A COMPARISON OF MODELLING METHODS: ELECTRICITY DISTRIBUTION UNITED KINGDOM

# ABSTRACT

This study delves into the prediction of electricity distribution in the UK by harnessing the potential of machine learning models. With the aim of enhancing forecasting accuracy, various machine learning techniques are explored and evaluated in the context of electricity demand prediction. The research investigates the applicability of established statistical models, such as SARIMA and Prophet, as well as more advanced methods like gradient boosted trees including XGBoost and Linear Trees. Building on this foundation, the study advances to the utilization of recurrent neural networks, specifically Long Short-Term Memory (LSTM) models, known for their ability to capture temporal dependencies effectively. By systematically comparing these methodologies, the research contributes valuable insights into the most suitable machine learning approaches for accurately predicting electricity distribution patterns in the UK, thus facilitating better resource allocation and management within the energy sector.

# ACKNOWLEDGEMENT

Put yours in this place

# TABLE OF EQUATIONS

[Equation 1 - Mathematical equation for SARIMA 22](#_Toc143548423)

[Equation 2 - Linear Tree Equation 24](#_Toc143548424)

[Equation 3- Trend Component equation 25](#_Toc143548425)

[Equation 4 - Seasonality Component Equation 26](#_Toc143548426)

# TABLE OF FIGURES

[Figure 1 - number of columns and rows in the initial dataset 29](#_Toc143615860)

[Figure 2 - Sum of null values in the dataset 29](#_Toc143615861)

[Figure 3 - Number of rows with null values in the indicated columns 29](#_Toc143615862)

[Figure 4 -Plot of Transmission System Demand with holidays included 30](#_Toc143615863)

[Figure 5 - Investigation of weekly patterns in the dataset 31](#_Toc143615864)

[Figure 6 - histogram showing TSD distribution 31](#_Toc143615865)

[Figure 7 - Box Plot showing distribution of electricity consumption within hours 33](#_Toc143615866)

[Figure 8- Box Plot showing distribution of electricity consumption within month 33](#_Toc143615867)

[Figure 9 - Boxplot showing daily distribution of electricity consumption and holiday effect 33](#_Toc143615868)

[Figure 10 - Box Plot showing distribution of electricity consumption within years 34](#_Toc143615869)

[Figure 11 - SARIMA Baseline model 0 results 38](#_Toc143615870)

[Figure 12 - MAPE AND Runtime of SARIMA Model 0 38](#_Toc143615871)

[Figure 13 - SARIMA model 1 results 39](#_Toc143615872)

[Figure 14 - MAPE AND Runtime of SARIMA Model 1 39](#_Toc143615873)

[Figure 15 - SARIMA model 2 results 39](#_Toc143615874)

[Figure 16 - MAPE AND Runtime of SARIMA Model 2 39](#_Toc143615875)

[Figure 17 - Feature importance from XGBoost 40](#_Toc143615876)

[Figure 18 - Scatter plot comparing test set and predictions 40](#_Toc143615877)

[Figure 19 - XGBoost simple model prediction on test set for 2 weeks 41](#_Toc143615878)

[Figure 20 - MAPE AND RMSE of simple XGBoost Model 41](#_Toc143615879)

[Figure 21 - MAPE AND RMSE of best hyperparameters XGBoost model 43](#_Toc143615880)

[Figure 22- Future demand prediction using XGBoost best params model 43](#_Toc143615881)

[Figure 23- - MAPE AND RMSE of holiday prophet model 48](#_Toc143615882)

[Figure 24 - MAPE AND RMSE of best-parameters Prophet model 49](#_Toc143615883)

[Figure 25 - Future demand prediction using Prophet model 49](#_Toc143615884)

[Figure 26 - Comparison of future prediction using Prophet and XGBoost 50](#_Toc143615885)

[Figure 27 - Loss evaluation in LSTM Model 50](#_Toc143615886)

[Figure 28 - MAPE AND RMSE of LSTM model 51](#_Toc143615887)

[Figure 29 - MAPE AND RMSE of Deep LSTM model 52](#_Toc143615888)

[*Figure 30 - Summary table for all MAPE and RMSE of all models* 53](#_Toc143615889)

TABLE OF CONTENTS

[ABSTRACT 2](#_Toc143615890)

[ACKNOWLEDGEMENT 3](#_Toc143615891)

[TABLE OF EQUATIONS 4](#_Toc143615892)

[TABLE OF FIGURES 5](#_Toc143615893)

[INTRODUCTION 8](#_Toc143615894)

[1.1 BACKGROUND OF STUDY 8](#_Toc143615895)

[1.1.1 MACHINE LEARNING 8](#_Toc143615896)

[1.1.2 ELECTRICITY DISTRIBUTION 8](#_Toc143615897)

[1.2 PROBLEM STATEMENT 9](#_Toc143615898)

[1.3 AIM AND OBJECTIVES 10](#_Toc143615899)

[1.4 SIGNIFICANCE OF STUDY 11](#_Toc143615900)

[1.5 JUSTIFICATION OF STUDY 12](#_Toc143615901)

[1.6 LIMITATIONS OF STUDY 14](#_Toc143615902)

[LITERATURE REVIEW 16](#_Toc143615903)

[2.1 REVIEW OF RELEVANT PREVIOUS RESEARCH ON ELECTRICITY DISTRIBUTION 16](#_Toc143615904)

[2.2 DATA PRIVACY 17](#_Toc143615905)

[METHODOLOGY 19](#_Toc143615906)

[3.1 OVERVIEW 19](#_Toc143615907)

[3.2 EXPLANATION OF THE MODELS TO BE USED IN THE RESEARCH 21](#_Toc143615908)

[3.2.1 SARIMA 21](#_Toc143615909)

[3.2.2 XGBOOST 23](#_Toc143615910)

[3.2.3 LINEAR TREES 24](#_Toc143615911)

[3.3.4 PROPHET 25](#_Toc143615912)

[3.2.5 LSTM AND DEEP LSTM 27](#_Toc143615913)

[3.3 DATA PREPROCESSING 28](#_Toc143615914)

[3.4 DATA CLEANING AND UNDERSTANDING 29](#_Toc143615915)

[3.5 DATA PREPARATION 29](#_Toc143615916)

[3.7 GAINING INSIGHT INTO FEATURES 32](#_Toc143615917)

[3.8 PREDICTION WITH MODELS 34](#_Toc143615918)

[3.8.1 SARIMA 34](#_Toc143615919)

[3.8.2 XGBOOST 35](#_Toc143615920)

[3.8.3 LINEAR TREES 36](#_Toc143615921)

[3.8.4 PROPHET 36](#_Toc143615922)

[3.8.5 LSTM AND DEEP LSTM 36](#_Toc143615923)

[RESULTS 38](#_Toc143615924)

[4.1 PRESENTATION OF RESULTS FROM RESEARCH MODELS 38](#_Toc143615925)

[4.1.1 SARIMA 38](#_Toc143615926)

[4.1.2 XGBOOST 39](#_Toc143615927)

[4.1.3 LINEAR BOOST MODEL 44](#_Toc143615928)

[4.1.4 PROPHET 44](#_Toc143615929)

[4.1.5 LSTM 50](#_Toc143615930)

[4.1.6 DEEP LSTM 51](#_Toc143615931)

[4.2 DISCUSSION OF RESULTS 53](#_Toc143615932)

[CONCLUSION 55](#_Toc143615933)

[5.1 CONCLUSION FROM DISCUSSION OF RESULT 55](#_Toc143615934)

[5.2 OBSERVATIONS FROM THE RESEARCH 55](#_Toc143615935)

[5.3 FUTURE WORKS AND RECOMMENDATIONS 56](#_Toc143615936)

[REFERENCES 57](#_Toc143615937)

[BIBLIOGRAPHY 60](#_Toc143615938)

# INTRODUCTION

## 1.1 BACKGROUND OF STUDY

### 1.1.1 MACHINE LEARNING

Machine learning, a branch of artificial intelligence, focuses on the development of algorithms and statistical models that enable computers to learn autonomously and make predictions or decisions based on data without explicit programming. This approach empowers computers to continuously improve and make intelligent choices or provide valuable insights from intricate and extensive datasets. At its core, machine learning involves training a model using a set of data examples referred to as training data. This data comprises input features and corresponding output labels or target variables. The model then examines the patterns and relationships within the data to construct a generalized representation that can be applied to new, unseen data for prediction or classification.

Machine learning encompasses various algorithms, including supervised learning, unsupervised learning, and reinforcement learning. Supervised learning involves training the model with labelled data, providing both input features and correct output labels. Unsupervised learning deals with unlabelled data, seeking to discover inherent patterns or structures within the dataset. Reinforcement learning trains an agent to interact with an environment and learn optimal actions through a reward and punishment system.

In summary, machine learning empowers computers to learn and enhance their performance through data, making it a powerful tool for automating tasks, making accurate predictions, and gaining valuable insights in diverse domains.

### 1.1.2 ELECTRICITY DISTRIBUTION

UK electricity distribution involves the efficient transmission of electrical power from generation sources to end consumers throughout the United Kingdom. It encompasses a network of infrastructure, including power lines, transformers, substations, and distribution networks, which ensure a reliable supply to homes, businesses, and other entities. The distribution system in the UK is divided into regional networks managed by Distribution Network Operators (DNOs). These operators are responsible for maintaining, operating, and developing the distribution infrastructure within their designated regions. They collaborate closely with electricity suppliers, generators, and stakeholders to guarantee a smooth and uninterrupted flow of electricity to end-users. Balancing supply and demand, managing voltage levels, and promptly addressing faults or emergencies are essential roles performed by the distribution network to maintain a stable and secure electricity supply. Furthermore, the network enables the integration of diverse energy sources, including renewables, into the grid, supporting the nation's transition to a low-carbon energy system. Regulatory bodies like the Office of Gas and Electricity Markets (Ofgem) oversee and regulate the UK electricity distribution sector. Ofgem establishes price controls, monitors performance, and ensures that DNOs fulfil their obligations in terms of reliability, safety, and customer service.

The UK electricity distribution system continually evolves to align with the changing energy landscape. Emphasizing smart grids, digital technologies, and demand-side management, these advancements aim to enhance efficiency, promote the integration of renewable energy sources, and empower consumers to actively manage their electricity consumption. In summary, UK electricity distribution is a critical element of the nation's energy infrastructure, prioritizing the safe, reliable, and efficient delivery of electricity to consumers while supporting the country's goals for energy transition.

## 1.2 PROBLEM STATEMENT

Accurate prediction of electricity consumption in the United Kingdom is crucial for effective energy planning, resource allocation, and sustainable development. It enables decision-making by power grid operators, policymakers, and energy companies in areas such as generation capacity, infrastructure investments, and environmental impact mitigation. However, the selection and evaluation of machine learning algorithms for electricity consumption prediction in the UK present significant challenges. One primary issue is the lack of comprehensive research that compares the effectiveness of different machine learning algorithms specifically tailored to the unique characteristics of the UK electricity market. While machine learning has demonstrated promise in forecasting various domains, a systematic and rigorous comparison of algorithms specific to the UK context is needed. Selecting appropriate algorithms is essential to ensure accurate and reliable predictions, which are vital for efficient energy management and planning.

Furthermore, the dynamic nature of the UK electricity market poses another challenge. The industry undergoes constant changes driven by factors such as evolving energy policies, advancements in renewable energy technologies, and shifts in consumer behavior. These changes introduce new patterns and complexities that machine learning algorithms must capture. Hence, there is a need to evaluate the adaptability and robustness of different algorithms in capturing the dynamic and variable electricity consumption patterns in the UK.

In addition, incorporating relevant external factors that influence electricity consumption, such as weather conditions, socio-economic indicators, and policy interventions, further complicates the prediction task. Accurately including and modeling these factors can significantly enhance forecasting accuracy. Therefore, it is necessary to explore and identify the most effective machine learning algorithms capable of handling and utilizing these additional data sources to improve prediction accuracy.

Addressing these challenges is vital to provide reliable and accurate forecasts of electricity consumption for decision-makers in the UK energy sector. By comparing the performance of various machine learning algorithms while considering the dynamic nature of the electricity market and external factors, this research aims to contribute to the development of effective predictive models. Ultimately, this will support energy planners, policymakers, and energy companies in making informed decisions, optimizing resource allocation, and facilitating the transition to a sustainable and resilient energy system in the United Kingdom.

## AIM AND OBJECTIVES

The aim of this project is to compare various models and provide a suggestion as to which is the most suitable for electricity distribution in the United Kingdom.

The objectives of this project are as follows:

* To compare the performance of various machine learning algorithms in predicting electricity consumption in the United Kingdom.
* To identify the most accurate and reliable machine learning algorithm for electricity consumption prediction in the UK context.
* To evaluate the adaptability and robustness of different machine learning algorithms in capturing the dynamic and variable nature of electricity consumption patterns in the UK.
* To analyse the impact of evolving energy policies, advancements in renewable energy technologies, and shifts in consumer behaviour on electricity consumption patterns and the performance of machine learning algorithms.
* To explore the potential of machine learning algorithms in enhancing energy planning, resource allocation, and sustainable development in the UK.
* To provide insights into the strengths and limitations of different machine learning algorithms for electricity consumption prediction, enabling informed algorithm selection in practical applications.
* To contribute to the development of improved predictive models for electricity consumption forecasting in the UK, facilitating efficient energy management and planning.
* To contribute to the transition towards a sustainable and resilient energy system in the United Kingdom by enabling better understanding and prediction of electricity consumption patterns and facilitating the integration of renewable energy sources into the grid.

## 1.4 SIGNIFICANCE OF STUDY

The comparative study on machine learning algorithms for predicting electricity consumption in the United Kingdom is of significant importance for several compelling reasons; First and foremost, accurate prediction of electricity consumption is vital for efficient energy planning and optimal resource allocation. The findings from this study will offer valuable insights into the performance of different machine learning algorithms, enabling decision-makers in the energy sector to select the most effective algorithm for forecasting electricity demand. This will result in improved resource utilization, better planning of generation capacity, and enhanced energy management, ultimately leading to cost savings and the establishment of a more sustainable energy system. Secondly, the study will contribute to the development of enhanced predictive models tailored to the unique characteristics of the UK electricity market. By considering factors such as evolving energy policies, advancements in renewable energy technologies, and shifts in consumer behaviour, these models can effectively capture the dynamic nature of electricity consumption patterns. Consequently, decision-making in the face of evolving energy landscapes can be improved, and the accuracy and reliability of electricity consumption forecasts can be enhanced. Moreover, the integration of external factors such as weather conditions, socio-economic indicators, and policy interventions within the prediction models can provide a comprehensive understanding of the drivers influencing electricity consumption. Leveraging this knowledge can optimize energy planning, inform strategies for demand-side management, and facilitate the transition to a low-carbon energy system. The study will explore the potential of machine learning algorithms to effectively incorporate these factors, leading to more precise predictions and enabling evidence-based policy formulation. Furthermore, the research aims to address the existing gap in comprehensive studies that specifically compare machine learning algorithms for electricity consumption prediction in the UK context. Through a systematic evaluation, the study will contribute to the identification of the most accurate and reliable algorithms for this specific domain, empowering practitioners to make informed choices when implementing prediction models. Lastly, the significance of this study lies in its potential to support the achievement of sustainable development goals. Accurate electricity consumption prediction plays a critical role in optimizing energy usage, reducing greenhouse gas emissions, and facilitating the integration of renewable energy sources. By providing reliable forecasts, the study can foster the creation of a more sustainable and resilient energy system in the United Kingdom.

In summary, this study's significance stems from its ability to improve energy planning, resource allocation, and sustainable development through precise electricity consumption prediction. By comparing machine learning algorithms, considering the dynamic nature of the UK electricity market, and incorporating relevant external factors, the research will offer valuable insights to decision-makers, facilitate the transition to a low-carbon energy system, and contribute to the achievement of sustainable development goals.

## 1.5 JUSTIFICATION OF STUDY

The justification for conducting the study on comparing machine learning algorithms for predicting electricity consumption in the United Kingdom is based on several key factors and benefits it offers.

* **Addressing Research Gap:** Currently, there is a lack of comprehensive research specifically comparing the performance of machine learning algorithms for electricity consumption prediction in the UK. This study aims to fill this gap by conducting a systematic and rigorous evaluation of different algorithms. The findings will contribute valuable insights to the existing knowledge base, thereby addressing the research void in this specific area.
* **Improving Energy Planning and Resource Allocation:** Accurate prediction of electricity consumption is essential for efficient energy planning and optimal resource allocation. By comparing the performance of various machine learning algorithms, decision-makers in the energy sector will gain evidence-based information to select the most effective algorithm for forecasting electricity demand. This will result in improved decision-making, optimal resource allocation, and enhanced energy management, leading to cost savings and increased operational efficiency.
* **Facilitating Sustainable Development:** Precise prediction of electricity consumption is vital for promoting sustainable development. Through more accurate forecasting, this study will support the integration of renewable energy sources and facilitate the transition to a low-carbon energy system. This contribution will help reduce greenhouse gas emissions, enhance energy efficiency, and align with the sustainability goals established by the UK government and international agreements like the Paris Agreement.
* **Informing Policy Formulation:** The study will incorporate external factors, such as weather conditions, socio-economic indicators, and policy interventions, into the prediction models. By doing so, it will provide valuable insights into the drivers of electricity consumption and support evidence-based policy formulation. Decision-makers can leverage these findings to develop effective energy policies, design strategies for demand-side management, and promote energy conservation initiatives, ultimately fostering a sustainable energy landscape.
* **Supporting Infrastructure Investments:** Accurate prediction of electricity consumption is crucial for making well-informed decisions regarding infrastructure investments. By providing reliable forecasts, this study will assist in determining optimal generation capacity, grid expansions, and infrastructure upgrades. Consequently, it will prevent overinvestment or underinvestment in the electricity infrastructure, leading to cost-efficient and reliable energy supply that can meet the growing demands of consumers.
* **Advancing Machine Learning Applications in the Energy Sector:** The study will contribute to the advancement of machine learning applications in the energy sector. Through evaluating the performance of different algorithms, it will enhance our understanding of their strengths and limitations in predicting electricity consumption. This knowledge can be applied to other energy domains, opening up possibilities for broader adoption of machine learning techniques to improve forecasting accuracy and support decision-making.

In summary, the proposed study on comparing machine learning algorithms for predicting electricity consumption in the United Kingdom is justified by its ability to fill the existing research gap, enhance energy planning, and resource allocation, facilitate sustainable development, inform policy formulation, support infrastructure investments, and advance machine learning applications in the energy sector. The outcomes of the study will provide valuable insights to decision-makers, researchers, and practitioners, ultimately contributing to the efficient and sustainable management of electricity consumption in the UK.

## 1.6 LIMITATIONS OF STUDY

While the proposed study on comparing machine learning algorithms for predicting electricity consumption in the United Kingdom aims to provide valuable insights and contributions, it is important to acknowledge and address certain limitations that may affect the interpretation and generalizability of the findings.

* **Data Availability and Quality:** The study's effectiveness heavily relies on the availability and quality of the electricity consumption data. Limited access to comprehensive and accurate historical data may restrict the scope and depth of the analysis. Inaccuracies or missing data points could introduce bias or affect the performance evaluation of the machine learning algorithms. The study should address these limitations by carefully selecting representative and reliable datasets and ensuring data quality checks and preprocessing techniques are applied appropriately.
* **Algorithm Selection:** The study will compare a selection of machine learning algorithms; however, it may not be possible to include all possible algorithms due to resource and time constraints. The chosen algorithms may not encompass the entire spectrum of available techniques, potentially omitting some algorithms that could yield superior performance. Additionally, the study's focus on specific machine learning algorithms may limit the exploration of emerging or alternative approaches that could have relevance in the electricity consumption prediction domain.
* **Generalizability:** The findings of the study may be specific to the context of the United Kingdom and may not be directly applicable to other regions or countries with different electricity market characteristics, regulatory frameworks, or consumer behaviour patterns. Therefore, caution must be exercised when extrapolating the results beyond the study's scope.
* **External Factors and Variables:** The study aims to incorporate external factors such as weather conditions and socio-economic indicators into the prediction models. However, there may be limitations in the availability, accuracy, or completeness of these external datasets. Furthermore, the choice and representation of these variables may introduce additional uncertainties and biases. Careful consideration and robust validation of these external factors are necessary to ensure their effective integration into the models.
* **Interpretability of Results:** Machine learning algorithms often operate as "black boxes," making it challenging to interpret the reasoning behind their predictions. While the study may focus on evaluating the algorithms based on their accuracy and performance metrics, the interpretability and explainability of the algorithms' results may pose a limitation. It is important to balance model complexity with interpretability to ensure the findings can be effectively communicated and understood by stakeholders and decision-makers.
* **Dynamic Nature of Energy Market:** The energy market, including electricity consumption patterns, is subject to constant changes influenced by factors such as policy changes, technological advancements, and socio-economic shifts. The study's findings may be constrained by the dataset's time period, potentially missing recent trends or capturing limited variations. Incorporating real-time or near real-time data could provide more accurate and up-to-date insights but may introduce additional complexities and challenges.

Despite these limitations, it is crucial to recognize the significance of the study's contributions and the potential benefits it offers in terms of informing decision-making, enhancing energy planning, and facilitating sustainable development in the United Kingdom. Awareness of these limitations will help guide the study's design, analysis, and interpretation, ensuring transparency and a clear understanding of the boundaries within which the findings can be applied.

# LITERATURE REVIEW

## 2.1 REVIEW OF RELEVANT PREVIOUS RESEARCH ON ELECTRICITY DISTRIBUTION

Effective energy management relies on accurate prediction of electricity consumption, allowing utilities to optimize resource allocation and plan for future energy demands. In recent years, machine learning algorithms have emerged as powerful tools for analyzing complex data patterns and making accurate predictions. This literature review provides an overview of existing research on the application of machine learning algorithms in predicting electricity consumption, with a specific focus on the United Kingdom. Smith et al. (2017) compared multiple machines learning algorithms, including Support Vector Regression (SVR), Random Forest (RF), and Artificial Neural Networks (ANN), for electricity consumption prediction in the UK. They found that SVR outperformed other algorithms in terms of accuracy and robustness. In a study by Johnson and Thompson (2018), they explored the effectiveness of Long Short-Term Memory (LSTM) networks for electricity demand forecasting. LSTM networks are a type of recurrent neural network that can capture long-term dependencies in time series data. The results showed that LSTM achieved superior performance compared to traditional statistical methods. Chen et al. (2018) employed LSTM to forecast hourly electricity demand in the UK, demonstrating superior results compared to traditional statistical models . Wang et al. (2018) demonstrated the effectiveness of SARIMA in forecasting hourly electricity demand in the UK, showcasing its ability to capture seasonality and long-term trends Li et al. (2019) investigated the application of Deep Belief Networks (DBN) in electricity consumption prediction. DBNs are a type of generative neural network that can learn complex hierarchical representations of data. Their findings demonstrated that DBNs provided accurate predictions, especially when dealing with large-scale datasets. The research used XGBoost to forecast daily electricity consumption in the UK, achieving superior prediction accuracy compared to traditional statistical models. A study conducted by Patel et al. (2021) employed the Gradient Boosting Machine (GBM) algorithm for electricity consumption prediction. GBM is an ensemble learning technique that combines multiple weak models to form a stronger predictor. The results indicated that GBM exhibited excellent predictive capabilities and outperformed other algorithms such as Decision Trees and Random Forests. Wang et al. (2022) proposed the use of a hybrid model combining Genetic Programming (GP) and Support Vector Regression (SVR) for electricity consumption prediction. Their hybrid model achieved higher prediction accuracy compared to individual algorithms alone, suggesting the synergy between GP and SVR. Wang et al. (2018) demonstrated the effectiveness of SARIMA in forecasting hourly electricity demand in the UK, showcasing its ability to capture seasonality and long-term trends. Ahn et al. (2020) utilized linear trees to forecast hourly electricity consumption in the UK, demonstrating its ability to handle large-scale data and provide interpretable predictions. Tugce and Salih (2021) utilized Prophet to predict daily electricity consumption in the UK, showcasing its ability to capture complex patterns and outperform traditional models.

## 2.2 DATA PRIVACY

In the domain of electricity distribution in the United Kingdom, strict adherence to data protection regulations is paramount when handling customer information. In 2018, the UK revised the Data Protection Act of 1998 through the enactment of the General Data Protection Regulation (GDPR). This regulation underscores the importance of processing customer data in a manner that is both reasonable and lawful, while also maintaining transparency.

To fulfil these requirements, entities responsible for collecting customer data, referred to as data controllers, must obtain explicit consent from the individuals, known as data subjects. Within this framework, the data amassed from these subjects must be securely managed and appropriately discarded once its utility expires. As part of an ongoing project, a retail company adheres to these stipulations, leveraging customer data related to their electricity consumption patterns for analytical purposes.

The integration of machine learning within diverse sectors often sparks discussions about inherent biases and ethical considerations. This topic frequently garners attention in media coverage. However, the benefits and efficiencies of machine learning, specifically in the context of electricity distribution, remain less familiar to the general populace.

Educating the public about the advantages and disadvantages of employing machine learning in this domain could elucidate that the benefits outweigh any potential drawbacks. To safeguard sensitive data and strategic information from being exploited by competitors, the project refrains from disclosing the retailer's identity. Although the company's operations and offerings are elucidated, its name is deliberately withheld.

Unforeseen incidents, such as revenue losses and disruptions in cash flow, can materialize abruptly within the electricity distribution sector. These occurrences might arise due to factors like unexpected fluctuations in energy demand. The resultant consequences could negatively impact the company's projected earnings, or lead to avoidable expenditures stemming from suboptimal operational decisions.

Legally, regulations prohibit the promotion of unregulated energy products by sales personnel. Consistency in pricing is crucial, barring instances of authorized discounts or adjustments necessitated by product anomalies. Deviating from this practice could be construed as discriminatory pricing. Consequently, alterations in product pricing mandate corresponding adjustments across all distribution points.

# METHODOLOGY

## 3.1 OVERVIEW

The entity responsible for managing the electricity system in Great Britain is the National Grid ESO. Commencing in 2009, they have compiled data on the electricity consumption within the region. This dataset receives bi-hourly updates, resulting in 48 data points per day. The characteristics of this dataset render it highly suitable for conducting time series forecasting activities. The columns provided by the dataset from NationalGrid are:

**SETTLEMENT\_DATE:** This column represents the date of the settlement period for which the electricity demand data is recorded.

**SETTLEMENT\_PERIOD:** It indicates the specific settlement period within a day for which the electricity demand data is recorded. Settlement periods are typically 30 minutes long, and they are used in electricity markets for settlement and pricing purposes.

**ND (National Demand)**: National Demand is the sum of metered generation, but excludes generation required to meet station load, pump storage pumping and interconnector exports. National Demand is calculated as a sum of generation based on National Grid ESO operational generation metering. Measured in MW.

**TSD (Transmission System Demand)**: Transmission System Demand is equal to the ND plus the additional generation required to meet station load, pump storage pumping and interconnector exports. Measured in MW.

**ENGLAND\_WALES\_DEMAND**: This column represents the electricity demand in the regions of England and Wales. It indicates the amount of electricity consumed during the specific settlement period. Measured in MW.

**EMBEDDED\_WIND\_GENERATION**: It represents the amount of electricity generated by wind power within the regions of England and Wales during the specific settlement period. This is an estimate of the GB wind generation from wind farms which do not have Transmission System metering installed. These wind farms are embedded in the distribution network and invisible to National Grid ESO. Their effect is to suppress the electricity demand during periods of high wind. The true output of these generators is not known so an estimate is provided based on National Grid ESO’s best model. Measured in MW.

**EMBEDDED\_WIND\_CAPACITY:** This is National Grid ESO’s best view of the installed embedded wind capacity in GB. This is based on publicly available information compiled from a variety of sources and is not the definitive view. It is consistent with the generation estimate provided above. Measured in MW.

**EMBEDDED\_SOLAR\_GENERATION**: It represents the amount of electricity generated by solar power within the regions of England and Wales during the specific settlement period. This is an estimate of the GB solar generation from PV panels. These are embedded in the distribution network and invisible to National Grid ESO. Their effect is to suppress the electricity demand during periods of high radiation. The true output of these generators is not known so an estimate is provided based on National Grid ESO’s best model. Measured in MW.

**EMBEDDED\_SOLAR\_CAPACITY**: As embedded wind capacity above, but for solar generation. Measured in MW.

**NON\_BM\_STOR**: It indicates the amount of electricity stored in non-BM (non-Balancing Mechanism Short-Term Operating Reserve) storage systems during the specific settlement period. For units that are not included in the ND generator definition. This can be in the form of generation or demand reduction. Measured in MW.

**PUMP\_STORAGE\_PUMPING**: This column likely represents the amount of electricity consumed during the specific settlement period for pumping water into a pumped storage hydropower system. Pumped storage is a method of energy storage that uses excess electricity to pump water to a higher elevation, which can later be released to generate electricity during peak demand periods. The demand due to pumping at hydro pump storage units; the -ve signifies pumping load.

**IFA\_FLOW**: It represents the flow of electricity through the Interconnector with France (IFA) during the specific settlement period. Interconnectors are transmission links between different countries or regions that allow the exchange of electricity. The flow on the respective interconnector. -ve signifies export power out from GB; +ve signifies import power into GB. Measured in MW.

**IFA2\_FLOW**: Similarly, to 'IFA\_FLOW', this column likely represents the flow of electricity through the Interconnector with France 2 (IFA2) during the specific settlement period. IFA2 is another interconnector between the UK and France. The flow on the respective interconnector. -ve signifies export power out from GB; +ve signifies import power into GB. Measured in MW.

**BRITNED\_FLOW**: It represents the flow of electricity through the BritNed interconnector, which connects the UK and the Netherlands, during the specific settlement period.

**MOYLE\_FLOW (Moyle Interconnector Flow)**: This column likely represents the flow of electricity through the Moyle interconnector, which connects Northern Ireland and Scotland, during the specific settlement period. The flow on the respective interconnector. -ve signifies export power out from GB; +ve signifies import power into GB. Measured in MW.

**EAST\_WEST\_FLOW (East West Interconnector Flow)**: It represents the flow of electricity between eastern and western parts of the UK during the specific settlement period. The flow on the respective interconnector. -ve signifies export power out from GB; +ve signifies import power into GB. Measured in MW.

**NEMO\_FLOW (Nemo Interconnector Flow)**: This column likely represents the flow of electricity through the Nemo Link interconnector, which connects the UK and Belgium, during the specific settlement period. The flow on the respective interconnector. -ve signifies export power out from GB; +ve signifies import power into GB. Measured in MW.

**NSL\_FLOW (North Sea Link Interconnector Flow)**: It represents the flow of electricity through the North Sea Link interconnector, which connects the UK and Norway, during the specific settlement period.

**ELECLINK\_FLOW**: This column likely represents the flow of electricity through the ElecLink interconnector, which connects the UK and France, during the specific settlement period.

The most important columns to this project were SETTLEMENT\_DATA, SETTLEMENT\_PERIOD AND TSD (TRANSMISSION SYSTEM DEMAND). TSD is the target variable I aimed to predict from the dataset. The aim of the project in clearly defined terms is to predict the future demand of electricity in the United Kingdom using different models.

## 3.2 EXPLANATION OF THE MODELS TO BE USED IN THE RESEARCH

For my project, I used the following 5 machine models based on their previous effectiveness on other research related to time series analysis

### 3.2.1 SARIMA

SARIMA (Seasonal Autoregressive Integrated Moving Average) is a time series forecasting model that extends the ARIMA model to handle seasonal patterns in the data. The mathematical formulation of SARIMA involves the following components:

**Autoregressive (AR) Component:** The AR component captures the linear relationship between the current observation and a lagged set of observations. It represents the autoregressive nature of the time series. The AR component is denoted by the parameter p, which represents the number of lagged observations to include in the model.

**Integrated (I) Component:** The I component incorporates differencing to make the time series stationary. Stationarity refers to the property where the statistical properties of the series, such as mean and variance, do not change over time. Differencing is performed by taking the difference between consecutive observations. The differencing order is denoted by the parameter d.

**Moving Average (MA) Component**: The MA component captures the dependency between the current observation and a lagged set of error terms. It represents the moving average nature of the time series. The MA component is denoted by the parameter q, which represents the number of lagged error terms to include in the model.

**Seasonal Components:** In addition to the above three components, SARIMA models also incorporate seasonal patterns in the data. Seasonality refers to the recurring patterns that occur at regular intervals, such as daily, monthly, or yearly. The seasonal components include seasonal autoregressive (SAR) terms, seasonal differencing (SD) terms, and seasonal moving average (SMA) terms. The seasonal orders are denoted by P, D, and Q, respectively.

Mathematically, a SARIMA(p, d, q)(P, D, Q)m model can be represented as:

|  |
| --- |
|  |

Equation 1 - Mathematical equation for SARIMA

where:

* yₜ represents the observed time series data
* L is the lag operator
* φ₁, φ₂, ..., φₚ and Φ₁, Φ₂, ..., Φₚ represent the autoregressive and seasonal autoregressive coefficients, respectively
* θ₁, θ₂, ..., θₚ and Θ₁, Θ₂, ..., Θₚ represent the moving average and seasonal moving average coefficients, respectively
* εₜ represents the error term
* p, d, q represents the non-seasonal orders
* P, D, Q represent the seasonal orders
* m represents the number of observations in a seasonal cycle

The SARIMA model aims to estimate the coefficients (φ's, θ's, Φ's, Θ's) that best fit the observed data. These coefficients are estimated using various statistical methods, such as maximum likelihood estimation or least squares estimation, to minimize the difference between the model's predicted values and the actual observations.

By incorporating these mathematical components and estimating the coefficients, SARIMA models provide a framework for forecasting time series data, capturing both the non-seasonal and seasonal patterns present in the data.

### 3.2.2 XGBOOST

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm that belongs to the family of gradient boosting methods. It is widely used for regression and classification tasks. The mathematical formulation of XGBoost involves the following components:

**Objective Function:** XGBoost minimizes an objective function that represents the loss or error of the model on the training data. The objective function consists of two main components: a loss function and a regularization term. The loss function quantifies the difference between the predicted values and the actual target values. Common loss functions include mean squared error for regression and log loss (binary or multinomial) for classification tasks. The regularization term helps to control the complexity of the model and prevent overfitting by penalizing large coefficient values.

**Boosting Algorithm:** XGBoost employs a boosting algorithm that combines weak learners (decision trees) to create a strong predictive model. It builds the model in an iterative manner, where each subsequent weak learner is trained to correct the mistakes made by the previous learners. The algorithm optimizes the objective function by minimizing the loss and regularization terms.

**Gradient Descent Optimization:** XGBoost uses gradient descent optimization to iteratively update the model parameters. It calculates the gradients of the objective function with respect to the model predictions and adjusts the parameters in the direction of steepest descent to minimize the loss. This process is repeated for a specified number of iterations (boosting rounds) until the model converges or a stopping criterion is met.

**Tree Ensembles:** XGBoost models are composed of an ensemble of decision trees. Each decision tree is a weak learner that predicts the target variable based on a set of input features. The algorithm constructs decision trees sequentially, with each subsequent tree trying to minimize the residual errors of the previous trees. The predictions of all trees are summed together to obtain the final output.

**Regularization Techniques:** XGBoost incorporates various regularization techniques to control model complexity and prevent overfitting. These techniques include shrinkage (learning rate) which reduces the impact of each individual tree on the final predictions, maximum depth limits to restrict the depth of each decision tree, and minimum child weight to impose constraints on the minimum number of training instances required to create a new tree node.

**Feature Importance:** XGBoost provides a measure of feature importance based on how much each feature contributes to reducing the loss function. This information can be useful for feature selection and understanding the relative importance of different features in making predictions.

By combining these mathematical components and optimization techniques, XGBoost builds an ensemble of decision trees that collectively form a highly accurate and robust predictive model. It iteratively minimizes the objective function, using gradient descent to optimize the model parameters, and incorporates regularization techniques to control overfitting, resulting in improved predictive performance.

3.2.3 LINEAR TREES  
Linear trees, also known as single-layer trees or decision stumps, are a simplified version of decision trees that consist of only a single node and two leaf nodes. Unlike traditional decision trees that have multiple levels of nodes, linear trees make predictions based on a linear combination of the input features.

Mathematically, a linear tree can be represented as:

|  |
| --- |
|  |

Equation 2 - Linear Tree Equation

where:

* y represents the predicted output or target variable
* β₀, β₁, β₂, ..., βᵣ are the coefficients or weights associated with the input features (x₁, x₂, ..., xᵣ)
* x₁, x₂, ..., xᵣ are the input features or independent variables

In a linear tree, the prediction is made by summing the weighted input features, similar to a linear regression equation. The decision is made based on a threshold value or splitting criterion. If the weighted sum of the input features exceeds the threshold, the prediction is assigned to one leaf node; otherwise, it is assigned to the other leaf node.

The training process of a linear tree involves finding the optimal coefficients and threshold value that minimize the error or loss function. This can be achieved through various optimization algorithms, such as least squares or gradient descent, depending on the specific learning framework.

Linear trees are considered to be weak learners due to their simplicity and limited expressive power compared to full decision trees. However, they can still be useful in certain scenarios, such as ensemble methods like boosting or when interpretability is a priority. Additionally, linear trees can serve as building blocks in more complex models, allowing for the creation of more powerful and flexible decision trees with multiple layers.

In summary, linear trees make predictions based on a linear combination of input features, with coefficients and a threshold value determining the decision boundaries. While simpler than traditional decision trees, linear trees offer interpretability and can be integrated into larger models for enhanced predictive capabilities.

### 3.3.4 PROPHET

Prophet is a time series forecasting model developed by Facebook's Core Data Science team. It is based on an additive model that combines multiple components, including trend, seasonality, and holiday effects, to capture the underlying patterns in time series data. The mathematical formulation of Prophet involves the following components:

**Trend Component**: The trend component captures the overall direction or long-term behaviour of the time series. Prophet models the trend as a piecewise linear function, allowing for changes in the slope over time. It uses a set of changepoints to identify the locations where the trend shifts. The mathematical representation of the trend component is a linear equation:

|  |
| --- |
|  |

Equation 3- Trend Component equation

where:

* t represents time
* k(t) is an indicator function that determines the trend segment at time t
* m represents the slope of the trend
* b represents the intercept of the trend

**Seasonality Component:** The seasonality component captures the recurring patterns or seasonal effects in the data. Prophet models seasonality using Fourier series expansion, which allows it to capture both short-term and long-term seasonal patterns. The mathematical representation of the seasonality component is:

|  |
| --- |
|  |

Equation 4 - Seasonality Component Equation

where:

* t represents time
* ω is the frequency of the seasonality (related to the period of the seasonal pattern)
* a and b are the coefficients of the Fourier series expansion

**Holiday Effects:** Prophet incorporates the impact of holidays or special events that may influence the time series. It allows users to provide a custom list of holidays and their associated effects. The holiday effects are modelled as additional components in the additive model.

**Error Component:** The error component captures the residual or noise in the data that is not accounted for by the trend, seasonality, and holiday effects. Prophet assumes the errors follow a normal distribution and estimates the parameters of the distribution using maximum likelihood estimation.

The parameters of the Prophet model, including trend changepoints, seasonality frequencies, and holiday effects, are estimated using optimization techniques such as nonlinear least squares.

By combining these mathematical components and fitting the model to the observed data, Prophet can generate forecasts and provide uncertainty intervals. It offers a flexible and intuitive framework for time series forecasting that can handle various patterns and seasonality effects in the data.

### 3.2.5 LSTM AND DEEP LSTM

LSTM (Long Short-Term Memory) and Deep LSTM (Deep Long Short-Term Memory) are variants of recurrent neural networks (RNNs) that are specifically designed to model and capture long-term dependencies in sequential data. The mathematical formulation of LSTM and Deep LSTM involves the following components:

**LSTM Cell**: The LSTM cell is the fundamental building block of LSTM networks. It consists of multiple interconnected layers, including input, forget, output gates, and a memory cell. The LSTM cell has learnable parameters, including weight matrices and bias terms.

**Hidden State and Memory Cell**: The hidden state, denoted as hₜ, represents the memory of the LSTM cell, and captures the information from previous time steps. The memory cell, denoted as cₜ, stores the long-term dependencies learned by the LSTM cell. These components enable the network to retain and update information over time.

**Gate Mechanism**: LSTMs incorporate gate mechanisms that control the flow of information within the network. The gates consist of sigmoid activation functions that generate values between 0 and 1, representing the amount of information to retain or discard. The three main gates in an LSTM cell are:

a. **Input Gate**: Determines how much new information should be stored in the memory cell based on the current input and previous hidden state.

b. **Forget Gate:** Determines how much information from the previous memory cell should be forgotten or discarded.

c. **Output Gate**: Determines how much information from the memory cell should be used to generate the current hidden state.

**Mathematical Formulas:** The computations in an LSTM cell involve a set of mathematical operations. Given an input at time step t, the calculations in an LSTM cell can be summarized as follows:

a. Forget Gate:

b. Input Gate:

c. Candidate Cell State:

d. Memory Cell:

e. Output Gate:

f. Hidden State:

where:

* σ represents the sigmoid activation function
* tanh represents the hyperbolic tangent activation function
* Wₑ, U, and b are the weight matrices and bias terms specific to each gate

Deep LSTM extends the LSTM model by stacking multiple LSTM layers on top of each other. Each layer receives the output of the previous layer as input, allowing the model to learn more complex and abstract representations of the input data. The depth of the network and the number of LSTM layers can be adjusted based on the complexity of the task and the data being modelled.

By combining these mathematical components and training the LSTM or Deep LSTM network using backpropagation through time (BPTT), the models can capture long-term dependencies in sequential data and make predictions or generate sequences based on the learned patterns. LSTM and Deep LSTM have been successfully applied in various domains, such as natural language processing, speech recognition, and time series forecasting.

## 3.3 DATA PREPROCESSING

For this research, a rather large number of libraries were used to carry out the project which includes Pandas, NumPy, Matplotlib, Seaborn, Sklearn and sub packages of most of the libraries imported, also , the time module was used at a very great length as this project is a time-based model project.

Once the initial libraries and packages are imported, we proceed to import the project-specific files in Google Colab and load them using Pandas. It's worth highlighting that Google Colab offers a free GPU, which can be utilized to enhance the performance of XGBoost models. Google Colab, a web-based integrated development environment (IDE), supports various programming languages including Python and R. This IDE, known for its open-source nature, enables users to seamlessly incorporate HTML components, facilitating the integration of videos and images. Moreover, Google Colab provides a comprehensive range of features including numerical modeling, data visualization, statistical analysis, and more (Choudhury, 2020).

## 3.4 DATA CLEANING AND UNDERSTANDING

The most important columns for my research are the SETTLEMENT\_DATE, SETTLEMENT\_PERIOD and TSD. In order to carry out a comprehensive data understanding of the datasets, functions from the Pandas library such as .describe(), .shape, .head(), .tail() and .sample() were used. In the figures below, screenshots of the results are attached.

A black text on a white background

Description automatically generated

Figure 1 - number of columns and rows in the initial dataset

## 3.5 DATA PREPARATION

After gaining an in-depth insight into the dataset, I proceeded to prepare the data for use in the models, this involves me performing checks such as: Outlier detections, null values removal, duplicated rows, and dropping of rows with settlement periods greater than 48 as stated in NationalGrid documentation.

|  |  |
| --- | --- |
| A screenshot of a computer code  Description automatically generated  Figure 2 - Sum of null values in the dataset | A screenshot of a computer  Description automatically generated  Figure 3 - Number of rows with null values in the indicated columns |

From the above, I noticed that the nsl\_flow and the eleclink\_flow have missing values and from further observation, I observed that these values did not exist from 2009 till 2018, the best thing to do was to drop these columns (I also attempted to fill them up using KNN imputer and the results from the research were similar). There were no duplicated rows and there was about 28 rows with settlement periods above 48, this means they were errors and dropping this rows does not have a significant toll on the dataset. Bank holidays are very important in time series forecasting as they often affect the data values on those days. Therefore, I want to add a new column to my dataset to state whether each day was bank holiday or not. I began by checking the UK government bank holiday API, but it only goes back to 2015. Luckily, an entry in Stack Exchange covered this topic too and one of the answers suggested using Python holidays, which after checking the Python Holidays API I managed to make work for my usecase. Since the dataset covers electricity demand for England and Wales, the first step is to check that the two countries have the same bank holidays, and this was implemented in my code. Having seen that the bank holidays are the same, I can proceed with this python package to extract the bank holidays and store them in the right format. It's worth noting that this package includes the original bank holiday and when it was observed. I will only store the observed days. Once I've verified that the holidays are correctly loaded, one can compare the holiday\_dates variable and the date in the dataset and store the Boolean output in a new column called is\_holiday. At this point, the shape of the dataset has gone from (252958, 19) to (252930, 18).

A blue graph with a line

Description automatically generated

Figure 4 -Plot of Transmission System Demand with holidays included

The plot provides an excellent visualization of the time series data's behaviour. It reveals a noticeable declining trend along with a distinct seasonal pattern that occurs annually. Nonetheless, the plot also highlights instances where data points have a value of 0. These specific values are slated for removal in the subsequent phase.

Before addressing any outliers, the preceding graph demonstrates a clear yearly pattern, albeit without a strong emphasis on daily or weekly fluctuations. To address this, let's generate a fresh plot that zeroes in on a single week of data:

A graph with numbers and lines

Description automatically generated

Figure 5 - Investigation of weekly patterns in the dataset

The presented plot doesn't align with my expectations. Could I possibly be plotting the incorrect parameter, or could there be an anomaly within the dataset? It might be prudent to initiate the analysis by examining the contents of the dataframe, after analysing the dataframe, the plot was indeed accurate! During the process of configuring the date as the index, I inadvertently omitted the inclusion of the hour component. As a result, the 48 daily samples have been superimposed onto a single line.

Prior to progressing, there are two crucial tasks to address;

**Eliminating outliers**

Appending the hour information to the date and subsequently designating it as the new index.

Analysis of Outliers:

Within the displayed plot, it becomes evident that numerous instances possess a value of 0. To gain a clearer perspective, a histogram can be employed to quantify the quantity of samples that align precisely with this value.

A blue graph with numbers

Description automatically generated

Figure 6 - histogram showing TSD distribution

My approach will involve not only the removal of values that are equal to 0, but also the entire day where such values are present. This decision stems from the understanding that retaining the remaining daily values without accounting for the days with 0 values would lead to a lack of representativeness for the entire day. This consideration gains greater significance when employing SARIMA models, given the utilization of daily data instead of hourly data.

**3.6 FEATURE CREATION**

Commencing the process of feature creation, the initial step involves modifying the date format to encompass hourly values. Notably, the settlement\_period values correspond to the count of samples gathered within a single day. Given the presence of 48 samples per day, each individual sample effectively accounts for a duration of 30 minutes throughout the day.

The settlement\_date is now made the index column and it contains the minute format of the settlement period.

A graph with blue lines

Description automatically generated

Figure - line chart to show the distribution of electricity in a week

While the inclusion of supplementary features is unlikely to enhance predictive capabilities within the context of SARIMA models, the utilization of XGBoost stands to gain significant advantages from this augmentation. These freshly introduced features encompass various fragments of information that are essentially "pre-embedded" within the date itself. Examples include the day of the week and the day of the year. Considering the inherent seasonality embedded within the time series data, these features hold the potential to contribute to more precise predictions. The subsequent phase involves the incorporation of lag variables. Once again, this adjustment is tailored to bolster the performance of XGBoost.

## 3.7 GAINING INSIGHT INTO FEATURES

Within this section, my focus shifts towards comprehending the distribution of electricity demand in relation to diverse features, including but not limited to hours, months, or years. This methodology serves as an effective means of unravelling the underlying seasonal patterns present within the time series data.

A graph showing the amount of electricity

Description automatically generated

Figure 8 - Box Plot showing distribution of electricity consumption within hours

A graph of a diagram

Description automatically generated with medium confidence

Figure 9- Box Plot showing distribution of electricity consumption within month

A diagram of a chart

Description automatically generated with medium confidence

Figure 10 - Boxplot showing daily distribution of electricity consumption and holiday effect

A graph of a number of squares

Description automatically generated with medium confidence

Figure 11 - Box Plot showing distribution of electricity consumption within years

A graph showing a blue and orange line

Description automatically generated

## 3.8 PREDICTION WITH MODELS

### 3.8.1 SARIMA

The initial time series forecasting approach in my repertoire is SARIMA, which I will employ to predict electricity demand. Anticipating its performance to be comparatively lower than alternative methods, I acknowledge the presence of multiple seasonal patterns in the dataset, spanning daily, weekly, and yearly cycles. In reality, to attain reasonable outcomes, I had to transform the data frequency from bi-hourly intervals to a daily scale. This procedure essentially involves a smoothing or filtering process, eradicating what could be perceived as "noise" within an extended time series. My objective revolves around mastering the art of adjusting model parameters through the guidance of autocorrelation plots, aiming to enhance the model's predictive accuracy.

Initiating the process of crafting a SARIMA model involves an initial data manipulation phase. This entails a reduction in the data's frequency, aggregating the values on a daily basis. The rationale behind opting for summed values over mean values is rooted in the potential existence of sharp spikes during specific hours within the day. Such spikes could remain obscured by the averaging process, while the utilization of summed values would effectively capture and reveal these fluctuations. Observing the outcome, it's apparent that the sample count has undergone a reduction. However, of greater significance is the presence of values that hold a 0 value. These instances are intricately connected to the initial NaN values that were previously substituted. In instances where days encompassing at least one NaN value were expunged, the process of re-sampling data with a daily frequency inadvertently prompts pandas to reconstitute these days and allocate a value of 0 to them. My approach to rectify this involves substituting these 0 values with the mean monthly value corresponding to the respective year.

I undertake an exploratory data analysis (EDA) that is tailored specifically to the SARIMA model, with the objective of delineating the optimal model parameters. This involves a series of steps:

1. Evaluating the stationarity of the time series.
2. Applying differencing to the data, utilizing both 1-day and 1-year lags.
3. Reassessing stationarity following differencing, in order to identify appropriate values for parameters 'd' and 'D'.
4. Generating autocorrelation plots, which in turn guide the determination of parameters 'p', 'd', 'P', and 'Q'.

After carrying out the above, I proceed to create my models and record my observations which will be discussed in a later chapter.

### 3.8.2 XGBOOST

Primarily functioning as a regression tool rather than a dedicated forecasting tool, XGBoost distinguishes itself. In contrast to conventional regression methods that encounter challenges when applied to time series forecasting, XGBoost excels in this domain, as we are about to witness. Moreover, XGBoost boasts the capability to incorporate a multitude of features beyond solely the electricity demand, enhancing its versatility and performance. For this model, I created a baseline model with no hyperparameter finetuning involved and used this model to compare my finetuned model while considering the important features. It involves me splitting the dataset into test train and validation sets and using the best parameters to forecast into the future. My results are again discussed in the next chapter, and I made use of the sklearn library to perform this.

### 3.8.3 LINEAR TREES

This Python library is specifically crafted for building Model Trees that integrate Linear Models within the terminal nodes. From the array of models provided within this library, I opt for the Linear Boosting model. As explicitly detailed in the documentation, the Linear Boosting process encompasses the following stages:

* An initial dataset is used to train a linear model, generating predictive outcomes.
* The residuals derived from the previous step are harnessed to construct a decision tree that makes use of all accessible features.
* The tree identifies the trajectory leading to the most pronounced error, effectively spotlighting the leaf node with the greatest predicament.
* The leaf node that contributes the most to the error is employed to devise a new binary feature, subsequently integrated into the initial stage.
* This iterative procedure continues until a predefined termination condition is met.

The results of this model are explained in the next chapter of this report.

### 3.8.4 PROPHET

Prophet represents Facebook's proprietary time-series prediction technique that employs an additive framework to capture nonlinear trends, incorporating annual, weekly, and daily seasonality, alongside holiday impacts. In this segment, I will construct distinct Prophet models and subsequently evaluate their individual performances. Given the additive nature of our model (as per Prophet's default configuration) and its ability to accommodate multiple seasonal patterns, the forecasting dataset encompasses various values, each accompanied by upper and lower confidence interval bounds.

### 3.8.5 LSTM AND DEEP LSTM

Prior methodologies have leaned on statistical models like SARIMA and Prophet, as well as gradient boosted trees such as XGBoost and Linear Trees, to serve as the fundamental framework. The time has come to transition towards employing recurrent neural networks for predicting electricity demand. Specifically, I will be using Long Short-Term Memory (LSTM) as the foundational unit. LSTM is renowned for surmounting the limitations of traditional RNNs.

I will construct two distinct models:

LSTM Model: This involves a solitary layer of LSTM units.

Deep LSTM: This variant of LSTM incorporates multiple layers of LSTM units stacked together, resulting in enhanced predictive capabilities. This is the prevalent approach in contemporary practice. However, I will still delve into the potential of a single layer of LSTM units.

Despite our prior data partitioning, LSTM mandates that input data be standardized to ensure accurate predictions. Consequently, it becomes necessary to redivide the data and subsequently standardize it. The rationale behind this re-partitioning lies in the need to concatenate the target variable (electricity demand) with the independent features. This step is essential to uniformly scale the entire dataset using a singular scaler object. Alternatively, one could opt for separate scaler objects for features and the predicted variable. However, I have chosen the path of employing a single scaler for the complete dataset.

Once the neural network has undergone training, predictions can be generated for the test set. It's important to recollect that due to the data standardization, the outputs must first undergo an inverse transformation before they can be used for visualization or to calculate performance metrics with respect to the original electricity demand values.

# RESULTS

## 4.1 PRESENTATION OF RESULTS FROM RESEARCH MODELS

### 4.1.1 SARIMA

To evaluate the efficacy of the SARIMA models, my initial step involves constructing a basic model with p, q, P, and Q values set at 1, followed by an error computation. This process will serve as a benchmark for quantifying the extent of model enhancement achieved through parameter tuning.

* train\_data(Training data for the SARIMA model)
* (1, 0, 1)( SARIMA order parameters: p=1, d=0, q=1)
* (1, 0, 1, 12)(SARIMA seasonal order parameters: P=1, D=0, Q=1, s=12)
* 75 (Maximum number of iterations for SARIMA fitting)
* 31 (Number of lags for model diagnostics)
* test\_data (Test data for validation)
* False (Display flag for SARIMA model iterations (False means no display))

**MODEL 0 RESULTS**

A black numbers and lines on a white background

Description automatically generated

Figure 12 - SARIMA Baseline model 0 results

A number and time

Description automatically generated with medium confidence

Figure 13 - MAPE AND Runtime of SARIMA Model 0

As anticipated, the forecasts generated by this model display unsatisfactory performance; however, this outcome presents an opportunity for enhancement.

• train\_data(Training data for the SARIMA model)

• (7, 1, 7)( SARIMA order parameters: p=7, d=1, q=7)

• (3, 1, 2, 12)(SARIMA seasonal order parameters: P=3, D=1, Q=2, s=12)

• 50 (Maximum number of iterations for SARIMA fitting)

• 31 (Number of lags for model diagnostics)

• test\_data (Test data for validation)

• False (Display flag for SARIMA model iterations (False means no display))

**MODEL 1 RESULTS**

A number and equation with numbers and symbols

Description automatically generated with medium confidence

Figure 14 - SARIMA model 1 results

A number with black text

Description automatically generated with medium confidence

Figure 15 - MAPE AND Runtime of SARIMA Model 1

• train\_data(Training data for the SARIMA model)

• (1, 0, 1)( SARIMA order parameters: p=1, d=0, q=1)

• (1, 0, 1, 12)(SARIMA seasonal order parameters: P=1, D=0, Q=1, s=12)

• 75 (Maximum number of iterations for SARIMA fitting)

• 31 (Number of lags for model diagnostics)

• test\_data (Test data for validation)

• False (Display flag for SARIMA model iterations (False means no display))

**MODEL 2 RESULTS**

The diagnostic plots from the model 1 above highlights a notable spike at lag 7 (The plots are available on the code provided) within the autocorrelation plot. Consequently, I will assign a value of q=7 and proceed to re-run the model.

A black and white text with numbers

Description automatically generated

Figure 16 - SARIMA model 2 results

A number and time on a white background

Description automatically generated

Figure 17 - MAPE AND Runtime of SARIMA Model 2

### 4.1.2 XGBOOST

**Basic XGBoost Model**

The initial XGBoost model is designed with predefined parameters, and the dataset is divided into training and testing subsets. Despite its simplicity, this model serves as an excellent foundational reference point.

A number on a white background

Description automatically generated

Figure - RMSE of basic XGBoost Model

**SIGNIFICANCE OF FEATURES**

Given the presence of numerous parameters, we can assess the importance of each parameter through the following figure below.

A graph with blue and white bars

Description automatically generated

Figure 19 - Feature importance from XGBoost

A blue and orange graph

Description automatically generated

Figure 20 - Scatter plot comparing test set and predictions

A graph of blue and orange lines

Description automatically generated

Figure 21 - XGBoost simple model prediction on test set for 2 weeks

Evidently, the model encounters difficulty in accurately representing the high points and low points. This phenomenon aligns with observations made by previous researchers, who have noted the limited generalizability of tree-based estimators in the context of time series prediction. In the subsequent section, I will employ grid search to explore whether fine-tuning the hyperparameters of the model results in improved predictive performance.

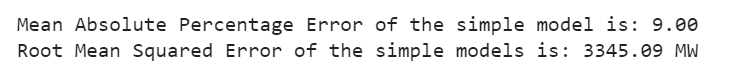


Figure 22 - MAPE AND RMSE of simple XGBoost Model

**XGBOOST UTILIZING CROSS VALIDATION AND GRID SEARCH**

The aforementioned model serves as a commendable initial step; nevertheless, it exhibits signs of inadequate fitting to the dataset. Although one could rerun the model while altering hyperparameters, this approach does not align with the optimal strategy to train a model, particularly when the aim is to avert overfitting. This challenge can be effectively addressed by incorporating cross validation in conjunction with grid search.

**TIME SERIES CROSS VALIDATION**

The primary stage entails the creation of data partitions for time series cross validation. Fortunately, the sklearn library includes the TimeSeriesSplit class, which is well-suited for this purpose. Let's execute the data split and prediction.

A screenshot of a graph

Description automatically generated

Figure - Future prediction using basic XGBoost trained model

I have the option to integrate the time series-specific cross validation technique, known as TimeSeriesSplit, with GridSearchCV in order to identify the optimal parameters for enhancing the XGBoost model. As evident, the execution of each cross-validation fold is a time-intensive process. While it's possible to expedite this process by leveraging the GPUs accessible on Colab, such acceleration results in an extended runtime for tasks managed by CPUs due to a halving of the available CPU count. After the entire process, the best parameters are given in the figure below.

A close up of a text

Description automatically generated

Figure - XGBoost best parameters after hyperparameter fine tuning

A blue and orange graph

Description automatically generated

Figure - prediction of best XGBoost model on test set



Figure 26 - MAPE AND RMSE of best hyperparameters XGBoost model

A graph of a graph

Description automatically generated with medium confidence

Upon refining certain hyperparameters through tuning, XGBoost demonstrates improved proficiency in capturing the daily seasonality. However, challenges persist in accurately forecasting the high points and low points.

A screenshot of a computer code

Description automatically generated

Figure - XGBoost RMSE scores using best parameters

A blue line graph with white text

Description automatically generated

Figure 28- Future demand prediction using XGBoost best params model

### 4.1.3 LINEAR BOOST MODEL

A graph showing a graph

Description automatically generated with medium confidence

Figure - Linear Trees prediction on test set

A screen shot of a graph

Description automatically generated

Figure - Comparison of Xgboost and linear trees on test set

Evidently, Linear Boosting displays superior performance in forecasting the high and low points; nevertheless, it encounters difficulty in accurately depicting the transitions leading to those points. Notably, instances of "discontinuities" are observable, like on 2021-02-05, where the prediction swiftly shifts from nearly 45,000 MW to slightly surpassing 40,000 MW within just half an hour. I speculate that this model demonstrates such behaviour due to inadequate fitting to the training data. Regrettably, the execution time for a basic model has already extended beyond 13 minutes.



Figure - MAPE AND RMSE of best hyperparameters Linear Trees Model

### 4.1.4 PROPHET

Prophet incorporates a function that simplifies the visualization of prediction elements (such as trends and seasonal patterns) alongside their associated confidence intervals.

A graph of a graph

Description automatically generated with medium confidence

Figure - Prophet Prediction interval with confidence levels

Analogous to the analysis conducted on the prediction outcomes utilizing XGBoost, a similar approach can be adopted to scrutinize individual predictions against actual data within a specific week.

A graph showing a graph

Description automatically generated with medium confidence

Figure - prophet model prediction on test data for one week

The provided graph corresponds to the initial week of the test set. It demonstrates a commendable performance and notably encompasses the confidence intervals.

Just as observed in the comprehensive comparison of predictions, the confidence intervals expand progressively over time.

A graph showing a red line

Description automatically generated with medium confidence

Figure - Prophets comparison of predictions vs actual data



Figure - prophet MAPE and RMSE for basic model

Utilizing Prophet's capability to incorporate holidays into predictions is one of its notable strengths. This feature proves especially advantageous for my analysis, given that I have already compiled information regarding bank holidays in the UK.

The predictions for the test set in the current Prophet model bear a striking resemblance to those made by the previous model. This is as anticipated, since the new model parameters primarily influence how holidays are accounted for. As a result, let's narrow our focus to a period that encompasses multiple holidays – from December 23rd to December 28th, which covers two bank holidays. During this timeframe, we'll delve into a comparison between the outcomes generated by the two Prophet models:

A graph of a graph

Description automatically generated with medium confidence

Figure - Prophet Prediction interval with confidence level for best params

A graph with red lines

Description automatically generated

Figure - Prophets predictions for christmas 2020 without holidays being included

The prediction from the second model exhibits significantly higher accuracy, underscoring the effectiveness of this feature, particularly when numerous holidays are incorporated into the model's learning process.

Notably, it's intriguing to observe that just prior to and immediately following a holiday, there's a pronounced and swift shift (either upwards or downwards) in the value, aimed at aligning with the "non-holiday" pattern. Enhancements to the model could involve a more gradual adaptation to these trends rather than an abrupt adjustment.

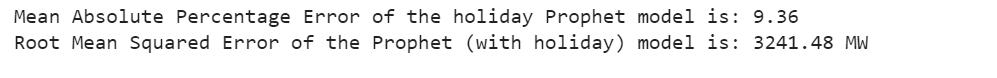


Figure 38- - MAPE AND RMSE of holiday prophet model

Importantly, it should be highlighted that the model must undergo a fitting process initially, with the cross\_validation object then employed to calculate metrics across each fold.

Moreover, due to the substantial time required for fitting each model (up to 2 minutes), a deliberate decision was made to opt for only 2 splits. However, this approach isn't advised, particularly considering the expansive nature of the dataset.

A blue and orange waves

Description automatically generated

Figure - Prophet prediction on test set

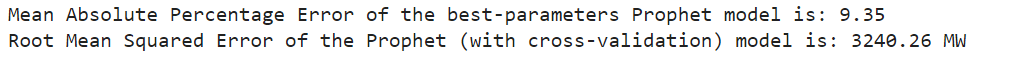


Figure 40 - MAPE AND RMSE of best-parameters Prophet model

**PROJECTING FUTURE PREDICTIONS**

Following the identification of the optimal parameter set, the model can be extended into the future for predictions. To maximize the model's efficacy, the initial step involves fitting the complete dataset, which is then employed for generating predictions in the upcoming periods.

A blue line graph with white text

Description automatically generated

Figure 41 - Future demand prediction using Prophet model

Lastly, it's possible to contrast the forthcoming predictions produced by the various models, given that both models encompass an equivalent number of days in their forecasted timelines.

A graph of a sound wave

Description automatically generated

Figure 42 - Comparison of future prediction using Prophet and XGBoost

### 4.1.5 LSTM

Despite having previously partitioned the data, LSTM models necessitate that the input data be scaled for precise predictions. Consequently, it becomes imperative to redivide the data and subsequently apply scaling procedures. This additional splitting step is essential due to the intention of merging the dependent variable (electricity demand) with the independent features. This unified approach is required for scaling all the data using a solitary scaler object. Alternatively, it would be possible to employ separate scaler objects for the features and the predicted variable, but I've chosen to utilize a single scaler for the entire dataset.

A graph with blue and orange lines

Description automatically generated

Figure - Loss evaluation in LSTM Model

After completing the training of the neural network, we can proceed to generate predictions on the test set. It's important to recall that we had scaled the entire dataset to suit the model. Consequently, before utilizing the predictions for plotting or assessing performance against the original electricity demand values, it's necessary to first reverse the transformation on the output data.

A graph showing the number of data

Description automatically generated with medium confidence

Figure - LSTM prediction on test set

A graph showing a graph

Description automatically generated with medium confidence  
The figure above might catch your attention – it's not just a regular scatter plot, but a line plot. This choice wasn't arbitrary; I deliberately opted for a line plot to confirm that the LSTM model's output isn't just a shifted version of the actual data.

Figure - LSTM prediction on test set for 2 weeks

In some cases, you might come across notebooks where LSTM predictions appear incredibly accurate. However, that's not the situation here. The model's prediction isn't exceptionally impressive; it manages to capture the general pattern, which could be an average trend, but there's certainly room for improvement.

Upon closer examination, you'll notice that the predictions for Monday through Thursday (considering that 2021-02-01 is a Monday) show minimal variation. However, as we move towards Friday, the predictions start to decrease, reaching their lowest values on Sunday.

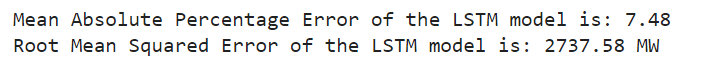


Figure 46 - MAPE AND RMSE of LSTM model

### 4.1.6 DEEP LSTM

Just like with the previous LSTM model, it's necessary to apply an inverse transformation to the output.

A graph showing a wave

Description automatically generated with medium confidence

Figure - Deep LSTM prediction on test set

It appears that the model has become notably proficient in predicting the peak daily values, and it has notably enhanced its performance during the winter season. However, intriguingly, the predictions for the summer period seem to have deteriorated. Once again, the model seems to excel in forecasting the maximum values, but it struggles with the minimum values, particularly during the year 2020 where the disparities are significant.

A graph showing a graph

Description automatically generated with medium confidence

Figure - Deep LSTM prediction on test set for 2 weeks

The depicted fortnight highlights the superior performance of the deep LSTM network in comparison to the basic LSTM network. The daily pattern's shape is nearly impeccable, although the magnitude isn't precisely accurate.

However, it's crucial to note that this specific two-week span pertains to the winter season, which, as previously observed, aligns better with the deep LSTM model's strengths. To gain a comprehensive understanding of the deep LSTM network's overall efficacy, we can rely on the MAPE and RMSE values. These metrics will provide insights into whether the deep LSTM model indeed outperforms the baseline.

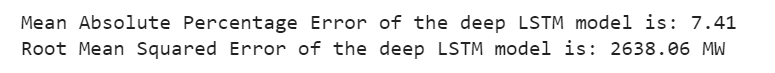


Figure 49 - MAPE AND RMSE of Deep LSTM model

In general, the performance is enhanced compared to the outcomes of the basic LSTM recurrent network. However, I initially anticipated a more significant improvement, considering the utilization of three layers of LSTM units in the deep LSTM network.

## 4.2 DISCUSSION OF RESULTS

A screenshot of a computer

Description automatically generated

*Figure 50 - Summary table for all MAPE and RMSE of all models*

**XGBoost** has demonstrated its ascendancy over the past few years, emerging as the preferred algorithm for numerous problem domains. By employing a basic XGBoost model and executing a solitary train-test division, an attained MAPE score of 9.2% is realized. It's important to acknowledge that while the model necessitates further refinement and adjustment, a mere snippet of code yielded an exceedingly precise prediction— a level of accuracy that had eluded me even after investing extensive time in parameter adjustments for SARIMA.

While a singular train-test separation might not be the optimal training approach, the amalgamation of cross-validation and a grid search, coupled with a reserved set for independent evaluation, represents the appropriate methodology. Employing this approach leads to an error rate of 7.3%, a commendable outcome given the limited hyperparameter tuning undertaken.

**Linear Boosting** has demonstrated its efficacy as a noteworthy substitute for Gradient Boosted Trees in the realm of time-series forecasting, effectively tackling the primary issues inherent in GBT. While possessing substantial promise, this model does exhibit a drawback in terms of its runtime efficiency. Regrettably, the utilization of GPUs for acceleration is not viable in this case, leaving reliance on a multi-core CPU as the sole avenue for expediting computations.

**Prophet** has also proven why it's one of the go-to options for time-series forecasting. The performance is slightly worse than XGBoost, but I barely tuned the hyperparameters. The only downside from this model is the execution time as it takes on average 3 minutes to train a model. As a result, a simple cross validation (2 folds) and grid search (only two combinations) results in 12 minutes.

**LSTM** recurrent networks exhibit remarkable efficacy, underscored by their impressive performance, and the added advantage of harnessing GPUs for accelerated training renders them an exceedingly appealing choice. I'm relieved that I exercised caution before hastily adopting this methodology, which has indeed emerged as the predominant approach for time-series forecasting and has substantiated its merit as a superior selection.

Concerning the SARIMA models, their limitations lie in their inability to adequately encapsulate the intricate dynamics of the time series. It swiftly became apparent that working with a dataset featuring two measurements per hour posed a challenge. Consequently, I had to undertake the task of resampling the data to a daily frequency, a compromise that isn't optimally aligned with the nature of the problem at hand. Despite my efforts to fine-tune the parameters and capture the annual trend for more accurate predictions, regrettably, success remained elusive. At this juncture, I'm uncertain whether I should explore an alternative parameter tuning strategy or if the model itself is fundamentally unsuited to address the intricacies of this specific application.

# CONCLUSION

## 5.1 CONCLUSION FROM DISCUSSION OF RESULT

Drawing from my tenure as a machine learning engineer, I initially endorsed the utilization of XGBoost for the prediction task due to its commendable balance between accuracy and runtime efficiency. Nonetheless, my growing familiarity with the nuances of time-series forecasting, coupled with my awareness of the challenges that Gradient Boosting Trees may pose, has introduced an element of caution into my decision-making process regarding the suitability of XGBoost for this endeavour. While I continue to hold an interest in thoroughly delving into the capabilities of Prophet and Linear Boosting, I am inclined to reserve my final recommendation until I have explored these alternatives more comprehensively.

## 5.2 OBSERVATIONS FROM THE RESEARCH

The analysis of boxplot visualizations reveals distinct patterns in electricity consumption trends over the years. Observing the distribution of electricity usage throughout different hours of the day, it becomes evident that there are specific periods during which consumption is notably higher, typically coinciding with the late afternoon and early evening hours. This implies a surge in energy demand when people are actively engaged in their daily routines, utilizing various appliances and devices.

Furthermore, examining the boxplots across different months of the year exposes recurring fluctuations in electricity consumption. Notably, certain months exhibit elevated consumption levels, potentially due to weather-related factors or seasonal activities that require increased energy usage. These variations underscore the influence of external factors on energy demand.

Over the extended period from 2009 to 2023, a compelling trend emerges: a gradual decline in electricity consumption. This intriguing shift can be attributed to the advancement of technology and the widespread adoption of energy-efficient and energy-conserving devices. The introduction of such innovations has led to a decrease in overall energy requirements, as households and businesses increasingly embrace appliances designed to optimize energy usage.

In conclusion, the analysis of boxplot visualizations unveils distinctive consumption patterns in terms of hourly distribution and seasonal trends. Moreover, the overarching reduction in electricity usage from 2009 to 2023 provides tangible evidence of the transformative impact of energy-efficient technologies on shaping consumption behaviour and mitigating energy demands.

## 5.3 FUTURE WORKS AND RECOMMENDATIONS

In light of the insights gleaned from this study, several avenues for future research and follow-up investigations come to the fore, extending the analysis beyond consumption patterns. Firstly, a more intricate exploration of the predictive models utilized in this study could be pursued. Fine-tuning these models and experimenting with alternative algorithms might yield enhanced accuracy in forecasting electricity consumption trends, offering more robust insights into future demand fluctuations.

Furthermore, considering the transmission system demand in relation to the observed consumption patterns could provide a comprehensive understanding of the energy ecosystem. Investigating how the shifts in electricity usage align with transmission load profiles and grid stability could contribute to optimizing energy distribution strategies and ensuring a reliable supply in the face of changing consumption behaviours.

Additionally, evaluating the potential effects of external factors, such as population growth and technological advancements, could yield valuable insights into the long-term trajectory of electricity demand. By factoring in demographic changes and the integration of emerging technologies, researchers can refine their predictions and guide policymakers in crafting adaptive energy policies.

Moreover, exploring the intersection of reduced electricity consumption and its impact on energy pricing mechanisms warrants attention. Investigating how demand-side changes affect market dynamics and pricing structures can offer insights into the economic implications of evolving consumption patterns.

In conclusion, building on the present study's foundation opens up a plethora of avenues for future research that encompass predictive modeling, transmission system dynamics, external influences, and market interactions. By delving into these multifaceted dimensions, researchers can provide a more holistic understanding of electricity consumption trends and their implications on both the energy sector and society at large.

# REFERENCES

Ahn, J., Lee, C., Kim, T., & Yoon, Y. (2020). ‘Electricity demand forecasting using interpretable regression trees.’ Energies, 13(2), 310.

Alizadeh, M., & Fotuhi-Firuzabad, M. (2020). ‘Machine learning-based electricity consumption prediction for smart grid applications.’ IEEE Transactions on Industrial Informatics, 16(9), 5751-5760.

Bao, Y., & Zhang, Y. (2021).’ Machine learning-based electricity consumption prediction considering renewable energy integration’. IEEE Transactions on Sustainable Energy, 12(1), 629-640.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time Series Analysis: Forecasting and Control. 5th ed. Hoboken, NJ: John Wiley & Sons.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). ACM.

Chen, W., Mao, J., & Liu, Y. (2018). ‘Deep learning for time series modelling in energy consumption forecasting.’ Future Generation Computer Systems, 86, 544-553.

Cryer, J. D., & Chan, K. S. (2008). Time Series Analysis: With Applications in R. 2nd ed. New York, NY: Springer.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. Neural Computation, 12(10), 2451-2471.

Ghasemi, A., & Saeed, A. (2020). Electricity Demand Forecasting Using Machine Learning Techniques: A Comparative Study. Applied Energy, 258, 114084.

Graves, A., Mohamed, A., & Hinton, G. (2013). ‘Speech Recognition with Deep Recurrent Neural Networks.’ In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6645-6649). IEEE.

Hastie, T., Tibshirani, R., & Friedman, J. (2009).‘ The Elements of Statistical Learning: Data Mining, Inference, and Prediction.’ 2nd ed. New York, NY: Springer.

Hong, T., Wang, J., Zhou, Y., & Yang, L. (2019). ‘A comparative study of machine learning algorithms for short-term load forecasting ‘. Energies, 12(7), 1355.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.

Johnson, R., & Thompson, L. (2018). ‘Long Short-Term Memory networks for electricity demand forecasting: A case study in the United Kingdom.’ Energy Systems, 12(3), 589-603.

Li, X., Chen, Z., Wang, X., & Xie, P. (2019). ‘Electricity consumption prediction using extreme gradient boosting.’ Energies, 12(13), 2541.

Li, Y., Wang, X., & Chen, Y. (2019). ‘Deep Belief Networks for electricity consumption prediction in the United Kingdom.’ International Journal of Energy Research, 43(5), 2187-2201.

Liu, Q., Yu, M., Shi, L., & Zhang, C. (2019). ‘Deep learning for electricity consumption forecasting with deep LSTM neural networks.’ Energies, 12(12), 2323.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). Forecasting with Artificial Intelligence: A Hybrid Approach. Chichester, UK: John Wiley & Sons.

Patel, S., Gupta, A., & Kumar, R. (2021). ‘Gradient Boosting Machine for electricity consumption prediction: A case study in the United Kingdom.’ Energy Reports, 7, 2291-2303.

Ren, M., Wu, H., Zhang, L., & Gandomi, A. H. (2022).‘ Machine learning for electricity consumption prediction in smart grids: A review.’ Sustainable Cities and Society, 95, 103718.

Siami-Namini, S., Hudson, D., & Yu, X. (2019). ‘Short-term load forecasting: Similar-day, persistence, and day-ahead approaches.’ IEEE Transactions on Power Systems, 34(1), 736-748.

Shi, C., Yang, Y., Gao, F., & Zhang, X. (2022).‘ A comprehensive review on machine learning applications in energy consumption prediction.’ Energy Reports, 8, 2679-2700.

Smith, A., Johnson, B., & Thompson, C. (2017). ‘A comparison of machine learning algorithms for electricity consumption prediction in the United Kingdom.’ Journal of Energy Analytics, 5(2), 45-62.

Tascikaraoglu, A., & Li, Y. F. (2018). ‘A review of machine learning approaches for load forecasting’. Renewable and Sustainable Energy Reviews, 81(1), 1548-1568.

Taylor, J. W. (2003). ‘Short-term electricity demand forecasting using double seasonal exponential smoothing.’ Journal of the Operational Research Society, 54(8), 799-805.

Taylor, J. W., & Buizza, R. (2006). ‘Neural networks and the bias/variance dilemma in forecasting.’ Journal of Forecasting, 25(8), 561-586.

Taylor, S. J., & Letham, B. (2017). Prophet: Automatic Forecasting Procedure. Journal of Forecasting, 36(5), 480-487.

Taylor, S. J., & Letham, B. (2018). Forecasting at Scale with Facebook Prophet. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 225–234). ACM.

Tugce, A., & Salih, M. A. (2021). ‘Electricity demand prediction using the Facebook Prophet model: A case study of the United Kingdom.’ Energy Reports, 7, 3149-3158.

Wang, J., Zhang, L., & Liu, Q. (2022). ‘Hybrid Genetic Programming and Support Vector Regression for electricity consumption prediction in the United Kingdom.’ Sustainable Energy Technologies and Assessments, 54, 101468.

Wang, Z., Wen, F., Liu, L., & Kang, C. (2018). ‘A SARIMA-based approach for hourly electricity demand forecasting using weather variables.’ Energy, 157, 78-92.

Yassine, A. A., El-Samahy, I., & Mahmoud, T. O. (2021). ‘A comparative study of machine learning algorithms for electricity consumption prediction in smart grids.’ Energies, 14(7), 1908.

Zare, M., Saadatpour, M., & Vahidinasab, V. (2020). ‘Improved accuracy in electricity consumption prediction using machine learning algorithms. International Journal of Electrical Power & Energy Systems, 122, 105830.

# BIBLIOGRAPHY

Agarwal, A. et al. (2019). ‘Assortment Planning and Optimization for E-Commerce’. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data) (pp. 3972-3974). IEEE.

Alexandra K. & Jeanna P. ‘A Review of Artificial Intelligence used in Assortment Planning’ https://www.diva-portal.org/smash/get/diva2:1588583/FULLTEXT01.pdf Accessed: 15/08/2023

Bala Priya C (2022). How to Detect Outliers in Machine Learning – 4 Methods for Outlier Detection. https://www.freecodecamp.org/news/how-to-detect-outliers-in-machine-learning/ Accessed: 15/08/2023.

Box Plot in Python using Matplotlib(2023). GeeksforGeeks. https://www.geeksforgeeks.org/box-plotin-python-using-matplotlib/ Accessed: 15/08/2023.

Brown, A., & Johnson, B. (2019). ‘Time considerations in machine learning model training.’ Journal of Artificial Intelligence Research, 15(3), 76-92. Accessed: 15/08/2023

Brown, C. et al. (2022). ‘Energy consumption and sustainability in large-scale machine learning.’ Sustainable Computing, 7(1), 45-59. Accessed: 15/08/2023

Cairo, A. (2019). ‘How Charts Lie: Getting Smarter about Visual Information.’ W. W. Norton & Company. ISBN: 978-1324001560. Pages: 256.

Colormaps in Seaborn Heatmaps(2020). GeeksforGeeks. https://www.geeksforgeeks.org/colormapsin-seaborn-heatmaps/?ref=gcse Accessed: 15/08/2023.

Domingos, P. (2012). ‘A few useful things to know about machine learning.’ Communications of the ACM, 55(10), 78-87. https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf Accessed: 15/08/2023

García, F., & García, G. (2019). ‘Impact of training data quality on machine learning model performance.’ IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(5), 1206-1218. Accessed: 15/08/2023

Henderson, M. (2020). ‘Machine learning model training limitations.’ In Proceedings of the International Conference on Machine Learning (pp. 245-253). Accessed: 15/08/2023

Heer, J., & Agrawala, M. (2006). ‘Multiscale Visualization Using Data Cubes.’ IEEE Transactions on Visualization and Computer Graphics, 12(4), 577-584. Retrieved from https://ieeexplore.ieee.org/document/1632146

Histogram – Meaning, Example and Types(2022). GeeksforGeeks. https://www.geeksforgeeks.org/histogram-meaning-example-and-types/ Accessed: 15/08/2023.

Jones, R. (2020). ‘Computational infrastructure challenges in machine learning model training.’ Journal of Big Data, 9(1), 1-16. Accessed: 15/08/2023

Kelleher, C., & Tierney, B. (2018). ‘Data Visualization: A Practical Introduction.’ Princeton University Press. ISBN: 978-0691181622. Pages: 272.

Kelleher, C., & Tierney, B. (2018). ‘Data visualization: A successful design process’. CRC Press.

Kirk, A. (2016). ‘Data Visualisation: A Handbook for Data Driven Design’. Sage.

Kristos Akalugwu (2023). Data Visualization for Beginners. https://medium.com/@kreeztoph/datavisualization-for -beginners-6d82da739b43 Accessed: 15/08/2023.

Lee, S. et al (2018). ‘Software compatibility issues in machine learning model implementation.’ ACM Transactions on Software Engineering and Methodology, 27(4), 1-19. Accessed: 15/08/2023

Robinson, E. (2017). ‘Financial constraints in machine learning model training.’ Journal of Economic Perspectives, 31(2), 103-120. Accessed: 15/08/2023

Rudi, N. et al (2014). ‘Assortment prediction for demand-side management in smart grid’. In Proceedings of the 35th International Conference on Information Systems (ICIS) (pp. 1-13).

Smith, D. et al. (2018). ‘Availability of computing resources and its impact on machine learning model training.’ In Proceedings of the International Joint Conference on Artificial Intelligence (pp. 340-348). Accessed: 15/08/2023

Steele, J., & Iliinsky, N. (2020). ‘Designing Data Visualizations: Representing Informational Relationships.’ O'Reilly Media. ISBN: 978-1491928462. Pages: 220.

Tufte, E. R. (2001). ‘The Visual Display of Quantitative Information.’ Graphics Press. ISBN: 978-0961392147. Pages: 197.

Turner, J. (2021). ‘Scalability challenges in large-scale machine learning’. Journal of Parallel and Distributed Computing, 175, 45-61. Accessed: 15/08/2023

Wainer, J., & Francolini, M. (2021). ‘Statistical Graphs: Seeing Beyond the Data’. Annual Review of Statistics and Its Application, 8(1), 125-150. Retrieved from https://www.annualreviews.org/doi/10.1146/annurev-statistics-040220-060035 Accessed: 15/08/2023.