastructure, or the implementation of energy efficiency measures.

These visualizations collectively shed light on how daily, weekly, and seasonal cycles impact energy consumption trends.

4.3.4 Energy Consumption vs Temperature Analysis Results

The scatter plot analysing energy consumption against average temperature (`tavg`) in Figure 4.4 uncovers a distinct pattern between temperature and energy usage. At lower temperatures, particularly around 0°C and below, energy consumption is notably higher, likely due to increased heating demands during colder periods. As temperatures rise towards a more moderate range, roughly between 15°C and 20°C, energy usage decreases, reflecting a reduced need for both heating and cooling systems.

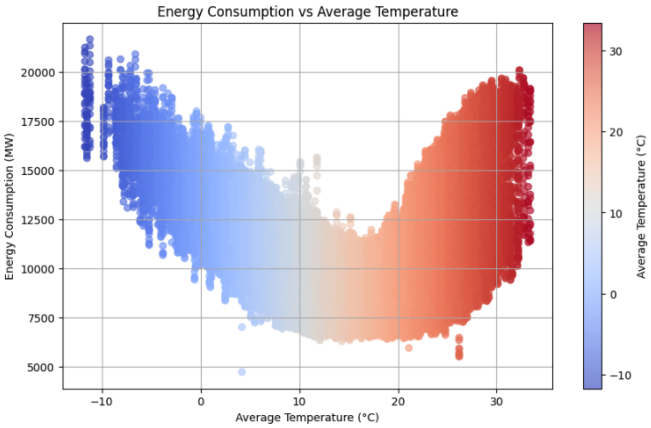


Figure 4.4: Energy Consumption vs Average Temperature

However, as temperatures exceed 25°C, energy consumption begins to rise again, likely driven by increased air conditioning usage during hotter periods. This creates a U-shaped pattern, indicating that extreme temperatures, both hot and cold, heavily influence energy demand as people adjust their environment for comfort. Additionally, the smooth transition from cool blue to warm red in the color map visually highlights the relationship between temperature extremes and peaks in energy consumption.

This plot emphasizes the strong connection between energy usage and temperature fluctuations, underscoring how extreme weather conditions drive changes in energy demand.

4.4 Individual Model Results

4.4.1 Linear Regression Results

The linear regression model was evaluated using key metrics such as MAPE, MAE, RMSE, and R^2. The model's performance between the standard and hyperparameter-tuned versions showed minimal differences. With a MAPE of 11%, the predictions were, on average, off by 11%, and an MAE of 1273 MW indicated the average error in forecasts. The RMSE was approximately 1627 MW , and an R^2 score of 0.557 revealed that the model explained 55.7% of the variation in energy consumption, showing room for improvement in capturing more complex patterns.The evaluation results from the linear regression experiments are shown in Appendix D.

Linear regression effectively captured trends based on features like time of day, day of the week, and other variables, but struggled with more complex, non-linear relationships, such as temperature fluctuations. Feature importance in Figure 4.7 revealed that holidays had a notable impact on energy consumption, but the model couldn't handle interactions between temperature changes and consumption over time. In Figure 4.5 and Figure 4.6 respectively showing the time-series and scatter plots, the model followed general trends but smoothed out sharp spikes in consumption, particularly during peak periods like summer and winter. The scatter plot showed a clear linear trend but revealed increasing variance at higher consumption levels, further highlighting the model's limitations in handling non-linear patterns. While the linear regression model successfully captured overall trends, its inability to manage non-linear relationships and complex dependencies suggests that more advanced models, such as decision trees or neural networks, would provide improved forecasting accuracy.

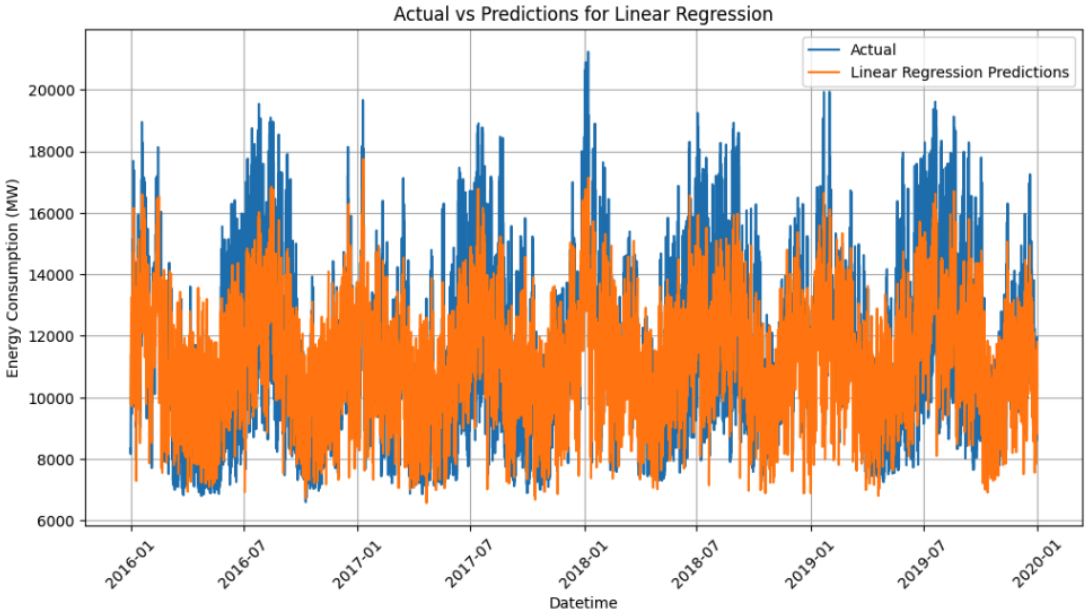


Figure 4.5: Linear regression plot: Actual vs Prediction from January 2016 to December 2019

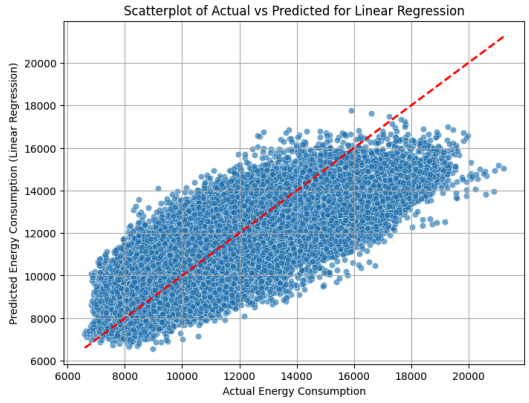


Figure 4.6: Linear regression plot: Scatter-plot Actual vs Predicted

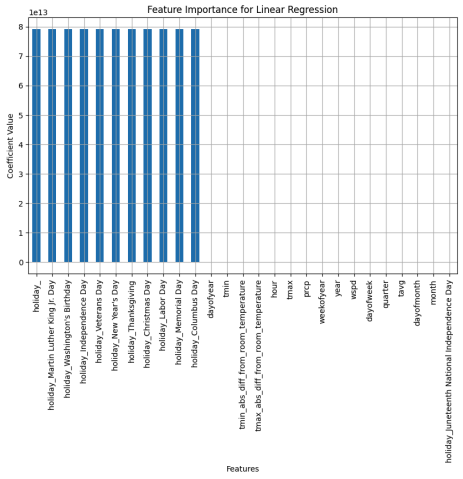


Figure 4.7: Linear regression plot: Feature Importance

4.4.2 Decision Tree Results

The Decision Tree Regressor model was evaluated using MAPE, MAE, RMSE, and R^2, showing a clear improvement after hyperparameter tuning of which you can find the hyperparameters in Table 3.2. The MAPE dropped from 6.18% to 5.79%, while the MAE decreased from 712 MW to 668 MW, indicating smaller forecast errors. The RMSE, which highlights larger prediction errors, decreased from 936.51 MW to 876.60 MW, showing the model's improved accuracy in handling deviations. The R^2 score increased from 0.853 to 0.872, meaning the tuned model explained 87.2% of energy consumption variations, a significant improvement over the standard model. The evaluation results from the decision tree experiments are shown in Appendix D

Decision trees are more effective than linear regression in capturing both linear and non-linear relationships within the data. As seen in the feature importance plot in Figure 4.10, key features such as hour, temperature, and tmax_abs_diff_from_room_temperature had a significant impact on energy consumption. Hyperparameter tuning further refined the model's ability to fit the data, resulting in more accurate predictions, especially during peak periods.

In Figure 4.8 and Figure 4.9, the time-series and scatter plots show that the tuned model closely followed actual energy consumption trends but occasionally overestimated during extreme peaks, particularly in summer and winter. Despite these minor limitations, the Decision Tree Regressor outperformed linear regression by capturing non-linear patterns and interactions.

Overall, the hyperparameter-tuned decision tree model delivered strong forecasting performance, successfully capturing trends and complex relationships. However, further refinement or advanced ensemble models may improve results, particularly for peak consumption periods.

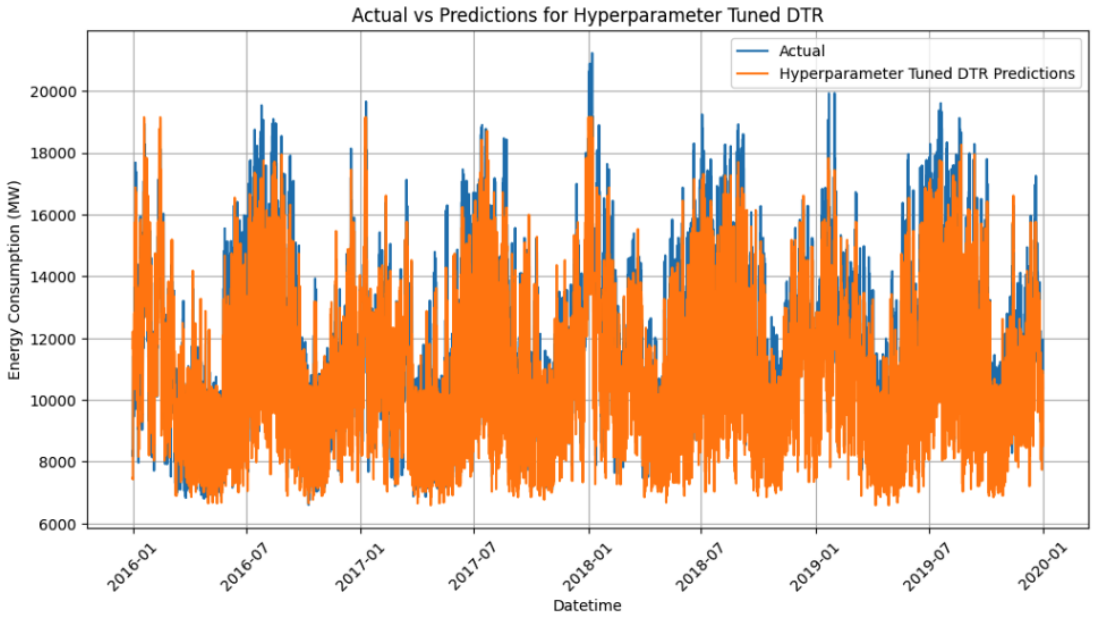


Figure 4.8: Decision Tree regression plot: Actual vs Prediction from January 2016 to December 2019

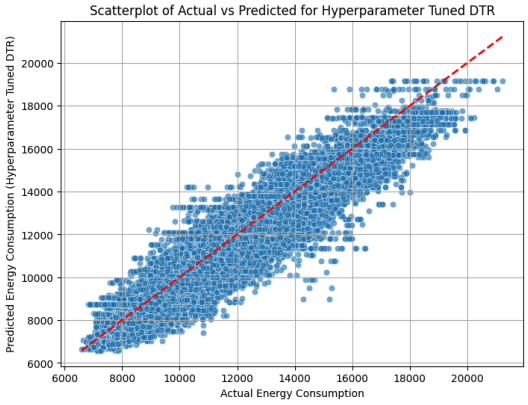


Figure 4.9: Decision Tree regression plot: Scatterplot Actual vs Predicted

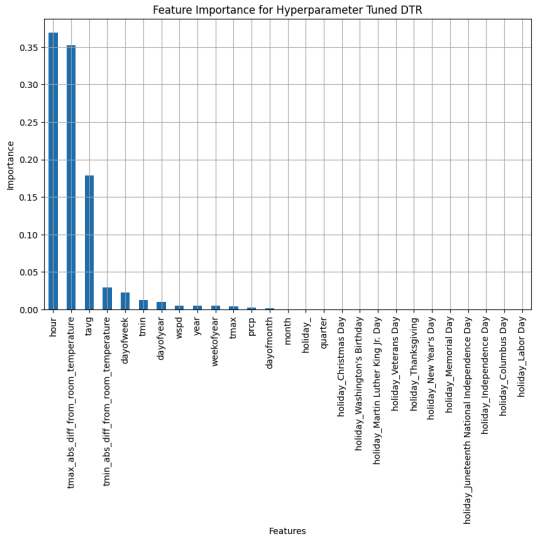


Figure 4.10: Decision Tree regression plot: Feature Importance

4.4.3 Random Forest Results

The Random Forest Regressor model was evaluated using MAPE, MAE, RMSE, and R^2. Both the standard model and the first hyperparameter-tuned versions in Table 3.3 performed similarly, with a MAPE of 4.92%, an MAE of 569 MW, and an RMSE of 735 MW. The R^2 score of 0.91 indicated that the model explained 90.9% of the variation in energy consumption. The

evaluation results from the random forest experiments are shown in Appendix D

The second hyperparameter-tuned version showed slight improvements, reducing the MAPE to 4.90%, the MAE to 567 MW, and the RMSE to 733 MW. The R^2 remained at 0.91, reflecting a minor enhancement in capturing energy consumption trends.

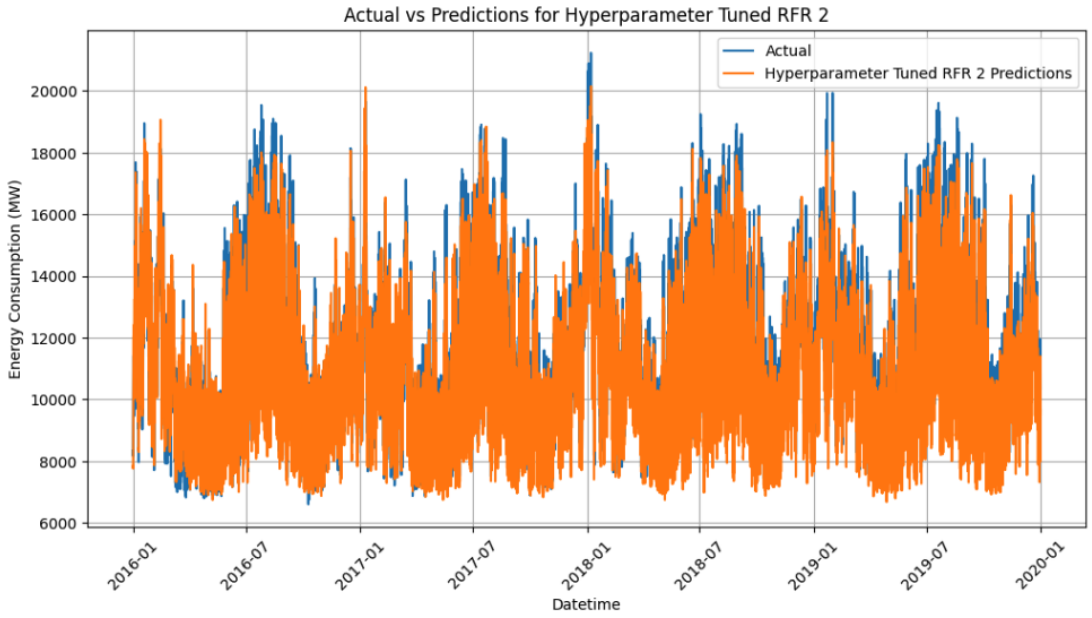


Figure 4.11: Random Forest regression plot: Actual vs Prediction from January 2016 to December 2019

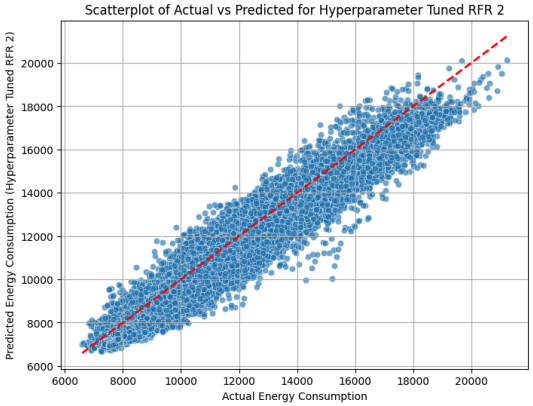


Figure 4.12: Random Forest regression plot: Scatter-plot Actual vs Predicted

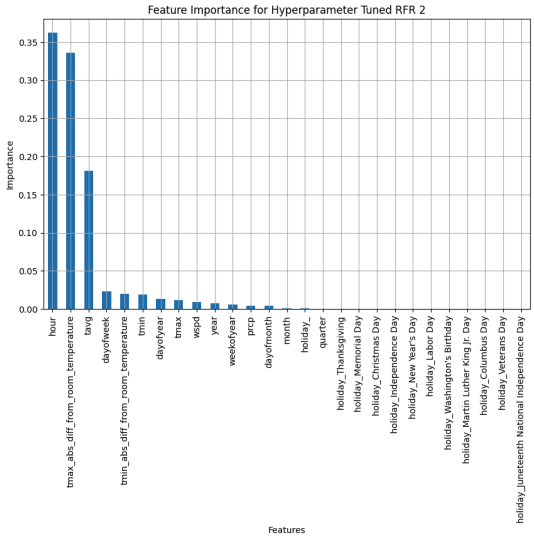


Figure 4.13: Random Forest regression plot: Feature Importance

Random Forest effectively handled both linear and non-linear relationships in the data, leveraging key features such as hour, temperature, and tmax_abs_diff_from_room_temperature,

as shown in the feature importance in Figure 4.13. These features had the most influence on energy consumption, allowing the model to adapt well to different usage patterns.

In Figure 4.11 and Figure 4.12, the time-series and scatter plots demonstrate the model's strong performance. The time-series plot shows that the model closely follows actual energy consumption trends, with only minor deviations during peak periods. The scatter plot indicates a strong correlation between predicted and actual values, with minimal variance, highlighting the model's capability to handle complex consumption patterns effectively.

Overall, the Random Forest Regressor, especially in its second hyperparameter-tuned form, demonstrated excellent forecasting capabilities, handling both trends and non-linear patterns. While improvements between versions were marginal, the model's generalization ability made it one of the best performers.

4.4.4 Gradient Boosting Results

The Gradient Boosting Regressor (GBR) model was evaluated using key metrics such as MAPE, MAE, RMSE, and R^2. The standard model yielded a MAPE of 5.94%, indicating predictions were off by 5.94% on average. The MAE of 694 MW represented the average forecast error, while the RMSE of 906 MW highlighted larger deviations. With an R^2 of 0.862, the model explained 86.2% of the variation in energy consumption.

After hyperparameter tuning of which the hyperparameters are shown in Table 3.4 , the performance improved significantly. The first tuned version reduced the MAPE to 5.09% and RMSE to 740 MW, increasing the R^2 to 0.91. The best results came from the second tuned version, where MAPE decreased to 4.96%, MAE to 568 MW, and RMSE to 718 MW. This version explained 91.4% of the variance, showcasing its ability to handle complex energy consumption patterns. The evaluation results from the gradient boosting experiments are shown in Appendix D.

The feature importance plot in Figure 4.16 revealed that key variables like hour, temperature, and tmax_abs_diff_from_room_temperature had the most influence on consumption, enabling the model to effectively capture both linear and non-linear relationships. Time-series and scatter plots shown in Figure 4.14 and Figure 4.15 respectively demonstrated the model's accuracy, with the tuned versions closely tracking actual consumption and showing strong correlations between predicted and actual values.

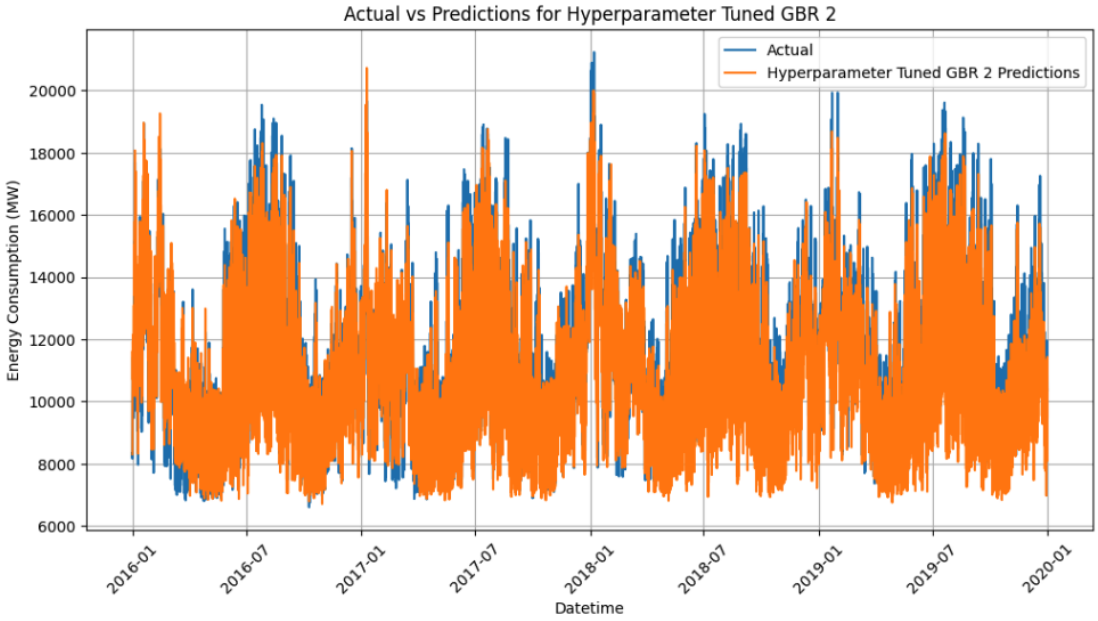


Figure 4.14: Gradient Boosting regression plot: Actual vs Prediction from January 2016 to December 2019

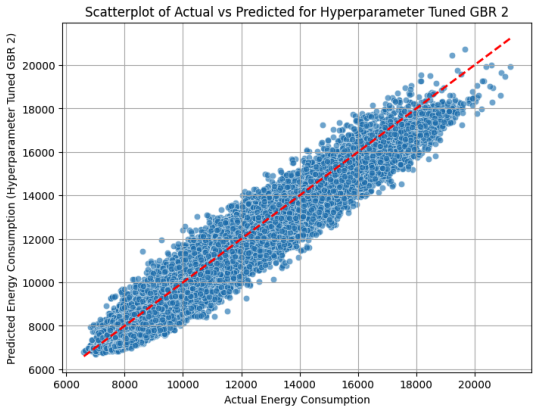


Figure 4.15: Gradient Boosting regression plot: Scatter-plot Actual vs Predicted

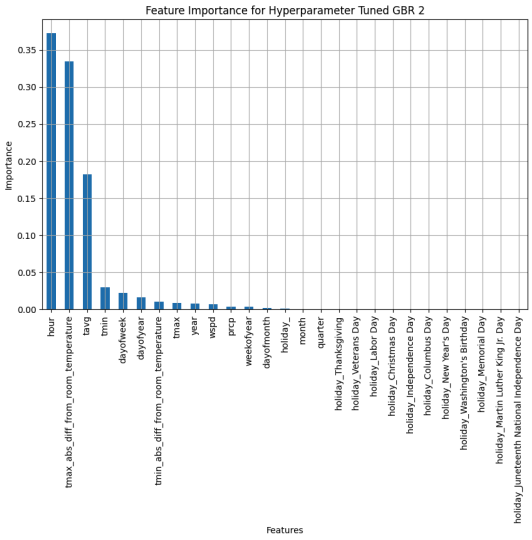


Figure 4.16: Gradient Boosting regression plot: Feature Importance

Overall, the Gradient Boosting Regressor, especially the hyperparameter-tuned versions, provided robust forecasting performance, making it one of the best models for predicting energy consumption patterns.

4.4.5 Extreme Gradient Boosting Results

Extreme Gradient Boosting Regressor the baseline model evaluations returned a MAPE of 5.18%, an MAE of 593 MW, and an RMSE of 755 MW. This was further reflected in its R^2 score of 0.904, which suggests that 90.4% of the variation in energy consumption was accounted for in the model.

After hyperparameter tuning which are shown in Table 3.5, the performance of the model improved. Results for the first tuned version showed a decrease in MAPE to 5.11%, and RMSE to 750 MW with an R^2 of 0.906. While the second tuned version outperformed the rest by further decreasing the MAPE to 4.84%, MAE to 556 MW, and the RMSE to 705 MW, it attained an R^2 of 0.917 thus explaining 91.7% of the variance. XGBoost experiment evaluation results are presented in Appendix D.

The feature importance plot in Figure 4.19 highlights the most influential features for XGBoost, with the highest tmax_abs_diff_from_room_temperature, followed by tavg, and hour being the top predictors of energy consumption. This allowed the model to capture both temporal and weather-related patterns effectively.

In Figure 4.17 and Figure 4.18, the time-series and scatter plots showcase the model's performance. The time-series plot demonstrates that Hyperparameter Tuned XGB 2 closely followed the actual consumption trends, with only minor deviations. The scatter plot shows a strong alignment between actual and predicted values, further validating the model's accuracy.

Overall, XGBoost, especially the hyperparameter-tuned versions, demonstrated robust forecasting capabilities. The improvements in metrics such as MAPE, MAE, and RMSE indicate that XGBoost was able to capture complex relationships and generalize well across different consumption periods.

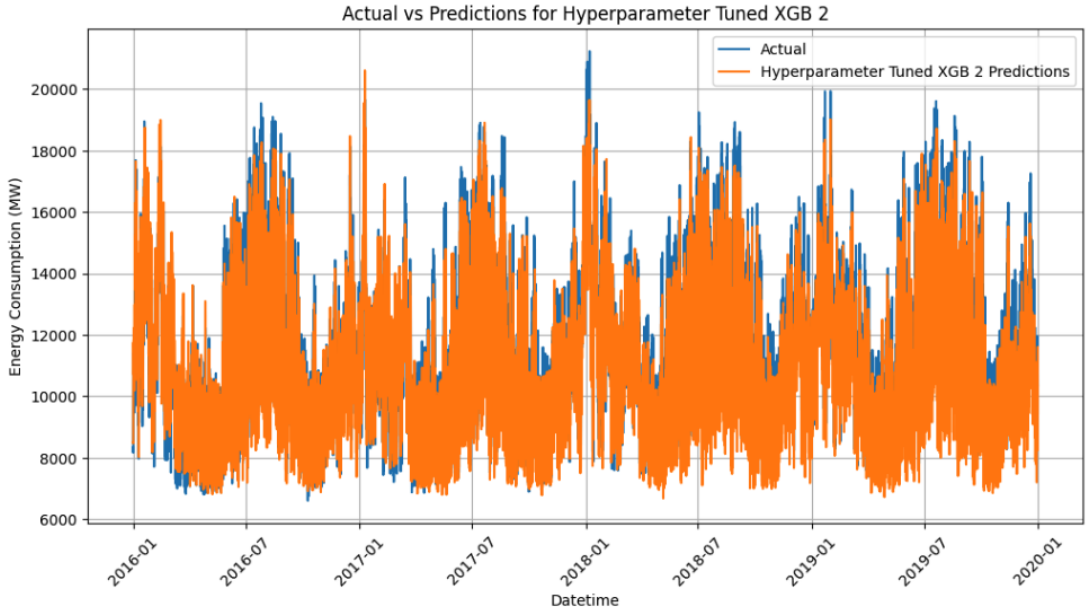


Figure 4.17: Extreme Gradient Boosting regression plot: Actual vs Prediction from January 2016 to December 2019

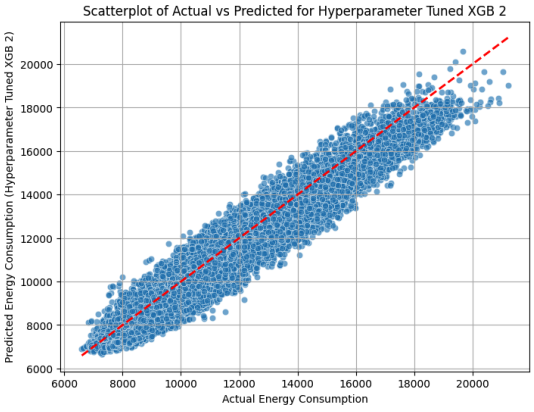


Figure 4.18: Extreme Gradient Boosting regression plot: Scatter-plot Actual vs Predicted

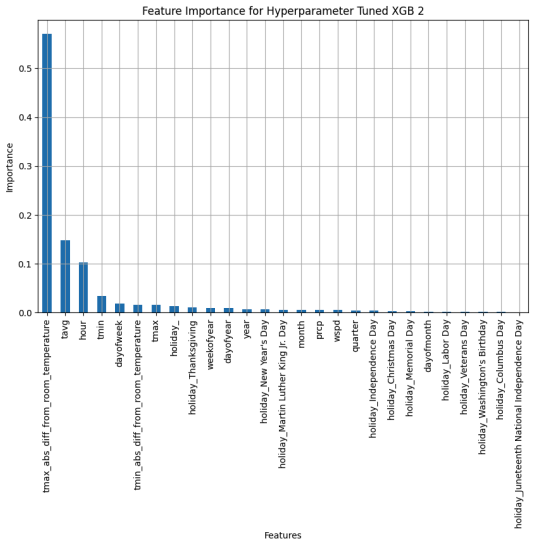


Figure 4.19: Extreme Gradient Boosting regression plot: Feature Importance

4.4.6 Support Vector Regressor (SVR) Results

The Support Vector Regressor (SVR) model was implemented despite initial computational concerns. This choice aimed to explore its capability in capturing non-linearities present in the energy consumption data. Key evaluation metrics, such as MAPE, MAE, RMSE, and R^2, were used to assess its performance. With a MAPE of 12.76%, the model's predictions deviated from

the actual energy consumption by 12.76% on average, while an MAE of 1546 MW indicated significant forecasting errors. The RMSE of approximately 2100 MW further highlighted large discrepancies in certain predictions, and an R^2 score of 0.262 revealed that the model only explained 26.2% of the variation in energy consumption.

As seen in Figure 4.20 and Figure 4.21, which display the time-series and scatter plots, the SVR model failed to capture the more intricate patterns of energy consumption, especially during peak periods. The model consistently under-predicted during times of high energy demand, as indicated by the narrow range of predicted values in comparison to the actual consumption. The scatter plot further demonstrated a low correlation between predicted and actual values, with predictions tightly clustered and diverging from the red dashed line representing the perfect correlation.

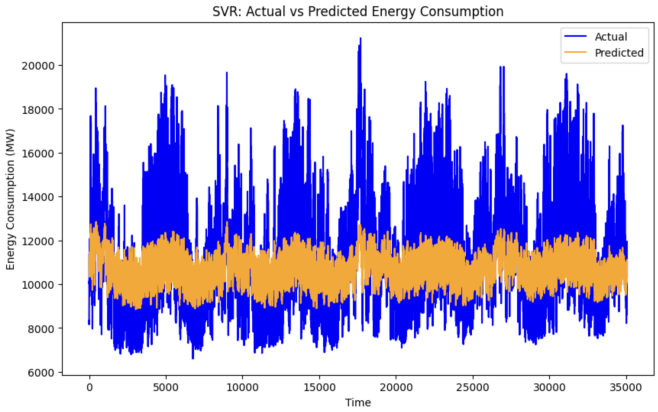


Figure 4.20: Support Vector regression plot: Actual vs Prediction from January 2016 to December 2019

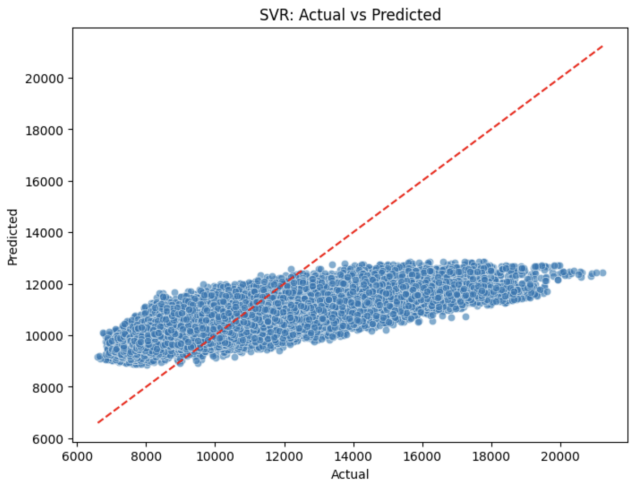


Figure 4.21: Support Vector regression plot: Scatter-plot Actual vs Predicted

Although SVR is known for its ability to model non-linear relationships, the results indicate that the model struggled to accurately capture the non-linear dynamics of energy consumption

in this case. Given its computational intensity and the large-scale nature of the dataset, the SVR model's performance was suboptimal when compared to other models like Decision Trees or Gradient Boosting, which better accounted for the complex relationships between the features.

4.4.7 Neural Network (MLP, CNN, LSTM) Results

Multilayer Perceptron (MLP) Results

As discussed in Chapter 3 the outcomes for the multilayer perceptron models are as follows: the MAPE for MLP Model 1 was 5.82%, while the MAE was 673 MW, the RMSE was 854.94 MW, and the R^2 was 0.878, hence explaining 87.8% of the variance within energy consumption. MLP Model 2 performed slightly worse, with a MAPE of 6.12%, an MAE of 698.74 MW, an RMSE of 872.74 MW, and an R^2 of 0.872.

The hyperparameter-tuned MLP models displayed significant improvements. Hyperparameter Tuned MLP 1 achieved a MAPE of 5.03% and an RMSE of 733.88 MW, which indicates that tuning improved the model's ability to capture complex relationships in the data. Similarly, Hyperparameter Tuned MLP 2 yielded a MAPE of 5.23% and an RMSE of 789.72 MW, further affirming the impact of hyperparameter optimization.

Figure 4.22 shows the residual error distribution, which indicates that most prediction errors were centered around zero, illustrating a good model. Figure 4.23 shows the training and validation loss curves, confirming that the model converged during training, with validation loss closely tracking training loss, a sign of minimal overfitting.

As seen in Figures 4.24 and 4.25, the actual versus predicted scatter plot reveals that the model follows the overall trends in the data but displays some spread in residuals at higher energy consumption levels. The time-series plot further reinforces this, showing that while the model tracks the major consumption trends over time, there is some smoothing of sharp peaks and troughs, particularly during periods of high consumption. These findings suggest that while the MLP captures general trends effectively, more sophisticated architectures like CNNs or LSTMs could further enhance performance

Bayesian Optimization was used for hyperparameter tuning, focusing on units, dropout, learning rate, batch size, and epochs (see Table 3.6). This tuning helped the models capture complex patterns in time, weather, and energy consumption behavior effectively.

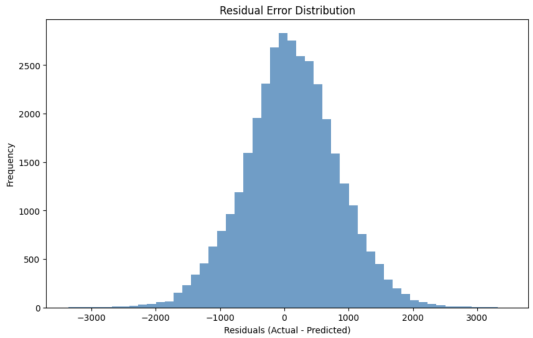


Figure 4.22: MLP Plot: Residual Error Distribution

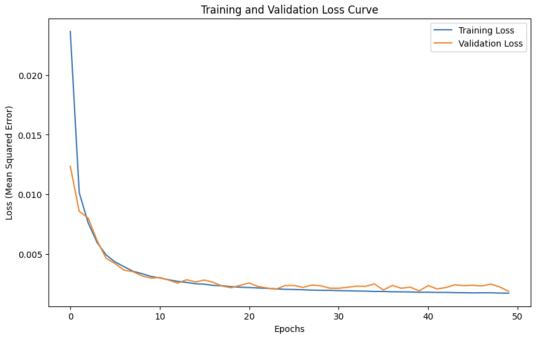


Figure 4.23: MLP Plot: Training and Validation Loss Curves

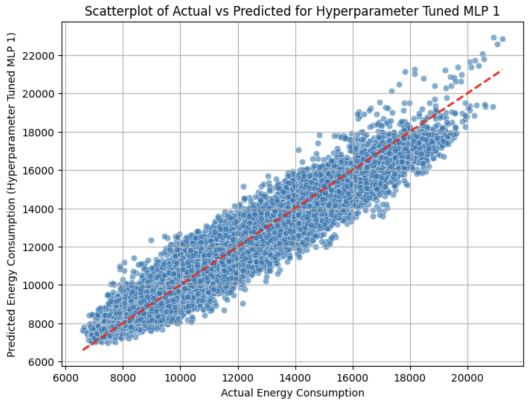


Figure 4.24: MLP Plot: Scatter Plot of Actual vs Predicted Values

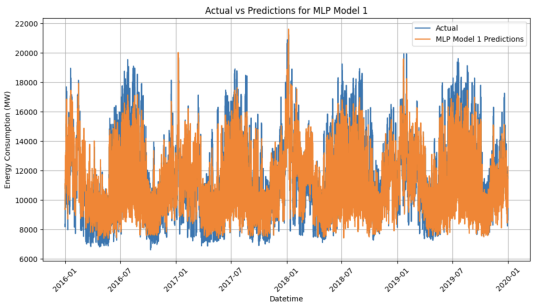


Figure 4.25: MLP Plot: Actual vs Prediction from January 2016 to December 2019

Convolutional Neural Network (CNN) Results

The Convolutional Neural Network (CNN) models for energy consumption forecasting were evaluated using the same metrics as previous models. The first CNN Model result provided a MAPE value of 8.59%, MAE at 964.80 MW, and RMSE at 1243.52 MW with the R^2 value at 0.741 and hence accounted for 74.1% of the variance. Performances for the second CNN model were far higher at a return of a MAPE of 7.11%, MAE of 787.18 MW, RMSE of 994.86 MW, and R^2 at 0.835.

Hyperparameter tuning improved the models further which are shown in Table 3.7. Hyperparameter-Tuned CNN 1 achieved a MAPE of 6.84%, RMSE of 1003.96 MW, and R^2 of 0.831, while Hyperparameter-Tuned CNN 2 had a MAPE of 7.47%, RMSE of 1091.21 MW, and R^2 of 0.801, demonstrating the benefits of tuning.

Figure 4.26 shows that the residual error distribution. Most prediction errors are centered around zero, though there are larger deviations compared to MLP models, indicating higher prediction uncertainty during periods of peak energy consumption. The training and validation loss curves, as shown in Figure 4.27, indicate the model is overfitting.

In Figure 4.28, the scatter plot illustrates that while the CNN models follow the overall data trend, there is residual spread, particularly at higher consumption levels. Figure 4.29 highlights how the CNN models capture general consumption patterns but tend to smooth sharp fluctuations over time.

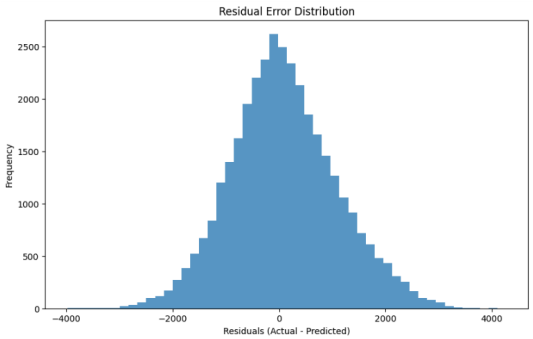


Figure 4.26: CNN Plot: Residual Error Distribution

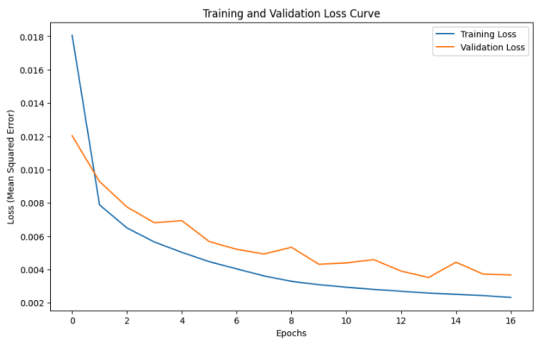


Figure 4.27: CNN Plot: Training and Validation Loss Curves

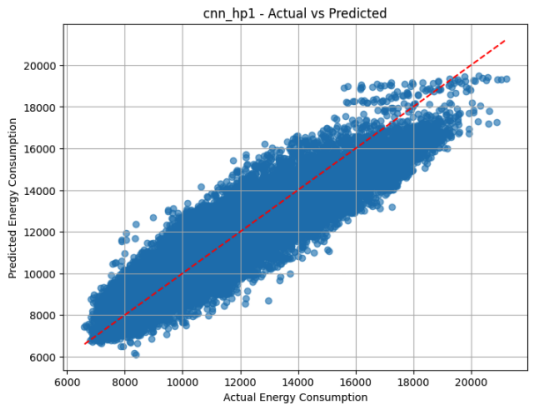


Figure 4.28: CNN Plot: Scatter Plot of Actual vs Predicted Values

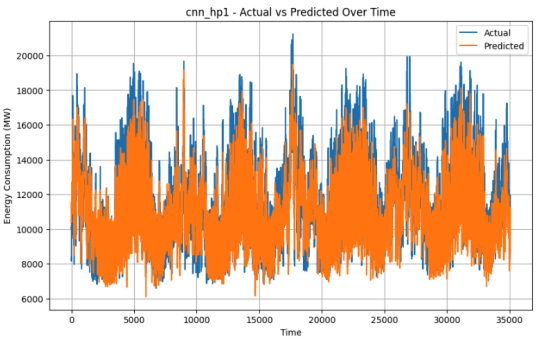


Figure 4.29: CNN Plot: Actual vs Prediction from January 2016 to December 2019

CNN models effectively captured short-term energy trends, though they struggled with sharp spikes and troughs during high consumption periods. Hyperparameter tuning, focusing on filters, kernel size, dropout rate, and dense units, improved model complexity and prediction accuracy. The CNN evaluations are summarized in Appendix D, showing the impact of different hyperparameter settings.

Long Short-Term Memory (LSTM) Results

The LSTMs models were better than the CNN and MLP. LSTM Model 1 was evaluated and the scored a MAPE of 4.64%, MAE of 528.47 MW, RMSE of 673.94 MW, and R^2 of 0.924. LSTM Model 2 performed worse, giving a MAPE of 4.90%, MAE of 565.19 MW, and RMSE of 717.06 MW, with an R^2 of 0.914.

Hyperparameter tuning resulted in minimal improvements. LSTM HP1 returned a MAPE of 4.97%, RMSE of 727.57 MW, and an R^2 of 0.911, while LSTM HP2 had a MAPE of 5.56%, RMSE of 780.45 MW, and an R^2 of 0.898, showing limited gains over the base models.

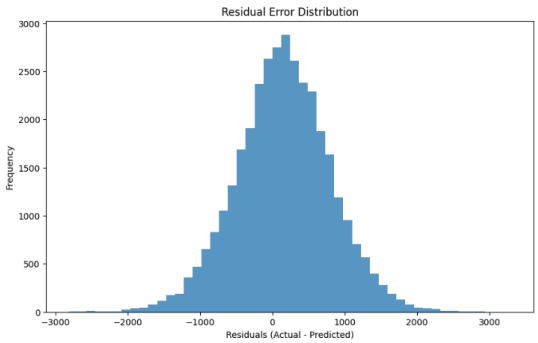


Figure 4.30: LSTM Plot: Residual Error Distribution

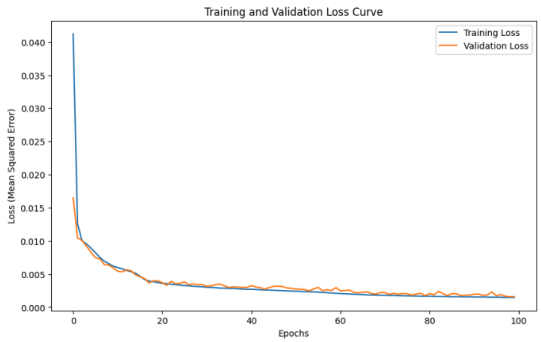


Figure 4.31: LSTM Plot: Training and Validation Loss Curves

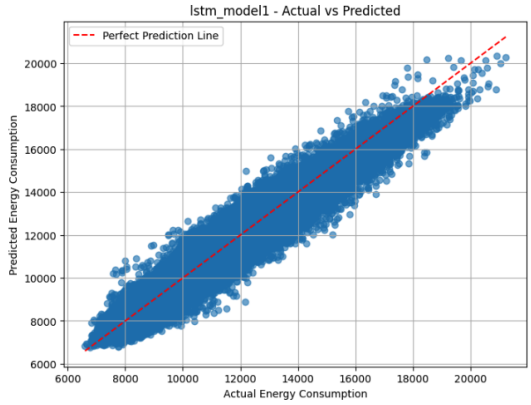


Figure 4.32: LSTM Plot: Scatter Plot of Actual vs Predicted Values

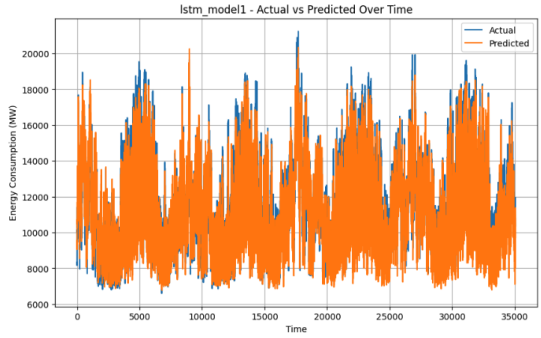


Figure 4.33: LSTM Plot: Actual vs Prediction from January 2016 to December 2019

Figure 4.30 shows the residual error distribution, which reveals most errors centered around zero with minimal skewness, indicating that the model performed well in terms of prediction accuracy. Figure 4.31 demonstrates that both the training and validation loss curves converged smoothly, showing no signs of overfitting and indicating proper generalization capabilities of the model.

In Figure 4.32, the scatter plot reveals that while LSTM models captured general trends,

some spread in residuals appeared, especially at higher consumption levels. Figure 4.33 highlights that although the models tracked overall patterns, they struggled with sharp spikes, particularly during periods of high demand.

LSTM models were effective in capturing long-term dependencies, a strength of recurrent neural networks, but struggled with rapid consumption fluctuations. Hyperparameter tuning, focused on LSTM units, dropout rates, batch size, and epochs (Table 3.8), improved performance, though its impact was less pronounced compared to other models. The LSTM evaluations are shown in Appendix D

4.4.8 Ensemble Learning Results

The ensemble learning models showed strong performance in forecasting energy consumption. The Ensemble_XGB_GBR model achieved a MAPE of 4.85%, MAE of 556.74 MW, RMSE of 703.42 MW, and an R^2 of 0.917, explaining 91.7% of the variance in consumption. The Ensemble_XGB_RFR_GBR model performed slightly better, with a MAPE of 4.73%, MAE of 544.69 MW, RMSE of 692.93 MW, and an R^2 of 0.920, indicating improved accuracy.

Figure 4.34 and Figure 4.36 display the actual vs. predicted energy consumption plots for the two ensemble models over time. Both models accurately tracked the major trends in energy consumption but smoothed out some of the sharp peaks, particularly during periods of high demand. This smoothing behavior is typical for ensemble models that average the outputs of multiple base learners, which reduces prediction variance.

In Figure 4.35 and Figure 4.37, the scatter plots of actual vs. predicted values for both ensemble models reveal that the models closely follow the true energy consumption trends, though some spread is observed at higher consumption levels. This suggests that while ensemble models effectively capture the broader patterns in energy consumption, they may struggle to fully account for extreme values.

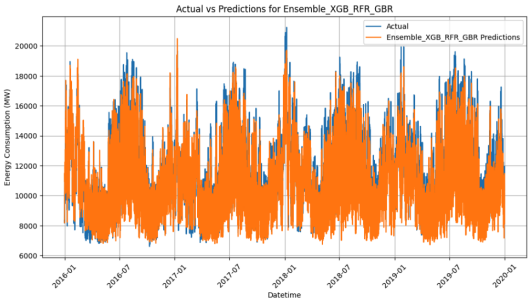


Figure 4.34: Ensemble Plot: Actual vs. Predicted for Ensemble_XGB_RFR_GBR

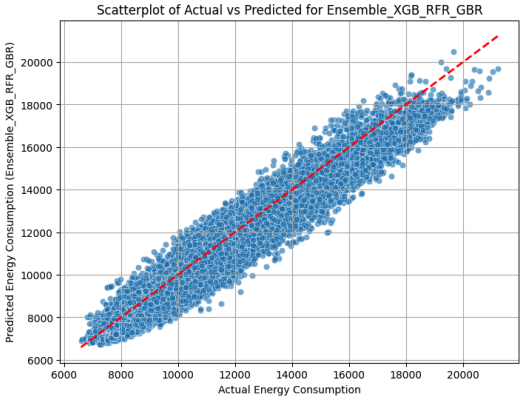


Figure 4.35: Ensemble Plot: Scatter Plot for Ensemble_XGB_RFR_GBR

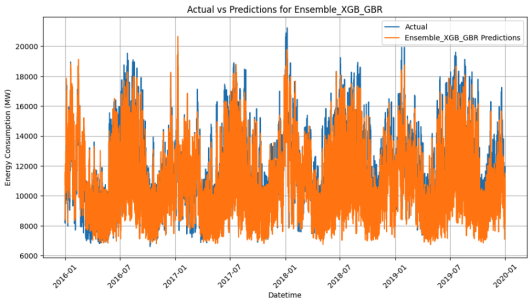


Figure 4.36: Ensemble Plot: Actual vs. Predicted for Ensemble_XGB_GBR

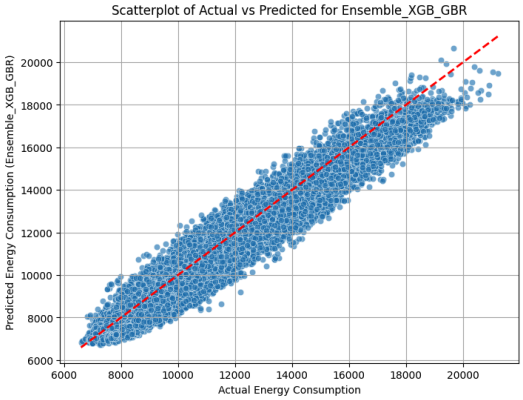


Figure 4.37: Ensemble Plot: Scatter Plot for Ensemble_XGB_GBR

By leveraging the strengths of XGBoost, Random Forest, and Gradient Boosting, ensemble learning improved performance by reducing variance and enhancing prediction accuracy. These models now had the strengths of Random Forest in avoiding overfitting, Gradient Boosting in iteratively correcting its mistakes, and XGBoost in its efficiency toward capturing complex patterns. The results indicated an improvement in the accuracy and stability of energy consumption forecasts.

4.5 Comparative Analysis

4.5.1 Comparison of All Models

Apendix D provides a detailed comparison of all the models tested, showing their performance based on MAPE, MAE, RMSE, and R² scores but below we have the Top 10 models ranked off the evaluation metrics in Table 4.1 below. These metrics allow us to evaluate how effectively each model predicted energy consumption.

Model Name	**Parameter Settings**	**MAPE**	**MAE**	**RMSE**	**R^2**	**Rank**
LSTM Model 1	batch size = 1000, 100 epochs, Adam optimizer	4.64%	528.47	673.94	92.41%	1
Ensemble_XGB_RFR_GBR	Voting Regressor, Hyperparameter tuned (XGB+RFR+GBR)	4.73%	544.69	692.93	91.97%	2
Hyperparameter Tuned XGB 2	n estimators: 100, 200, max depth: 3, 5, 7, learning rate: 0.01, 0.05, 0.1, subsample: 0.8, 1.0, colsample bytree: 0.8, 1.0, gamma: 0, 0.5, 1, reg lambda: 1, 1.5, reg alpha: 0, 0.5	4.84%	556.41	705.37	91.68%	3
Ensemble_XGB_GBR	Voting Regressor, Hyperparameter tuned (XGB+GBR)	4.85%	556.74	703.42	91.73%	4
LSTM Model 2	batch size = 365, 150 epochs, Adam optimizer	4.90%	565.19	717.06	91.40%	5
Hyperparameter Tuned RFR 2	n estimators: 50, 100, 200, max depth: 10, 20, None, min samples split: 2, 5, 7, min samples leaf: 1, 2, 3	4.90%	567.24	733.06	91.02%	6
Random Forest Regressor	None	4.92%	569.17	735.46	90.96%	7
Hyperparameter Tuned RFR 1	n estimators: 50, 100, max depth: 10, 20, None, min samples split: 2, 5, min samples leaf: 1, 2	4.92%	569.17	735.46	90.96%	8
Hyperparameter Tuned GBR 2	n estimators: 50, 100, 150, max depth: 3, 5, 7, learning rate: 0.05, 0.1, subsample: 0.8, 1.0, min samples split: 2, 5, 7, min samples leaf: 1, 2, 3	4.96%	568.65	718.42	91.37%	9
Hyperparameter Tuned LSTM 1	LSTM units: 50-150, Random Search, dropout: 0.1-0.4, batch size: 128, epochs: 50	4.97%	576.64	727.57	91.15%	10

Table 4.1: Model Performance Comparison

4.5.2 Discussion of Best Model

From Table 4, the LSTM proved to be the best in our comparative analysis of forecasting models, with a MAPE of 4.64% and R^2 0.924, thus showing better capability in terms of capturing temporal dependencies. Then, traditional machine learning models also showed different degrees of success for two close followers: XGBoost (MAPE 4.84%, R^2 0.917) and Random Forest (MAPE 4.92%, R^2 0.909).Gradient Boosting and an ensemble combining XGBoost, Random Forest, and Gradient Boosting also performed well. Decision Trees improved significantly upon the Linear Regression baseline but fell short of the more advanced models. Neural network performance varied, with MLPs and CNNs under-performing compared to LSTM. More interestingly, SVR fared worst, even ending up behind the baseline of Linear Regression. The key implications of this analysis are that LSTM performs well in time series forecasting within this context, while it also underlines the very good performance shown by ensemble and boosting methods within traditional machine learning approaches.

4.5.3 Effect of Feature Engineering

The success of the top-performing models can be largely attributed to the inclusion of key engineered features, particularly time-related and temperature-related variables. Features such as hour of the day, temperature deviations from room temperature, and day of the week were consistently identified as significant predictors of energy consumption.

For example, during extreme weather conditions, temperature fluctuations were key drivers of energy demand, particularly for heating and cooling systems. Models like LSTM and the ensemble approaches benefited from these features by accurately predicting consumption spikes during periods of high demand, such as hot summers or cold winters

4.5.4 Trade-offs Between Model Complexity and Accuracy

There is a clear trade-off between model complexity and accuracy. Simpler models like Linear Regression struggled to capture non-linear dependencies in the data, resulting in a higher MAPE of 11%. In contrast, more complex models, such as LSTM and ensemble models, provided significantly better accuracy but required greater computational resources and longer training times, especially during hyperparameter tuning.

Support Vector Regression (SVR), although theoretically capable of handling non linearities, performed poorly in this context. With a MAPE of 12.76% and an R^2 of only 0.262, it became clear that SVR was not well-suited for large-scale energy consumption data. This highlights the importance of selecting models that can efficiently manage both the complexity and scale of the data.

4.6 Discussion of Key Findings

4.6.1 Model explainability with SHAP

The SHAP (SHapley Additive exPlanations) method was used to evaluate feature importance across several models, providing a detailed understanding of how individual features contribute to predictions. The summary plots for each model offer a clear visualization of feature impacts, with the horizontal axis representing the SHAP values (impact on model output) and color gradients indicating feature values (low to high). These results are essential in validating the relationship between the input variables and energy consumption.

Random Forest Regression Model (Hyperparameter Tuned 2)

As shown in Figure 4.38, the SHAP summary plot highlights that tavg, hour, and tmax_abs_diff_from_room_temperature were the most impactful features in the tuned Random Forest model. tavg had the largest SHAP values, significantly influencing energy consumption predictions. The plot also reveals that higher values of tavg and hour tended to increase energy demand, while variations in temperature differentials (tmax_abs_diff_from_room_temperature, tmin_abs_diff_from_room_temperature) also played critical roles.

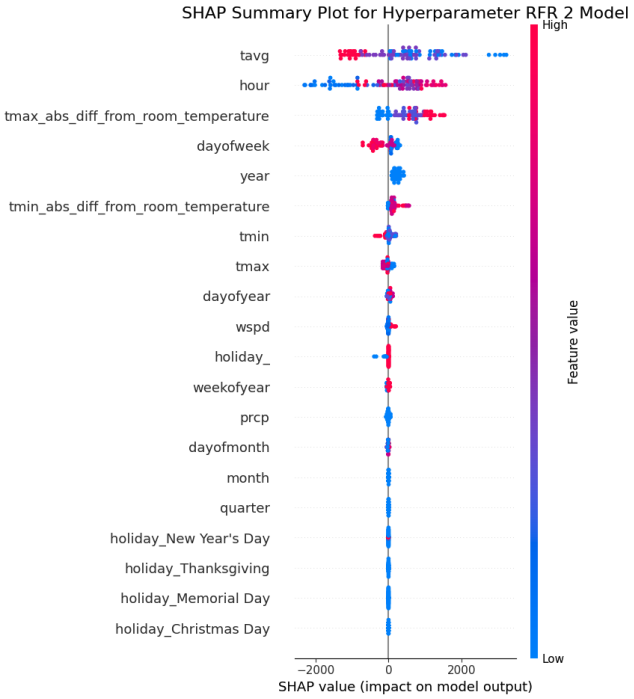


Figure 4.38: SHAP Summary Plots for Random Forest Regression Model (Hyperparameter Tuned 2)

Gradient Boosting Regression Model (Hyperparameter Tuned 2)

In Figure 4.39, the SHAP summary plot for the hyperparameter-tuned Gradient Boosting model underscores the importance of hour, tavg, and tmax_abs_diff_from_room_temperature. Similar to the Random Forest model, these features showed the highest SHAP values, with hour leading in impact. The feature's importance is closely linked to the daily variation in energy demand, which peaks during certain hours. Moreover, weather-related variables such as average temperature and temperature differences further explain fluctuations in consumption.

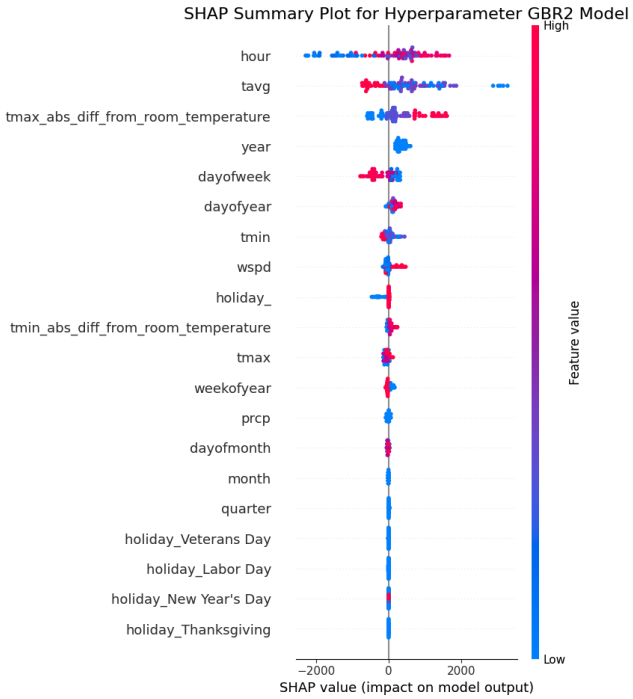


Figure 4.39: SHAP Summary Plots for Gradient Boosting Regression Model (Hyperparameter Tuned 2)

XGBoost Regression Model (Hyperparameter Tuned 2)

Figure 4.40 illustrates the SHAP summary for the XGBoost model, where hour, tavg, and tmax_abs_diff_from_room_temperature again emerge as the dominant features. The XGBoost model's SHAP values indicate that peak consumption hours and temperature deviations from room temperature exert the most influence on energy consumption. Notably, higher tavg values led to an increase in energy usage, demonstrating the model's sensitivity to temperature variations.

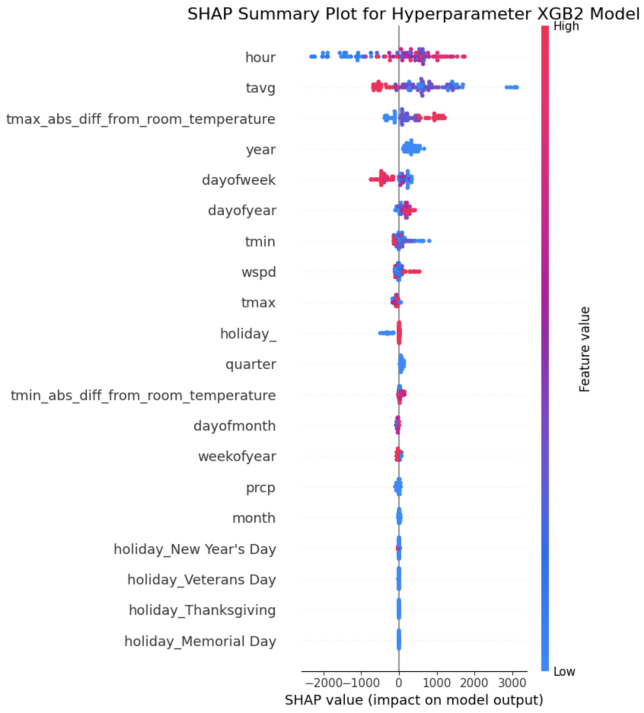


Figure 4.40: SHAP Summary Plots for XGBoost Regression Model (Hyperparameter Tuned 2)

These insights from SHAP not only confirm the predictive relevance of weather and temporal features but also help refine model interpretation, thereby improving the explainability and trust in the machine learning models employed for energy forecasting.

4.6.2 Impact of Hyperparameter Tuning

Hyperparameter tuning was very important as it improved the performance of most models that was used. For example models like Random Forest and XGBoost, their hyperparameter tuning significantly reduced errors and improved generalization. The Random Forest model tuning parameters like n_estimators, max_depth, and min_samples_split helped to reduce overfitting and allowed the model to understand the different patterns in energy forecasting. As a result, the hyperparameter tuned Random Forest achieved a MAPE of 4.90% which is lower than the normal model.

Similarly, in XGBoost, hyperparameter tuning improved performance by increasing the learning rate, depth of trees, and the number of boosting rounds step wise. The MAPE reduced from 5.18% to 4.84%, and increased in the R^2 from 0.904 to 0.917, showing that Hyperparameter tuning enabled XGBoost to capture non-linear relationships in the data more effectively.

The LSTM model implemented in Chapter 3 showed that hyperparameter tuning was not as important as the base LSTM delivered the best performance, with a MAPE of 4.64% and R^2 of 0.924 which was better than the hyperparameter tuned LSTM models. This showed the

capability in handling time-series forecasting tasks because it can capture both short-term and long-term patterns in the energy data.

However, tuning did not yield positive results in all cases. Support Vector Regression (SVR) showed minimal improvement due to the high computational cost and poor scalability of the algorithm, particularly when handling large datasets with complex features. Despite attempts to optimize the kernel function and regularization parameters, the model's MAPE remained high at 12.76%, demonstrating that SVR was not well-suited for the large-scale, non-linear nature of energy consumption forecasting in this study.

4.6.3 Limitations

Several limitations were encountered during the modeling process. The Decision Tree Regressor, while effective in capturing non-linear patterns, exhibited tendencies toward overfitting, particularly when no restrictions were placed on tree depth. The overfitting issue arose because decision trees tend to fit to the noise in the training data, reducing their ability to generalize to unseen data. Hyperparameter tuning, specifically limiting the maximum depth and adjusting the minimum samples per leaf, mitigated this problem to some extent, but the model still underperformed compared to ensemble methods.

Computational constraints were another significant limitation, particularly for SVR and CNN models. The SVR model, due to its quadratic time complexity, became computationally expensive as the dataset size increased, which limited the exploration of complex kernel functions or grid searches for parameter optimization. Similarly, training CNN models required substantial computational resources and time, especially when tuning parameters such as the number of epochs and batch sizes. This limited the ability to test more sophisticated configurations or train the models on longer time horizons.

Furthermore, while feature engineering improved model performance, the reliance on historical weather and holiday data introduced forecast uncertainty, especially during unanticipated events or in scenarios with rapid changes in energy consumption patterns. This highlights the need for dynamic feature updates or the inclusion of real-time data sources in future models.

4.6.4 Practical Implications

Results from this study will be of direct practical relevance to energy providers, especially in respect to grid stability and demand forecasting. The best-performing models, particularly the LSTM and Ensemble learning models, demonstrated a high level of accuracy in predicting energy consumption patterns, which can be leveraged to optimize energy distribution, reduce operational costs, and improve grid reliability.

From Chapter 2, XGBoost and Random Forest models can handle large datasets and complex features like temperature and holiday effects should be integrated into energy management systems to forecast demand spikes during extreme weather conditions or holidays. With correct forecasting, Dominion energy or any energy company will prepare to parallel the quantity either by generating more energy in time or having it distributed so that at no particular time is the grid strained, yet not underutilized.

The LSTM model, with its capacity to capture temporal dependencies, is particularly useful for long-term planning. Energy providers can use this model to predict seasonal variations in energy consumption, allowing for better resource allocation during high-demand periods such as winter heating or summer cooling seasons. Moreover, by identifying consumption patterns that correlate with temperature deviations and time-of-day effects, energy companies can implement demand-response strategies, encouraging consumers to shift usage during peak hours to reduce strain on the grid.

Incorporating these machine learning models into real-time energy management platforms could significantly enhance the ability to monitor and react to fluctuating energy demands, improving overall grid efficiency and reducing the likelihood of outages or excess energy production.

4.6.5 Comparison with Literature

The results of this study are consistent with the findings of existing literature on energy consumption forecasting. Studies such as those by Touzani, Granderson, and Fernandes 2018 and Zeyu Wang et al. 2018 have emphasized the efficacy of ensemble methods like Gradient Boosting and Random Forest in capturing complex, non-linear relationships in energy data. The MAPE scores for Random Forest (4.92%) and XGBoost (4.84%) in this study align with the MAPE values reported in similar studies, reinforcing the robustness of these models in energy forecasting.

The superior performance of the LSTM model in this study is also consistent with findings from Kuo and Huang 2018, who highlighted the strength of recurrent neural networks (RNNs) in time-series forecasting due to their ability to model long-term dependencies. The MAPE of 4.64% achieved by the LSTM in this study is comparable to other studies in the field, where LSTMs have outperformed traditional models in scenarios involving sequential data.

The results are also consistent with Khan et al. 2024 earlier in Chapter 2 the weighted average assigns different weights to the predictions from each model, with the aim of giving more importance to models that perform better, thus enhancing the overall prediction accuracy. The combination Xgboost, Gradient Boosting and Random forest out performed each of their individual evaluations.

On the other hand, the poor performance of SVR mirrors findings from studies such as

Dong, Cao, and S. E. Lee 2005, which noted the computational limitations of SVR when applied to large datasets with complex feature spaces. As in this study, the high computational cost and limited scalability of SVR restricted its effectiveness in energy forecasting, especially when compared to more scalable algorithms like XGBoost or Random Forest.

Additionally, the challenges associated with overfitting in Decision Trees have been widely documented in the literature. Studies such as Tso and Yau 2007 and Shi et al. 2018 note that while decision trees are useful for their interpretability, they often require pruning or ensemble techniques like Random Forest to mitigate overfitting. The findings in this study, where the Decision Tree model was outperformed by ensemble methods, are consistent with these observations.

In conclusion, the results of this study not only align with existing literature but also contribute to the growing body of evidence that ensemble methods and neural networks are among the most effective tools for energy consumption forecasting, especially when combined with robust feature engineering and hyperparameter tuning.

4.7 Summary

The result and discussion chapter has provided an elaborate assessment of the various machine learning models for the energy consumption forecast-from base models like Linear Regression to the Support Vector Regression, then Decision Trees, Random Forest, Gradient Boosting, XGBoost, and neural networks including MLP, CNN, and LSTM. The models were evaluated based on their ability to understand the linear and non-linear relationships within the data, with performance metrics such as MAPE, MAE, RMSE, and R^2 being used for comparative analysis. Based off model evaluations, LSTM was the best-performing model, with MAPE of 4.64% and an R^2 of 0.924, making it highly effective in capturing temporal dependencies in the energy consumption data. Ensemble methods, especially combining XGBoost, Random Forest, and Gradient Boosting, showed a strong performance, with MAPE values as low as 4.73%, validating their capability in capturing complex patterns in the data.

Feature engineering, particularly the inclusion of time and weather-related variables, was critical in enhancing model performance. Hyperparameter tuning further improved accuracy in models like Random Forest and XGBoost, but not all models benefited equally, as SVR continued to underperform due to its computational inefficiencies and inability to handle the scale of the dataset.

Overall, the results demonstrated that advanced machine learning models, particularly LSTM and ensemble approaches, provide robust forecasting capabilities for energy consumption, with practical implications for energy providers in optimizing grid stability, reducing operational costs, and improving demand forecasting.

Chapter 5

Case Study

5.1 Introduction to the Case Study

This chapter presents a case study on energy consumption forecasting from 1st January 2020 to 31st July 2024 using the top 3 machine learning models developed earlier in the research. The chapter, therefore, is structured into the following sections: First, the description of the case study data gives an overview of the data characteristics and preprocessing. Then, it follows that the Application of the Developed Models details the application of the predictive models to the case study data. The Case Study Results section covers the performance assessment of the models concerning key metrics. Further, Discussion analyzes the implications of the results presented in this chapter, focusing on the comparisons between the models and their limitations. Lastly, Summary concludes this chapter by underlining the most important findings and their contributions to energy forecasting.

5.2 Description of the Case Study Data

The case study data, spanning from 2020 to 2024, captures a notable increase in energy consumption during this period, as illustrated in Figure 3.5. This rise may be linked to various factors, including economic growth, population increase, the shift to remote work following the COVID-19 pandemic, and regional infrastructure expansions. The dataset used for the analysis contains 40,174 observations across 30 features, with an average energy consumption of 12,690 MW, a minimum value of 7,438 MW, and a distinct upward trend in consumption. This period has been specifically selected for the case study to evaluate the performance of the developed models on this more recent and dynamic dataset.

5.3 Application of the Developed Models

The Data preprocessing techniques used in Chapter 3 to create weather and Temporal features were also applied to the case study data. The best models developed and implemented in Chapter 4 were reintroduced for the case study, The saved model was loaded and retrained on data from January 2020 to June 2023. The data was split into training and testing sets based on a cutoff date of July 1, 2023, with the training data covering 2020–2023 and the test data covering the remainder of 2024. The features were normalized before retraining the model, and predictions were made on the test data. Model performance was evaluated using key metrics, including RMSE, MAE, MAPE, and R² from results analsis section in Chapter 3, which demonstrated the model's accuracy on the 2024 data. Finally, the retrained model was saved, and a CSV file comparing actual energy consumption to predicted values was exported for further analysis.

5.4 Results of the Case Study

The evaluation metrics results of the top 3 model implemented on the case study data are shown in Table 5.1 below. These metrics allow us to evaluate how effectively each model predicted energy consumption.

Model Name	**Parameter Settings**	**MAPE**	**MAE**	**RMSE**	**R^2**	**Rank**
LSTM Model 1	batch size = 1000, 100 epochs, Adam optimizer	2.91%	422.35	581.10	94.54%	1
Hyperparameter Tuned XGB 2	n estimators: 100, 200, max depth: 3, 5, 7, learning rate: 0.01, 0.05, 0.1, subsample: 0.8, 1.0, colsample bytree: 0.8, 1.0, gamma: 0, 0.5, 1, reg lambda: 1, 1.5, reg alpha: 0, 0.5	5.02%	732.44	934.62	85.88%	2
Ensemble_XGB_RFR_GBR	Voting Regressor, Hyperparameter tuned (XGB+RFR+GBR)	5.17%	754.66	955.62	85.23%	3

Table 5.1: Model Performance Comparison

5.5 Discussion

Table 5.1 shows that the LSTM model is the best performing model with a MAPE of 2.91%, MAE of 422.35 MW, RMSE of 581.10 MW, and an R² score of 94.54%. The LSTM improved when compared to its performance in Chapter 4, where it had a MAPE of 4.64%. The performance drop seen in the Ensemble and Hyperparameter tuned XGBoost models can be

attributed to the dynamic nature of energy consumption in the 2024 period, which may have introduced more complex patterns not as effectively captured by these models. The Hyper-parameter tuned XGBoost performed better than ensemble method but still showed a slight decrease in accuracy compared to Chapter 4 due to its averaging effect, which may smooth out important fluctuations in the data.

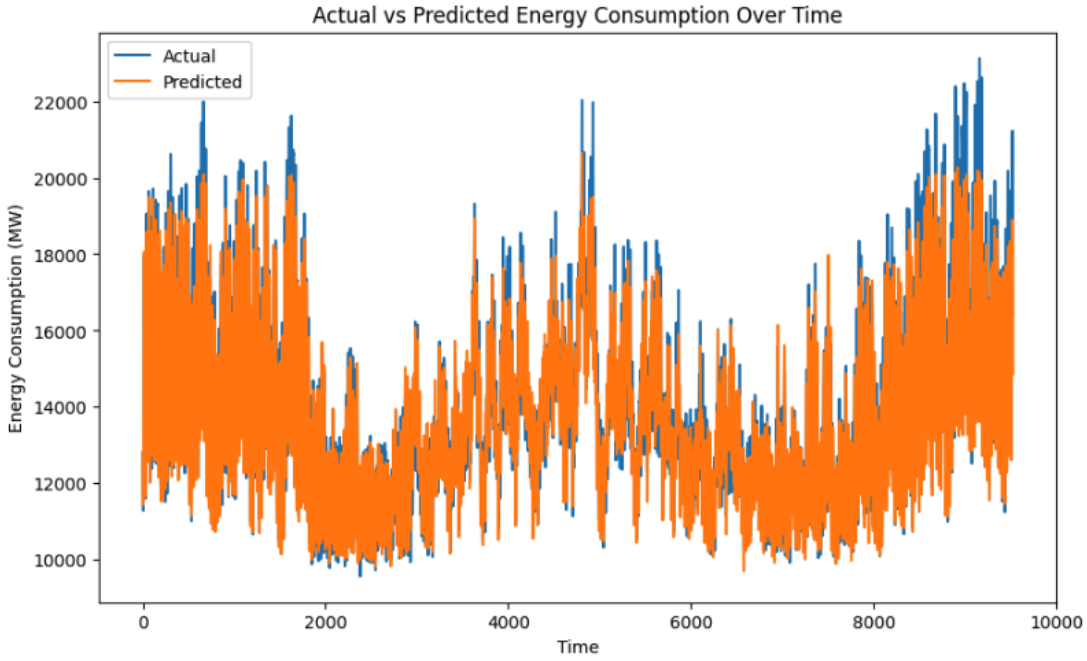


Figure 5.1: LSTM Case Study Plot: Actual vs Prediction from July 2023 to July 2024

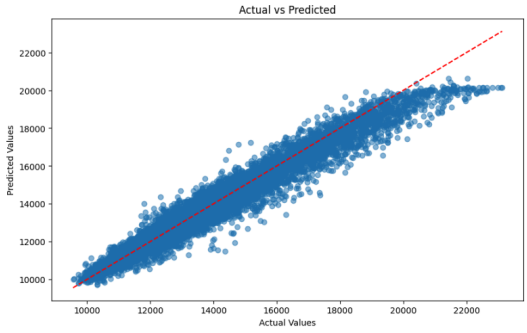


Figure 5.2: LSTM Case Study Plot: Scatter Plot of Actual vs Predicted Values

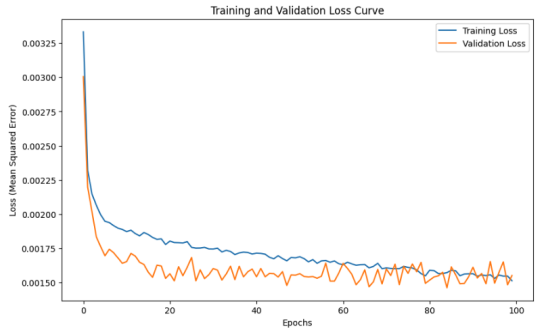


Figure 5.3: LSTM Case Study Plot: Training and Validation Loss Curves

The LSTM plots Figure 5.1 Actual vs Predicted, Figure 5.2 Scatterplot, and Figure 5.3 Loss Curves provide additional insights. The time-series plot shows a close alignment between actual and predicted values, with the LSTM model effectively capturing both long-term trends and short-term fluctuations in energy consumption. The scatterplot, which compares actual

vs predicted values, shows a strong correlation, although some residual variance is noticeable at higher consumption levels. The training and validation loss curves indicate that the LSTM model converged well without signs of overfitting, with validation loss closely tracking the training loss, which is a positive indicator for model generalization. These observations confirm the superiority of LSTM in time-series forecasting, as highlighted in Chapter 4.

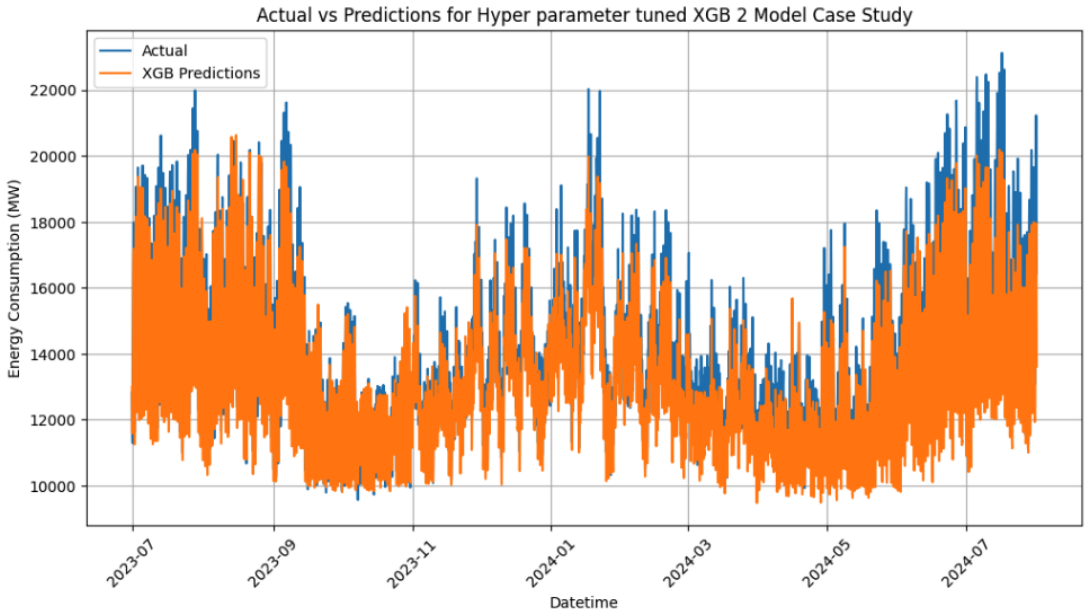


Figure 5.4: Hyperparameter Tuned Xgboost 2 Case Study plot: Actual vs Prediction from July 2023 to July 2024

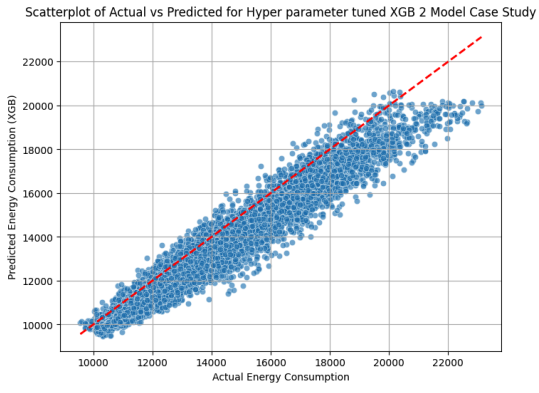


Figure 5.5: Hyperparameter Tuned Xgboost 2 Case Study plot: Scatter Plot of Actual vs Predicted Values

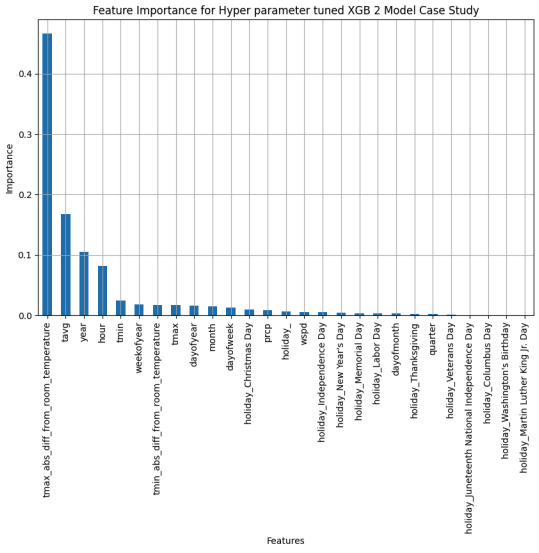


Figure 5.6: Hyperparameter Tuned Xgboost 2 Case Study plot: Feature Importance

The XGBoost model's plots, including the scatterplot and actual vs predicted energy consumption in Figure 5.5 and Figure 5.4 respectively, reveal strong predictive capability but with more noticeable deviations compared to the LSTM. The scatterplot shows a strong alignment between actual and predicted values, though it suffers from slight over-predictions during peak consumption periods. The feature importance plot in Figure 5.6 emphasizes that temperature-related variables, such as the difference between maximum temperature and room temperature, played the most significant role in the model's predictions. This confirms the model's sensitivity to weather conditions, which aligns with findings from Chapter 4, where similar variables drove the model's predictive power.

5.6 Summary

In summary, the case study demonstrated the practical application of the top-performing models for energy consumption forecasting between 2020 and 2024. The LSTM model outperformed the other models, achieving the highest accuracy with a MAPE of 2.91% and an R^2 score of 94.54%. This significant improvement over its performance in Chapter 4 highlights its capability in capturing both long-term and short-term trends in energy consumption. The hyperparameter-tuned XGBoost model and the ensemble method, while still performing well, exhibited a slight drop in accuracy due to the dynamic nature of the 2024 data, particularly in capturing more complex patterns. These findings underscore the strengths of recurrent neural networks in time-series forecasting, while also emphasizing the importance of feature selection, particularly temperature-related variables, for boosting model performance.

Chapter 6

Further Work

6.1 Introduction

This chapter addresses challenges faced during the project and suggests future improvements in energy consumption forecasting using machine learning. While the research met its objectives, issues related to data collection, model development, and computational limitations emerged, offering opportunities for future exploration to enhance model accuracy and applicability.

6.2 Data Collection and Preprocessing

A key challenge was the limited availability of granular weather data, and I also had issues using the API provided by PJM Interconnection LLC, soeach year of the data consumption was downloaded Manually. To resolve this, future work should try to implement data extraction using the API which can ensure seamless data collection. Incorporating micro-level weather variations, real-time consumption, and customer count data could further improve performance. Future studies should consider higher-resolution data sources or smart meter systems to enhance short-term forecasts.

Feature engineering primarily focused on daily weather and temporal trends. However, using hourly weather data could further fine-tune the model's hourly predictions. Additionally, integrating lagged or auto-regressive features which are valuable in time series forecasting would enhance forcast accuracy.

6.3 Model Development and Experimentation

Several models (XGBoost, Random Forest, LSTM, MLP) were successfully developed, but deep learning models like CNN faced computational challenges due to their complexity. To

address this, future work should explore cloud-based infrastructures like AWS, GCP or Azure, which offer scalable resources for training deeper models and more complex hyperparameter tuning.

Transfer learning is another area to explore. Pre-trained models from similar cloud domains can be fine-tuned for specific energy forecasting tasks, reducing training time and improving accuracy.

Another area for future exploration is the development of ensemble models that combine traditional machine learning techniques with deep learning approaches. While ensemble learning was applied in this research, a more diversified ensemble could include models from different families (e.g., statistical models, decision trees, neural networks), potentially resulting in more robust forecasts. Also, another form of machine learning should be explored, such as hybrid models.

6.4 Managing Uncertainty and Enhancing Model Explainability

Forecasting energy consumption is a very challenging task since there are lots of uncertainties generated by external factors such as meteorological or economic changes. Current models leveraging machine learning methods have been quite accurate but cannot handle uncertainty. Future work should develop methods that allow quantification of uncertainty-Bayesian Neural Networks or Quantile Regression-to come up with probabilistic forecasts. These techniques offer several probable outcomes that are of great use to energy managers in decision-making, especially during peak demand periods or periods when there is a shortage in supply.

Moreover, explainability remains crucial, especially for deep learning models like LSTMs and CNNs, which are often considered ”black boxes.” While SHAP plots were useful in this study, future work should explore more sophisticated explainability tools such as LIME, Integrated Gradients, or Deep SHAP. These techniques can provide more detailed insights into how features like temperature or time of day influence predictions, fostering greater trust in the models' results. Enhanced interpretability will not only build confidence among stakeholders but also guide further model improvements.

Additionally, applying interpretability tools to time-series models can offer deeper understanding of how temporal patterns are captured. Analyzing LSTM layers or CNN filters for seasonal trends, for instance, could provide actionable insights for energy managers and support more robust forecasting.

6.5 Broader Applications and Future Case Studies

Though this research focused on energy consumption forecasting for the PJM DOM region, future work should explore applying these models to other sectors—industrial, residential, or transportation—which have unique consumption patterns. Comparative studies across sectors or regions (e.g., Europe vs. U.S.) would reveal the generalizability of different models, highlighting which perform best under specific conditions.

As renewable energy sources like solar and wind become integral to modern grids, future models should incorporate production data from these sources. This would help energy planners manage the variability of renewable energy and improve demand-supply balancing.

Future studies should also consider developing real-time forecasting systems. With advances in edge computing and real-time data platforms, models could be continuously updated, offering instant recommendations for grid operators and energy managers. Additionally, transfer learning could accelerate the adaptation of pre-trained models to new regions or sectors, reducing the time required to deploy forecasting systems.

Finally, longitudinal case studies spanning multiple years and seasons would help evaluate model performance under varying conditions, such as economic downturns or extreme weather. Such studies would provide invaluable feedback, ensuring that machine learning models are not only effective but practical for real-world energy forecasting.

6.6 Summary

This research has established a solid foundation for energy consumption forecasting using machine learning. However, future work could enhance model accuracy and practical application by incorporating higher-resolution data, advanced optimization techniques, cloud computing resources, and uncertainty quantification. Addressing these aspects will further contribute to global energy sustainability efforts.

Chapter 7

Conclusion

This project started with a simple aim which was to develop and evaluate machine learning models to forecast energy consumption. Four objectives were set: model development, performance evaluation, identifying consumption drivers, and a case study application. All objectives have been met, as follows:

1. Model Development: Various machine learning models, including linear regression, Random Forest, Gradient Boosting, and neural networks, were successfully developed for energy consumption forecasting (Chapter 3).

2. Performance Evaluation: The models were rigorously evaluated using RMSE, MAE, and MAPE metrics. XGBoost emerged as the best-performing model, achieving the highest accuracy while balancing computational efficiency (Chapter 4).

3. Identifying Key Drivers: The SHAP analysis revealed that weather variables, especially temperature, significantly influenced energy consumption patterns, affirming the importance of feature engineering (Chapter 4).

4. Case Study Application: The developed models were validated on a real-world data from the PJM Interconnection which focused on the Dominion Energy region, demonstrating their practical applicability in forecasting energy consumption for large-scale systems ((Chapter 5)).

The model was measured using the evaluation metrics which showed the performances of various machine learning models used in chapter 3 (Linear Regression, Decision tree, Random Forest, Gradient Boosting, XGBoost, Ensemble learning (Gradient Boosting, XGBoost) , Ensemble learning(Random Forest, Gradient Boosting, XGBoost) and neural networks (MLP, CNN, and LSTM)). All the models were tested using RMSE, MAE, MAPE, and R^2 metrics and the best Performing Model was LSTM with a MAPE (4.64%), MAE (528 MW), RMSE

(673 MW), R^2 (92.41%). Ensemble_XGB_RFR_GBR and the Hyperparameter Tuned XGB 2 were also top performers with their results as follows MAPE (4.73%), MAE (544.69 MW), RMSE (692 MW), R^2 (91.97%) and MAPE (4.84%), MAE (556 MW), RMSE (705 MW), R^2 (91.68%) respectively.The LSTM model was the best at capturing temporal and weather related patterns, showcasing its strength in handling sequential data. Feature importance analysis indicated that temperature and time-related features were among the most significant drivers of energy consumption, this was also confirmed using SHAP.

The case study results indicated that the LSTM model outperformed others, achieving a MAPE (2.91%), an MAE (422 MW), and an R^2 (94.54%). The hyperparameter-tuned XGBoost model and the ensemble method, although performing well, displayed slightly lower accuracy, with MAPE values of 5.02% and 5.17%, respectively. These findings underscore the effectiveness of machine learning models, particularly neural networks, in real-world energy consumption forecasting. However, challenges such as the limited granularity of weather data and the computational complexity of neural networks point to opportunities for further refinement

Though we were able to build success models, some models like SVR and CNN performed poorly. The models faced several limitation which are highlighed in the limitation subsection in Chapter 4. The Decision Tree Regressor, while good at capturing complex patterns, often overfit the training data, especially without proper depth limitations. This made it hard for the model to perform well on new data. Although tuning helped, it still fell short compared to other models.

Another challenge was the limited computing power which presented an obstacle, especially for the SVR and CNN models. The SVR's high computational cost made it tough to explore more complex settings. Similarly, training CNNs was slow and resource-intensive, which restricted the ability to fine-tune them. Several proposed solution to this limitations are discussed in further works Chapter 6

References

Akpan, Usenobong Friday and Godwin Effiong Akpan (2012). "The contribution of energy consumption to climate change: a feasible policy direction". In: *International Journal of Energy Economics and Policy* 2.1, pp. 21–33.

Amasyali, Kadir and Nora M El-Gohary (2018). "A review of data-driven building energy consumption prediction studies". In: *Renewable and Sustainable Energy Reviews* 81, pp. 1192–1205.

Arghira, Nicoleta et al. (2013). "Forecasting energy consumption in dwellings". In: *Advances in intelligent control systems and computer science*, pp. 251–264.

Bassi, Abnash et al. (2021). "Building energy consumption forecasting: A comparison of gradient boosting models". In: *Proceedings of the 12th International Conference on Advances in Information Technology*, pp. 1–9.

Cryer, Jonathan D and Natalie Kellet (1991). *Time series analysis*. Springer.

Cunningham, Pádraig, Matthieu Cord, and Sarah Jane Delany (2008). "Supervised learning". In: *Machine learning techniques for multimedia: case studies on organization and retrieval*. Springer, pp. 21–49.

Deb, Chirag et al. (2017). "A review on time series forecasting techniques for building energy consumption". In: *Renewable and Sustainable Energy Reviews* 74, pp. 902–924.

Di Persio, Luca and Nicola Fraccarolo (2023). "Energy consumption forecasts by gradient boosting regression trees". In: *Mathematics* 11.5, p. 1068.

Dong, Bing, Cheng Cao, and Siew Eang Lee (2005). "Applying support vector machines to predict building energy consumption in tropical region". In: *Energy and Buildings* 37.5, pp. 545–553.

Gupta, Aakash, Ankur Bansal, Kshitij Roy, et al. (2021). "Solar energy prediction using decision tree regressor". In: *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, pp. 489–495.

Hu, Yusha and Yi Man (2023). "Energy consumption and carbon emissions forecasting for industrial processes: Status, challenges and perspectives". In: *Renewable and Sustainable Energy Reviews* 182, p. 113405.

Jiang, Tammy, Jaimie L Gradus, and Anthony J Rosellini (2020). "Supervised machine learning: a brief primer". In: *Behavior therapy* 51.5, pp. 675–687.

Jozi, Aria et al. (2019). "Decision support application for energy consumption forecasting". In: *Applied Sciences* 9.4, p. 699.

Khan, Shahzeb Ahmad et al. (2024). "Effective Voting-based Ensemble Learning for Segregated Load Forecasting with Low Sampling Data". In: *IEEE Access*.

Kim, Yang-Seon et al. (2024). "Investigating the Impact of Data Normalization Methods on Predicting Electricity Consumption in a Building Using different Artificial Neural Network Models." In: *Sustainable Cities and Society*, p. 105570.

Kumar, Vijendra et al. (2023). "Advanced machine learning techniques to improve hydrological prediction: A comparative analysis of streamflow prediction models". In: *Water* 15.14, p. 2572.

Kuo, Ping-Huan and Chiou-Jye Huang (2018). "A high precision artificial neural networks model for short-term energy load forecasting". In: *Energies* 11.1, p. 213.

Lee, YW, KG Tay, and YY Choy (2018). "Forecasting electricity consumption using time series model". In: *International Journal of Engineering & Technology* 7.4.30, pp. 218–223.

Ljung, Greta M, J Ledolter, and B Abraham (2014). "George Box's contributions to time series analysis and forecasting". In: *Applied stochastic models in business and industry* 30.1, pp. 25–35.

Ma, Minglu and Zhuangzhuang Wang (2019). "Prediction of the energy consumption variation trend in South Africa based on ARIMA, NGM and NGM-ARIMA models". In: *Energies* 13.1, p. 10.

Mahesh, Batta (2020). "Machine learning algorithms-a review". In: *International Journal of Science and Research (IJSR).[Internet]* 9.1, pp. 381–386.

Moon, Jihoon et al. (2019). "A comparative analysis of artificial neural network architectures for building energy consumption forecasting". In: *International Journal of Distributed Sensor Networks* 15.9, p. 1550147719877616.

Ospina, Raydonal et al. (2023). "An overview of forecast analysis with ARIMA models during the COVID-19 pandemic: Methodology and case study in Brazil". In: *Mathematics* 11.14, p. 3069.

Ozturk, Suat and Feride Ozturk (2018). "Forecasting energy consumption of Turkey by Arima model". In: *Journal of Asian Scientific Research* 8.2, p. 52.

Pourahmadi, Mohsen (2001). *Foundations of time series analysis and prediction theory*. Vol. 379. John Wiley & Sons.

Sauer, João et al. (2022). "Extreme gradient boosting model based on improved Jaya optimizer applied to forecasting energy consumption in residential buildings". In: *Evolving Systems*, pp. 1–12.

Shcherbakov, Maxim, Valeriy Kamaev, and Nataliya Shcherbakova (2013). "Automated electric energy consumption forecasting system based on decision tree approach". In: *IFAC Proceedings Volumes* 46.9, pp. 1027–1032.

Shi, Kunpeng et al. (2018). "An improved random forest model of short-term wind-power forecasting to enhance accuracy, efficiency, and robustness". In: *Wind energy* 21.12, pp. 1383–1394.

Touzani, Samir, Jessica Granderson, and Samuel Fernandes (2018). "Gradient boosting machine for modeling the energy consumption of commercial buildings". In: *Energy and Buildings* 158, pp. 1533–1543.

Tso, Geoffrey KF and Kelvin KW Yau (2007). "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks". In: *Energy* 32.9, pp. 1761–1768.

Tunç, Murat, Ünal Çamdali, and Cem Parmaksizoğlu (2006). "Comparison of Turkey's electrical energy consumption and production with some European countries and optimization of future electrical power supply investments in Turkey". In: *Energy Policy* 34.1, pp. 50–59.

Wang, Zeyu et al. (2018). "Random Forest based hourly building energy prediction". In: *Energy and Buildings* 171, pp. 11–25.

Yuan, Chaoqing, Sifeng Liu, and Zhigeng Fang (2016). "Comparison of China's primary energy consumption forecasting by using ARIMA (the autoregressive integrated moving average) model and GM (1, 1) model". In: *Energy* 100, pp. 384–390.

Appendix A

Terms of Reference and Ethics

Department of Computing and Mathematics Computing and Digital Technology Postgraduate Programmes Terms of Reference Coversheet	
Student name:	Anthony Anayo Eze
University I.D.:	22536729
Academic supervisor:	Anthony Bukowski
External collaborator (optional):	
Project title:	Forecasting Energy Consumption Using Machine learning
Degree title:	MSc Data Science
Project unit code:	6G7V0007_2223_9F
Credit rating:	
Start date:	4-07-2023
ToR date:	4-07-2023
Intended submission date:	29-09-2023
Signature and date student:	 4-07-2023
Signature and date external collaborator (if involved):	

This sheet should be attached to the front of the completed ToR and uploaded with it to Moodle.

Manchester Metropolitan University
MSc Data Science
2023/2024
Project Terms of Reference

Table of Contents

PROJECT TITLE 4

PROJECT BACKGROUND 4

AIM 5

PRODUCT AIMS 5

OBJECTIVES 5

HARDWARE RESOURCES REQUIRED 6

SOFTWARE RESOURCES REQUIRED 6

PROJECT DELIVERABLES 7

PROJECT PLAN 8

Project Title

Forecasting Energy Consumption Using Machine learning

Project Background

Global energy consumption has witnessed a substantial 100% increase over the past 40 years, as reported by a study (Wang et al., 2018). This surge in energy demand and the subsequent energy crisis have compelled individuals to adopt more efficient energy usage practices and make lifestyle adjustments. Notably, buildings have emerged as the predominant consumer of energy worldwide, with people spending more than 90% of their daily lives indoors (Wang et al., 2018). Consequently, addressing energy consumption in buildings has become imperative in the pursuit of sustainable energy practices.

Time series forecasting is an essential technique used to predict future values based on historical data. It finds applications in diverse domains such as finance, meteorology, healthcare, and energy consumption (Cerqueira et al., 2020; Hong et al., 2020). This project aims to develop a machine learning model specifically tailored for accurately forecasting energy consumption, utilizing advanced algorithms to analyze historical energy data and make precise predictions.

Accurate energy consumption forecasting is crucial for effective resource planning, energy management, and sustainability initiatives. Traditional statistical methods often struggle to capture the complex temporal dependencies and patterns inherent in energy consumption data. Machine learning algorithms, such as Long Short-Term Memory (LSTM), Support Vector Regression (SVR), and Random Forest Regression (RFR), have shown promise in capturing and leveraging these patterns, making them suitable for energy consumption forecasting.

The project entails collecting and pre-processing relevant historical energy consumption data. This process includes data cleaning, addressing missing values, and potentially performing feature engineering to enhance the dataset's quality and suitability for machine learning models. The student will then train and fine-tune the selected machine learning models using the prepared dataset, aiming to optimize their performance.

To evaluate the models' performance, several appropriate metrics, including mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R-squared), will be utilized. Comparisons between different models will be made, and their respective strengths and limitations will be analyzed to understand their effectiveness for energy consumption forecasting tasks (Xiao et al., 2015).

Furthermore, the project will explore the integration of additional input features to enhance prediction accuracy. Factors such as weather data, time of day, and historical energy consumption patterns will be investigated to assess their impact on the forecasting models. Additionally, the project will delve into techniques for model interpretability, aiming to uncover the underlying factors influencing energy consumption patterns.

The successful completion of this project will equip the student with valuable knowledge and skills in machine learning-based time series forecasting for energy consumption. The outcomes of the project will contribute to improved resource planning, energy management, and sustainability initiatives. The accurate predictions obtained will enable stakeholders to make informed decisions, optimize energy usage, and reduce costs.

Aim

1. Develop a machine learning model tailored for accurate energy consumption forecasting.

2. Explore and compare the performance of advanced machine learning algorithms such as LSTM, SVR, and RFR for energy consumption prediction.

3. Evaluate and analyze the strengths and limitations of the selected models using appropriate evaluation metrics.

4. Investigate the impact of integrating additional input features, such as weather data, time of day, and historical energy consumption patterns, on prediction accuracy.

5. Explore techniques for model interpretability to gain insights into the factors influencing energy consumption patterns.

6. Acquire knowledge and skills in machine learning-based time series forecasting for energy consumption, contributing to the advancement of sustainable energy practices.

Product Aims

The primary product of this project will be a machine learning model capable of accurately forecasting energy consumption. This model will be developed using advanced algorithms such as LSTM, SVR, and RFR, and will be trained and fine-tuned using the collected and pre-processed historical energy consumption data. The project will also produce an evaluation framework, utilizing appropriate metrics to assess the performance of the models.

Additionally, the project aims to provide insights into the strengths and limitations of the implemented models, allowing stakeholders to make informed decisions regarding their application. Experiments with additional input features, such as weather data and historical consumption patterns, will be conducted to enhance the accuracy of predictions. The project will also explore techniques for model interpretability, providing explanations for the factors influencing energy consumption patterns.

The final deliverable of the project will be a comprehensive report documenting the methodology, experimental results, analysis of the implemented models, and recommendations for future research. This report will serve as a valuable

Objectives

The objective of this project is to develop a machine learning-based model for accurate forecasting of energy consumption. By leveraging advanced algorithms such as Long Short-Term Memory (LSTM), Support Vector Regression (SVR), and Random Forest Regression (RFR), the project aims to analyze historical energy consumption data and make precise predictions for future energy usage. The project seeks to:

1. Conduct a comprehensive literature survey on Energy Consumption trends, statistical analysis techniques, and forecasting models.

2. Collect and pre-process relevant historical energy consumption data, ensuring data quality through cleaning, handling missing values, and potentially performing feature

engineering.

3. Train and fine-tune the selected machine learning models using the prepared dataset to optimize their performance for energy consumption forecasting.

4. Evaluate and compare the performance of different models using appropriate evaluation metrics such as mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R-squared).

5. Investigate the impact of integrating additional input features such as weather data, time of day, and historical energy consumption patterns on the accuracy of energy consumption predictions.

6. Explore techniques for model interpretability, providing insights into the factors influencing energy consumption patterns and enhancing the understanding of the driving forces behind energy demand.

7. Acquire knowledge and skills in machine learning-based time series forecasting for energy consumption, contributing to advancements in sustainable energy practices.

8. Produce a comprehensive report detailing the methodology, experimental results, analysis of the implemented models, and recommendations for future research.

Hardware Resources Required

1. Computing Resources: Machine learning algorithms, especially those based on deep learning models like LSTM, can be computationally intensive. Therefore, access to a sufficiently powerful computing infrastructure is necessary. This may include high-performance CPUs or GPUs

2. Internet access for data collection and research purposes.

Software Resources Required

1. Programming Language: The project will require a programming language suitable for implementing machine learning algorithms and handling data processing tasks. Python, with libraries such as NumPy, pandas, and scikit-learn, is recommended due to its extensive support for machine learning frameworks and data manipulation.

2. Machine Learning Libraries: The project will utilize machine learning libraries to implement and train the forecasting models. Essential libraries include scikit-learn for SVR and RFR, etc.

3. Data Visualization Tools: To analyze and visualize the historical energy consumption data, data visualization libraries such as Matplotlib and Seaborn can be used. These tools facilitate the exploration of patterns and trends within the data, aiding in the interpretation of results.

4. Integrated Development Environment (IDE): An IDE such as Jupyter Notebook or Google Collab can be employed for code development, experimentation, and documentation purposes. These environments provide a user-friendly interface for writing and executing code, as well as displaying visualizations and documenting project

progress.

5. Data Cleaning and Pre-processing Tools: Various data cleaning and pre-processing techniques will be applied to the collected energy consumption data. Libraries like Pandas provide functionalities for handling missing values, performing data transformations, and feature engineering.

6. Performance Evaluation Tools: To evaluate the performance of the machine learning models, metrics such as mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R-squared) will be employed. These metrics can be calculated using libraries like scikit-learn.

7. Documentation Tools: Documentation is essential for capturing the project's progress and findings. Tools such as Markdown or LaTeX can be used to write the project report, while productivity suites like Microsoft Office or Google Docs can be employed for organizing project-related documentation.

Project Deliverables

1. Project Proposal: A comprehensive document outlining the research objectives, methodology, and expected outcomes of the project titled "Forecasting Energy Consumption Using Machine Learning." The proposal will provide an overview of the project's scope, rationale, and significance.

2. Project Terms of Reference (TOR): A detailed document specifying the scope, responsibilities, and deliverables of the project. The TOR will define the project's boundaries, stakeholders, resources required, and project management procedures specific to the "Forecasting Energy Consumption Using Machine Learning" project.

3. Project Plan: A well-structured plan that includes a timeline, tasks, and milestones for the "Forecasting Energy Consumption Using Machine Learning" project. The plan will outline the sequential steps involved in collecting, processing, modeling, and evaluating energy consumption data, ensuring efficient project execution.

4. Literature Review: A summary of existing research on energy consumption forecasting using machine learning techniques. This review will provide a comprehensive understanding of the current state of the field, identify research gaps, and inform the selection of appropriate machine learning models for the project.

5. Data Collection and Pre-processing Report: A report detailing the process of collecting relevant historical energy consumption data and pre-processing it for analysis. This report will document the data sources, data cleaning techniques employed, handling of missing values, and any feature engineering performed.

6. Machine Learning Model Implementation: The implementation of machine learning models for energy consumption forecasting, including Long Short-Term Memory (LSTM), Support Vector Regression (SVR), Random Forest Regression (RFR), or other selected models. This deliverable will include the code, trained models, and any necessary utility functions or modules.

7. Model Evaluation and Performance Analysis: An evaluation report assessing the performance of the implemented machine learning models using appropriate evaluation metrics such as mean absolute error (MAE), root mean squared error (RMSE), and

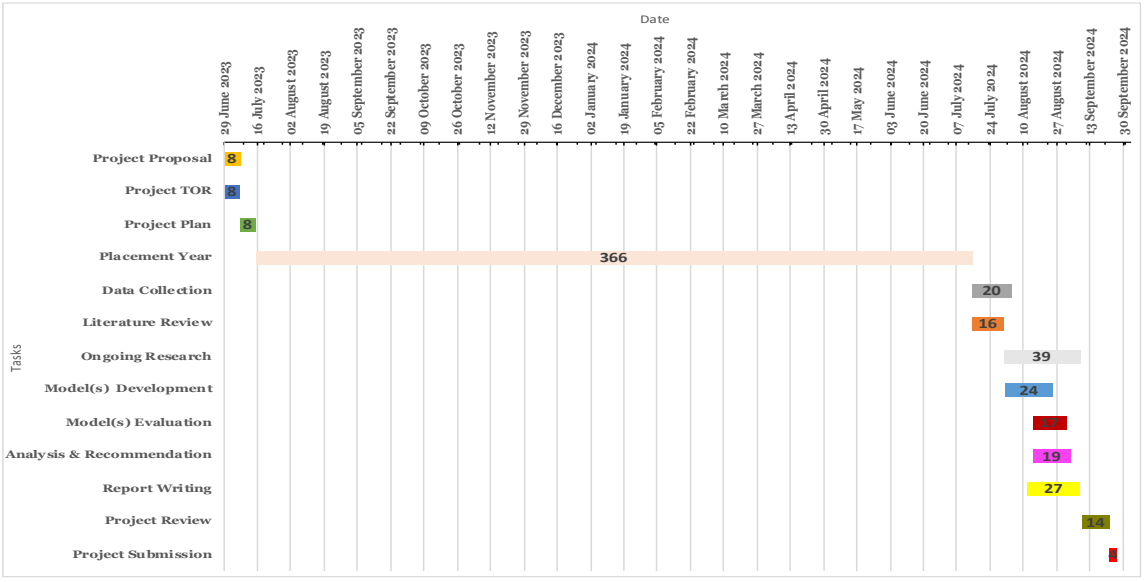
coefficient of determination (R-squared). The report will compare the results obtained from different models and analyze their strengths and limitations.

8. Integration of Additional Features and Experimentation: Documentation of experiments conducted to explore the integration of additional input features such as weather data, time of day, and historical energy consumption patterns. This deliverable will present the findings and insights gained from these experiments, demonstrating the impact on the accuracy of energy consumption predictions.

9. Final Project Report: A comprehensive report encompassing all project deliverables, including the project proposal, TOR, project plan, literature review, data collection and pre-processing report, machine learning model implementation, model evaluation and performance analysis, and the integration of additional features. The final report will discuss the implications of the findings, provide recommendations, and conclude the "Forecasting Energy Consumption Using Machine Learning" project.

Project Plan

Tasks	Start Date	Duration (Days)	End Date
Project Proposal	29/06/2023	8	07/07/2023
Project TOR	29/06/2023	8	07/07/2023
Project Plan	07/07/2023	8	15/07/2023
Placement Year	15/07/2023	366	15/07/2024
Data Collection	15/07/2024	20	04/08/2024
Literature Review	15/07/2024	16	31/07/2024
Ongoing Research	31/07/2024	39	08/09/2024
Model(s) Development	01/08/2024	24	25/08/2024
Model(s) Evaluation	15/08/2024	17	01/09/2024
Analysis & Recommendation	15/08/2024	19	03/09/2024
Report Writing	12/08/2024	27	08/09/2024
Project Review	09/09/2024	14	23/09/2024
Project Submission	23/09/2024	4	27/09/2024



References:

[1] Wang, Z., Wang, Y., Zeng, R., Srinivasan, R.S. and Ahrentzen, S., 2018. Random Forest based hourly building energy prediction. *Energy and Buildings*, *171*, pp.11-25.

[2] Cerqueira, V., Torgo, L. and Mozetič, I., 2020. Evaluating time series forecasting models: An empirical study on performance estimation methods. *Machine Learning*, *109*, pp.1997-2028.

[3] Hong, T., Pinson, P., Wang, Y., Weron, R., Yang, D. and Zareipour, H., 2020. Energy forecasting: A review and outlook. IEEE Open Access Journal of Power and Energy, 7, pp.376-388.

[4] Xiao, L., Wang, J., Dong, Y. and Wu, J., 2015. Combined forecasting models for wind energy forecasting: A case study in China. *Renewable and Sustainable Energy Reviews*, *44*, pp.271-288.

START HERE - Basic Information

This form must be completed for all student projects.

Before you proceed

Some activities inherently involve increased risks or approval by external regulatory bodies, so a proportional ethics review is not recommended and a full ethical review may be required.

These may include:

i. Approval from an external regulatory body (including, but not limited to: NHS (HRA), HMPPS etc.);
ii. Misleading participants;
iii. Research without the participants' consent;
iv. Clinical procedures with participants;
v. The ingestion or administration of any substance to participants by any means of delivery;
vi. The use of novel techniques, even where apparently non-invasive, whose safety may be open to question;
vii. The use of ionising radiation or exposure to radioactive materials;
viii. Engaging in, witnessing, or monitoring criminal activity;
ix. Engaging with, or accessing terrorism related materials;
x. A requirement for security clearance to access participants, data or materials;
xi. Physical or psychological risk to the participants or researcher;
xii. The project activity takes place in a country outside of the UK for which there is currently an active travel warning issued by the authorities (see info button);
xiii. Animals, animal tissue, new or existing human tissue, or biological toxins and agents;
xiv. The sharing of participant personal data with a third party, regardless of the form under which the data is presented.

If any of these activities are fundamental to your project, please contact your supervisor to determine if a full application is required.

This form must be completed for each research project which you undertake at the University. It must be approved by your supervisor (where relevant) PRIOR to the start of any data collection.

In completing this form, please consult the University's Research Ethics and Governance standards.

A1a Please confirm that you will abide by the University's Research Ethics and Governance standards in relation to this project.

- (•) Yes
- ( ) No

A1b Data Protection

The University is responsible for complying with the UK General Data Protection Regulation whenever personal data is processed. Under the Data Protection Policy, all staff and students have a responsibility to comply with the regulation in their day-to-day activities. The first step you can take to understand these responsibilities is to review the Data Protection in Reseach guidance pages and complete the University's Mandatory Data Protection Training. Student training is available through Moodle (in the 'Skills Online' section – please follow this link). To make sure your knowledge is up to date, all staff and students must complete the training every two years. If you have any issues in accessing the data protection training or have any questions about the training, please contact dataprotection@mmu.ac.uk.

Have you reviewed the Data Protection guidance pages and completed the Data Protection Training in the last two years?

- (●) Yes
- ( ) No

A2 Are you submitting this application as a learning experience, for a unit which already has ethical approval? (please confirm with your supervisor)

- ( ) Yes
- (●) No

A3 Student details

Title	First Name	Surname
	Anthony Anayo	Eze

Email: ANTHONY.A.EZE@stu.mmu.ac.uk

A3.1 Manchester Metropolitan University ID number

22536729

A4 Supervisor

Title	First Name	Surname
Dr	Anthony	Bukowski

Faculty: Science and Engineering

Telephone: n/a

Email: a.bukowski@mmu.ac.uk

A5 Which Faculty is responsible for the project?

Science and Engineering

A6 Course title

Data Science

A7 Project title

Forecasting Energy Consumption Using Machine learning

A8 What is the proposed start date of your project?

04/07/2023

A9 When do you expect to complete your project?

29/09/2023

A10 Please describe the overall aims of your project (3-4 sentences). Research questions should also be included here.

1. Develop a machine learning model tailored for accurate energy consumption forecasting.
2. Explore and compare the performance of advanced machine learning algorithms such as LSTM, SVR, and RFR for energy consumption prediction.
3. Evaluate and analyze the strengths and limitations of the selected models using appropriate evaluation metrics.
4. Investigate the impact of integrating additional input features, such as weather data, time of day, and historical energy consumption patterns, on prediction accuracy.
5. Explore techniques for model interpretability to gain insights into the factors influencing energy consumption patterns.
6. Acquire knowledge and skills in machine learning-based time series forecasting for energy consumption, contributing to the advancement of sustainable energy practices.

A11 Please describe the research activity

The objective of this project is to develop a machine learning-based model for accurate forecasting of energy consumption. By leveraging advanced algorithms such as Long Short-Term Memory (LSTM), Support Vector Regression (SVR), and Random Forest Regression (RFR), the project aims to analyse historical energy consumption data and make precise predictions for future energy usage. The project seeks to:
1. Conduct a comprehensive literature survey on Energy Consumption trends, statistical analysis techniques, and forecasting models.
2. Collect and pre-process relevant historical energy consumption data, ensuring data quality through cleaning, handling missing values, and potentially performing feature engineering.
3. Train and fine-tune the selected machine learning models using the prepared dataset to optimize their performance for energy consumption forecasting.
4. Evaluate and compare the performance of different models using appropriate evaluation metrics such as mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R-squared).
5. Investigate the impact of integrating additional input features such as weather data, time of day, and historical energy consumption patterns on the accuracy of energy consumption predictions.
6. Explore techniques for model interpretability, providing insights into the factors influencing energy consumption patterns and enhancing the understanding of the driving forces behind energy demand.
7. Acquire knowledge and skills in machine learning-based time series forecasting for energy consumption, contributing to advancements in sustainable energy practices.
8. Produce a comprehensive report detailing the methodology, experimental results, analysis of the implemented models, and recommendations for future research.

A12 Please provide details of the participants you intend to involve (please include information relating to the number involved and their demographics; the inclusion and exclusion criteria)

None

A13 Please upload your project protocol

Documents

Type	Document Name	File Name	Version Date	Version	Size
Project Protocol	22536729_Term_Of_Reference	22536729_Term_Of_Reference.pdf	10/07/2023	v1	242.1 KB

Project Activity

B1 Are there any Health and Safety risks to the researcher and/or participants?

- ○ Yes
- ◉ No

B2 Please select any of the following which apply to your project

- ☐ Aspects involving human participants (including, but not limited to interviews, questionnaires, images, artefacts and social media data)
- ☐ Aspects that the researcher or participants could find embarrassing or emotionally upsetting
- ☐ Aspects that include culturally sensitive issues (e.g. age, gender, ethnicity etc.)
- ☐ Aspects involving vulnerable groups (e.g. prisoners, pregnant women, children, elderly or disabled people, people experiencing mental health problems, victims of crime etc.), but does not require special approval from external bodies (NHS, security clearance, etc.)
- ☐ Project activity which will take place in a country outside of the UK
- ☑ None of the above

B2.4 Is this project being undertaken as part of a larger research study for which a Manchester Metropolitan application for ethical approval has already been granted or submitted?

- ○ Yes
- ◉ No

Data

F1 How and where will data and documentation be stored?

INTERNET, KAGGLE, GOOGLE SCHOLAR

F2 Will you be using personal data? Personal data is anything than can be used to identify a living individual, directly or indirectly. Pseudonymised data is still personal data.

- ○ Yes
- ◉ No

Insurance

F3 Does your project involve:

- ☐ Pregnant persons as participants with procedures other than blood samples being taken from them? (see info button)
- ☐ Children aged five or under with procedures other than blood samples being taken from them? (see info button)
- ☐ Activities being undertaken by the lead investigator or any other member of the study team in a country outside of the UK as indicated in the info button? If 'Yes', please refer to the 'Travel Insurance' guidance on the info button
- ☐ Working with Hepatitis, Human T-Cell Lymphotropic Virus Type iii (HTLV iii), or Lymphadenopathy Associated Virus (LAV) or the mutants, derivatives or variations thereof or Acquired Immune Deficiency Syndrome (AIDS) or any syndrome or condition of a similar kind?
- ☐ Working with Transmissible Spongiform Encephalopathy (TSE), Creutzfeldt-Jakob Disease (CJD), variant Creutzfeldt-Jakob Disease (vCJD) or new variant Creutzfeldt-Jakob Disease (nvCJD)?
- ☐ Working in hazardous areas or high risk countries? (see info button)
- ☐ Working with hazardous substances outside of a controlled environment?
- ☐ Working with persons with a history of violence, substance abuse or a criminal record?
- ☑ None of the above

Additional Information

G1 Do you have any additional information or comments which have not been covered in this form?

- ○ Yes
- ◉ No

G2 Do you have any additional documentation which you want to upload?

- ○ Yes
- ◉ No

Signatures

H1 I confirm that all information in this application is accurate and true. I will not start this project until I have received Ethical Approval.

- ◉ I confirm

H2 Please notify your supervisor that this application is complete and ready to be submitted by clicking "Request" below. Do not begin your project until you have received confirmation from your supervisor - it is your responsibility to ensure that they do this.

Signed: This form was signed by Anthony Bukowski (A.Bukowski@mmu.ac.uk) on 11/07/2023 10:21 AM

H3 Have you been instructed by your supervisor to request a second signature for this application?

- ○ Yes
- ◉ No

H4 By signing this application you are confirming that all details included in the form have been completed accurately and truthfully. You are also confirming that you will comply with all relevant UK data protection laws, and that that research data generated by the project will be securely archived in line with requirements specified by the University, unless specific legal, contractual, ethical or regulatory requirements apply.

Signed: This form was signed by Anthony Anayo Eze (ANTHONY.A.EZE@stu.mmu.ac.uk) on 10/07/2023 4:42 PM

Appendix B

Dataset

Available on request from Anthony Bukowski

However, Please find a the link to PJM Data Miner 2 (Data Source)

Appendix C

All Experimental code

Available on request from Anthony Bukowski

Also you can find all the experimental code in their respective notebooks by navigating to my github page (Notebooks)

Appendix D

Experimental results

Model Name	Parameter Settings	MAPE	MAE	RMSE	R2	Rank
LSTM Model 1	batch size = 1000, 100 epochs, Adam optimizer	4.64%	528.47	673.94	92.41%	1
Ensemble_XGB_RFR_GBR	Voting Regressor Hyperparameter tuned (XGB+RFR+GBR)	4.73%	544.69	692.93	91.97%	2
Hyperparameter Tuned XGB 2	n estimators: 100, 200 max depth: 3, 5, 7 learning rate: 0.01, 0.05, 0.1 subsample: 0.8, 1.0 colsample bytree: 0.8, 1.0 gamma: 0, 0.5, 1 reg lambda: 1, 1.5 reg alpha: 0, 0.5	4.84%	556.41	705.37	91.68%	3
Ensemble_XGB_GBR	Voting Regressor Hyperparameter tuned (XGB+GBR)	4.85%	556.74	703.42	91.73%	4
LSTM Model 2	batch size = 365, 150 epochs, Adam optimizer	4.90%	565.19	717.06	91.40%	5
Hyperparameter Tuned RFR 2	n estimators: 50, 100, 200 max depth: 10, 20, None min samples split: 2, 5, 7 min samples leaf: 1, 2, 3	4.90%	567.24	733.06	91.02%	6
Random Forest Regressor	None	4.92%	569.17	735.46	90.96%	7
Hyperparameter Tuned RFR 1	n estimators: 50, 100max depth: 10, 20, Nonemin samples split: 2, 5min samples leaf: 1, 2	4.92%	569.17	735.46	90.96%	8
Hyperparameter Tuned GBR 2	n estimators: 50, 100, 150 max depth: 3, 5, 7 learning rate: 0.05, 0.1 subsample: 0.8, 1.0 min samples split: 2, 5, 7	4.96%	568.65	718.42	91.37%	9

	min samples leaf: 1, 2, 3					
Hyperparameter Tuned LSTM 1	LSTM units: 50-150 Random Search dropout: 0.1-0.4 batch size: 128, epochs: 50	4.97%	576.64	727.57	91.15%	10
Hyperparameter Tuned MLP 1	units: 64, 128 Bayesian Optimization dropout: 0.2, 0.3 learning rate: 1e-4 to 1e-3	5.03%	574.58	733.88	90.99%	11
Hyperparameter Tuned GBR 1	n estimators: 50, 100, 150 max depth: 3, 5 learning rate: 0.05, 0.1 subsample: 0.8 min samples split: 2, 5 min samples leaf: 1, 2	5.09%	583.50	740.60	90.83%	12
Hyperparameter Tuned XGB 1	n estimators: 50, 100max depth: 3, 5learning rate: 0.05, 0.1subsample: 0.8colsample bytree: 0.8gamma: 0, 1	5.11%	587.03	750.42	90.58%	13
XGBoost Regressor	None	5.18%	593.49	755.54	90.46%	14
Hyperparameter Tuned MLP 2	batch size: 1000, epochs: 150 Bayesian Optimization	5.23%	614.00	789.72	89.57%	15
Hyperparameter Tuned LSTM 2	epochs: 100, batch size: tuned Random Search	5.56%	629.92	780.45	89.82%	16
Hyperparameter Tuned DTR	max depth: 5, 10, 15, 20 min samples split: 2, 10, 20, 25 min samples leaf: 1, 5, 10, 15 max features: None, sqrt, log2	5.79%	668.72	876.60	87.15%	17

MLP Model 1	batch size = 1000, 150 epochs, Adam optimizer	5.82%	673.31	854.94	87.78%	18
Gradient Boosting Regressor	None	5.94%	694.49	906.84	86.25%	19
MLP Model 2	batch size = 365, 150 epochs, Adam optimizer	6.13%	698.74	872.74	87.26%	20
Decision Tree Regressor	None	6.18%	712.03	936.51	85.34%	21
Hyperparameter Tuned CNN 1	Conv1D filters: 32-128 Random Searchkernel sizes: 2-5Dense units: 30-100dropout: 0.0-0.5	6.84%	785.91	1003.96	83.15%	22
CNN Model 2	batch size = 365, 150 epochs, Adam optimizer	7.11%	787.18	994.86	83.45%	23
Hyperparameter Tuned CNN 2	epochs: 100 Random Search	7.47%	850.62	1091.21	80.09%	24
CNN Model 1	batch size = 1000, 100 epochs, Adam optimizer	8.59%	964.80	1243.52	74.14%	25
Linear Regression	None	11.00%	1273.26	1627.63	55.71%	26
Hyperparameter Tuned LR 2	poly degree: 1, 2, 3 poly interaction only: True, False poly include bias: True, False linear fit intercept: True, False linear positive: True, False	11.00%	1273.26	1627.62	55.71%	27
Hyperparameter Tuned LR 1	fit intercept: True, False positive: True, False	11.00%	1273.26	1627.62	55.71%	28
Suppor Vector Regression	None	12.76%	1546.25	2100.75	26.21%	29